สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ตัวควบคุมดิจิทัลแบบเฟสล็อกโหมดและโหมดทั่วไป สำหรับใช้ในระบบงานอุตสาหกรรม

Phase-Locked and Conventional Mode Digital Controller for Industrial Plant



นาย เจตนา สิงห์สินธุ์

Mr. Jetanar Singsin

เลขหมู่
เลขทะเบียน 20168
วัน, เดือน, ป 18 พ.ค. 2537

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาตรมหาบัณฑิต
สาขาวิศวกรรมไฟฟ้า
บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง
ปีการศึกษา 2536

ISBN 947-621-050-5

ลิขสิทธิ์ของบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Phase-Locked and Conventional Mode Digital Controller for Industrial Plant

Mr. Jetanar Singsin

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE
MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING
GRADUATE SCHOOL
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
1993

ISBN 974-621-050-5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์ ตัวควบคุมดิจิทัลแบบเฟสล็อกโหมดและโหมดทั่วไปสำหรับใช้ในระบบงานอุตสาหกรรม

นักศึกษา นายเจตนา สิงห์สินธุ์

อาจารย์ที่ปรึกษา รศ.ดร. โยธิน เปรมปราณีรัชต์

ระดับการศึกษา วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมไฟฟ้า

ปีการศึกษา 2536

บทคัดย่อ

ผลงานวิจัยนี้ อธิบายถึงหลักการของเฟสล็อกลูปที่ประยุกต์ใช้เป็นตัวดิจิทัลคอนโทรลเลอร์สำหรับระบบงานอุตสาหกรรมทั่วๆไป ตัวคอนโทรลเลอร์นี้อยู่บนพื้นฐานของไมโครโปรเซสเซอร์โดยใช้ซอฟต์แวร์โปรแกรม ทำงานเหมือนกับเฟส/ความถี่ดีเทกเตอร์ ซึ่งสัญญาณนาฬิกาภายในใช้เป็นวงจรกำหนดจังหวะการทำงานให้กับไมโครโปรเซสเซอร์ ตัวดิจิทัลคอนโทรลเลอร์นี้ยังสามารถใช้งานในโหมดทั่วไปได้เช่น P PI และ PID ส่วนของไมโครคอมพิวเตอร์ที่มาสนับสนุน เปรียบเสมือนเป็นหน่วยหลักที่คอยควบคุมการทำงานและทำหน้าที่เป็น ดาต้าล็อกกิ่ง และพล้านต์มอนิเตอริง โดยรูปแบบของพล้านต์ มาแสดงบนจอ CRT อยู่ในรูปของ ฮอริซอนทอลบาร์ และแสดงผลกราฟแนวโน้มของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title Phase-locked and conventional mode digital controller for industrial plant

Name Mr. Jetanar Singsin

Thesis Advisor Assoc. Prof. Dr. Yothin Prempraneerach

Level of Study Master of Engineering in Electrical Engineering

Academic 1993

Abstract

A general purpose industrial digital controller based on the phase-locked loop principle is described. The controller by Phase Locked Loop can be achieved with a microprocessor. Software replaces the phase/frequency detector and the internal clock provides the curcuit timing. It can be used in conventional mode such as P , PI , PID . Microcomputer support the system which is in acting as monitoring which use for plant simulation and data logging in CRT display in form of horizontal bar and trend recording.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอกราบขอบคุณ คุณพ่อ และ คุณแม่ ซึ่งเป็นผู้มีพระคุณสูงสุด

งานวิจัยนี้สำเร็จลุล่วงด้วยดี เพราะพร้อมด้วยปัจจัยหลายอย่าง ผู้สนับสนุน เงินทุน ที่ปรึกษาต่างๆเหล่านี้เกิดจาก อาจารย์ และเพื่อน ของ ภาควิชาระบบควบคุม คณะวิศวกรรมไฟฟ้า สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง

โดยเฉพาะอย่างยิ่ง ขอขอบพระคุณอย่างสูง แด่ รศ.ดร. โยธิน เปรมปราณีรัตช์ ซึ่งเป็นที่ปรึกษาให้ข้อคิดเห็น เพื่อชี้แนะแนวทางในงานวิจัยนี้ พร้อมทั้งยังได้เป็นภาระในการจัดหาเงินทุนให้ด้วย

และยังได้แรงเชียร์ จากเพื่อนนักศึกษาปริญญาโท ดังนี้ คุณอนุตรชัย ณ ถลาง คุณสมศักดิ์ อภิรักษ์สมบัติ คุณสุมิตร คุณอธิชาติ และ น้องหญิง ซึ่งขอขอบคุณอย่างยิ่ง

สุดท้าย ขออาราธนาคุณพระศรีรัตนตรัย ได้คุ้มครอง ประเทศชาติ พระมหากษัตริย์ พร้อมทั้งประชาชนไทย ตลอดจนเพื่อนร่วมโลก และผู้ที่เอ่ยนามมาข้างต้น ด้วยเทอญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	IX
สารบัญภาพ	X
คำย่อและสัญลักษณ์	XVII
บทที่	
1 บทนำ	1
1.1 บทนำ	1
1.2 วัตถุประสงค์ของงานวิจัย	2
1.3 ขอบเขตของงานวิจัย	2
1.4 ประโยชน์ที่ได้รับจากงานวิจัย	3
2 ทฤษฎีดิจิทัลเฟสล็อกลูป	5
2.1 นิยามของหลักการทำงานเฟสล็อกลูปตามแบบฉบับ	5
2.2 นิยามดิจิทัลเฟสล็อกลูป	8
2.3 ดิจิทัลเฟสล็อกลูปแบบต่าง ๆกับสัญญาณแอนะล็อกตัวกลาง	9
2.3.1 อิทธิพลของเฟสดีเทกเตอร์แบบต่างๆ ทางไดนามิค	9
2.3.2 กระบวนการล็อกของเฟสล็อกลูป	16
2.3.3 การแทรคกิ้งของ DPLL ในสภาวะล็อก	21
2.4 ระบบเฟสล็อกลูปแบบดิจิทัลล้วนและแบบใช้ซอฟต์แวร์เป็นหลัก	25
2.4.1 โครงสร้างของบล็อกฟังก์ชันใน ADPLL	25
2.4.2 เฟสดีเทกเตอร์แบบดิจิทัลล้วน	26
2.4.3 ดิจิทัลลูปฟิลเตอร์	34
2.4.4 ดิจิทัลคอนโทรลออสซิลเลเตอร์(DCO)	40
2.4.5 ตัวอย่างการใช้ ADPLL เพิ่มเติม	43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3	**โหมดการควบคุมของคอนโทรลเลอร์**	48
	3.1 นิยามระบบควบคุม	48
	3.2 การควบคุมแบบเปิด-ปิด(	49
	3.3 การควบคุมแบบต่อเนื่อง	49
	3.3.1 โหมดการควบคุมแบบพรอปพรอชันแนล	49
	3.3.2 โหมดการควบคุมแบบอินทีกรัล	51
	3.3.3 โหมดการควบคุมแบบดีริเวทีฟ	52
	3.3.4 โหมดการควบคุมแบบคอมโพสิต	53
	3.4 บทนิยามอัลกอลิธึมของการควบคุม	53
	3.5 อัลกอลิธึมของโหมดควบคุม	57
	3.5.1 โหมดพรอปพรอชันแนล	57
	3.5.2 โหมดอินทีกรัลหรือโหมดรีเซต	61
	3.5.3 โหมดดีริเวทีฟ	67
	3.5.4 โหมดคอมโพสิต	70
4	**การออกแบบวงจรตัวดิจิทัลคอนโทรลเลอร์**	77
	4.1 โครงสร้างของตัวดิจิทัลคอนโทรลเลอร์	77
	4.1.1 ลักษณะทางสถาปัตยกรรมของ 8031โดยย่อ	78
	4.1.2 การออกแบบ 8031 ไมโครคอนโทรลเลอร์บอร์ด	80
	4.2 วงจรต่าง ๆในตัวดิจิทัลคอนโทรลเลอร์	82
	4.2.1 วงจรสแกนคีย์บอร์ด	82
	4.2.2 วงจรแสดงผล	85
	4.2.3 วงจร ADC และ DAC	87
	4.2.4 เฟสดีเทกเตอร์(PFD)และการเลียนแบบโดยซอฟต์แวร์	89
	4.2.5 วงจรใช้ในโหมดเฟสล็อกลูป	90
5	**หลักการออกแบบซอฟต์แวร์และการใช้งาน**	98
	5.1 จุดมุ่งหมายของการออกแบบซอฟต์แวร์	98
	5.2 วิธีการออกแบบซอฟต์แวร์โปรแกรมพล้านต์มอนิเตอริง	100
	5.2.1 แนวคิดการออกแบบซอฟต์แวร์พล้านต์มอนิเตอริง	100
	5.2.2 ทำไมต้องเลือกใช้ภาษา C++	101

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.3 โครงสร้างของซอฟต์แวร์โปรแกรม 102
5.2.4 หน้าที่ของสัปรูทีนในไฟล์ต่างๆ 103
5.3 แนวทางการออกแบบซอฟต์แวร์ของตัวดิจิทัลคอนโทรลเลอร์ 110
5.3.1 ไมโครคอนโทรลเลอร์และซอฟต์แวร์ 110
5.3.2 โครงสร้างของโปรแกรมมอนิเตอร์ของตัวดิจิทัลคอนโทรลเลอร์ 115
5.3.3 การจัดการทำงานของโปรแกรม 117
5.3.4 หน้าที่ของรูทีนคีย์บอร์ด 117
5.3.5 หน้าที่ของรูทีนแสดงผลบนตัวเลข 7 ส่วน 119
5.3.6 หน้าที่ของรูทีนเซตพารามิเตอร์ของตัวดิจิทัลคอนโทรลเลอร์ 121
5.3.7 หน้าที่ของรูทีนเลือกโหมดการสั่งการจากคีย์บอร์ด 122
5.3.8 หน้าที่ของรูทีนโหมดการควบคุมแบบ P 122
5.3.9 หน้าที่ของรูทีนโหมดการควบคุมแบบ PI 124
5.3.10 หน้าที่ของรูทีนโหมดการควบคุมแบบ PID 126
5.3.11 หน้าที่ของรูทีนโหมดการควบคุมแบบ PLL 127
5.4 วิธีการใช้งานตัวดิจิทัลคอนโทรลเลอร์ 132
5.4.1 การเซตโหมดการควบคุม 133
5.4.2 การเซตพารามิเตอร์การควบคุม KP , KI ,KD 134
5.4.3 การเซตค่าเซตพอย์น 136
5.4.4 แสดงค่า process variable (PV) 136
5.4.5 การเซตอัตราบอด 137
5.4.6 การเซตหมายเลขสถานี 137
5.5 วิธีการติดตั้งโปรแกรมพล้านต์มอนิเตอริง 138
5.5.1 การแนะนำการติดตั้งโปรแกรม 138
5.5.2 ระบบงานของโปรแกรม 141
5.5.3 เริ่มต้นรันโปรแกรมพล้านต์มอนิเตอริง 142
5.5.4 การใช้เมนูหลักชื่อ <FILE> 143
5.5.5 การใช้เมนูหลักชื่อ <OPTIONS> 145
5.5.6 การใช้เมนูหลักชื่อ <MONITOR> 149
5.5.7 การใช้เมนูหลักชื่อ <CONFIG> 156

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6 การทดสอบตัวดิจิทัลคอนโทรลเลอร์และโปรแกรมพล้านต์มอนิเตอริง 159
6.1 ลักษณะโดยทั่วไปของตัวดิจิทัลคอนโทรลเลอร์ 159
6.2 ผลการทดสอบดิจิทัลคอนโทรลเลอร์ในโหมดต่าง ๆ 161
6.2.1 การทดสอบในโหมดการควบคุมแบบเฟสล็อก 161
6.2.2 การทดสอบการควบคุมในโหมดพรอปพรอชันแนล 171
6.2.3 การทดสอบการควบคุมในโหมดพรอปพรอชันแนล-อินทีกรัล 174
6.2.4 การทดสอบการควบคุมในโหมดพรอปพรอชันแนล-อินทีกรัล-ดีริเวทีฟ 178
6.3 การวิเคราะห์ผลการทดลองในโหมดเฟสล็อก ในระบบควบคุมความเร็วมอเตอร์จำลอง 182
6.4 การวิเคราะห์ผลการทดลองในโหมด PI ในระบบควบคุมความเร็วมอเตอร์จำลอง 185
6.5 เออเรอร์ที่เกิดขึ้นจากการรับค่าอินพุตป้อนกลับ PV จากพล้านต์ 187
6.6 เหตุผลเปรียบเทียบการตอบสนองกระบวนการ ในขณะที่ใช้โหมดเฟสล็อก และโหมด PI ในการควบคุมความเร็วมอเตอร์ 188

7 สรุปวิจารณ์ในงานวิจัยตลอดจนปัญหาและอุปสรรค์ด้านเทคนิคของงานวิจัย 196
7.1 ข้อสรุปผลการทดลอง 196
7.1.1 ผลการทดสอบตัวดิจิทัลคอนโทรลเลอร์ 196
7.1.2 สรุปผลการทดสอบกับพล้านต์ออร์เดอร์หนึ่ง ระหว่าง PLL กับ PI 199
7.1.3 ผลการทดสอบโปรแกรมพล้านต์มอนิเตอริง 199
7.2 ความเป็นไปได้ในการนำผลงานวิจัยและพัฒนาไปใช้ในการผลิตเชิงอุตสาหกรรม 200
7.3 ข้อเสนอแนะและข้อคิดเห็น 201

บรรณานุกรม 202

ภาคผนวก

ก. ขบวนการพูล-อินของดิจิทัลเฟสล็อกลูปโดยใช้เฟสดีเทกเตอร์แบบ 4 203
ข. มอเตอร์ควบคุมโดยเฟสล็อกลูปบนไมโครโปรเซสเซอร์โดยใช้ซอฟต์แวร์เลียนแบบเฟสดิเทกเตอร์ 208

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค. ซอฟต์แวร์ลิสต์ภาษาแอสเซมบลี 8031 ของโปรแกรมของตัวดิจิทัล
คอนโทรลเลอร์ 215
ง. ซอฟต์แวร์สิสต์ภาษาซีของโปรแกรมพล้านต์มอนิเตอริง 238

ประวัติผู้เขียน 320

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่		หน้า
2-1	*โลว์พาสฟิลเตอร์ออร์เดอร์หนึ่งแบบต่างๆ*	23
4-1	*จำแนกการถอดรหัสแอดเดรสอุปกรณ์ภายนอก*	81
4-2	*แสดงไบต์ข้อมูลที่ถูกส่งจากวงจรสแกนคีย์เมื่อกดปุ่ม*	85
5-1	*แสดงการจัดหน่วยความจำของไมโครคอนโทรลเลอร์ 8031*	111
5-2	*แสดงความแตกต่างระหว่างเนตเวอร์กแบบสตาร์ และแบบลูป*	141
6-1	*การเปรียบเทียบผลการทดลองระหว่าง PLL กับ PI ในเงื่อนไขเดียวกัน*	189
6-2	*การเปรียบเทียบโอเวอร์ชูตของ PI และ PLL ใน τ_m ของพล้านต์ต่างๆ*	191
6-3	*การเปรียบเทียบอันเดอร์ชูตของ PI และ PLL ใน τ_m ของพล้านต์ต่างๆ*	192
6-4	*การเปรียบเทียบไรส์ไทม์ของ PI และ PLL ใน τ_m ของพล้านต์ต่างๆ*	193
6-5	*การเปรียบเทียบเซตตลิงไทม์ของ PI และ PLL ใน τ_m ของพล้านต์ต่างๆ*	194
6-6	*การเปรียบเทียบออฟเซตเออร์เรอร์ของ PI และ PLL ใน τ_m ของพล้านต์ต่างๆ*	195

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูปที่		หน้า
2-1	*(a) บล็อกไดอะแกรมของ ระบบเฟสล็อกลูป (b) ฟังก์ชันถ่ายโอนของ VCO (c) ฟังก์ชันถ่ายโอนของ PD*	5
2-2	*โครงสร้างของเสถียรภาพทางไดนามิคและทางสถิตของ PLL ออร์เดอร์สอง แบบเชิงเส้น*	7
2-3	*การกระทำของ PD แบบ 2 (a)-(e) แสดงการตอบสนองของ PD เมื่อเฟสเออเรอร์มีค่าต่างๆ*	10
2-4	*ผลการกระทำของเฟสดีเทกเตอร์แบบ 3 (a)-(g) ความแตกต่าง 7 แบบ ของออฟเซตของเฟสและความถี่*	12
2-5	*ผลการกระทำของ PD แบบ 4 โดย (a)-(d) แบ่งเป็น 4 กรณีของออฟเซต*	15
2-6	*ถ้า DPLL ใช้เฟสดีเทกเตอร์แบบ 4 รูปจำลองเชิงเส้นสามารถใช้อธิบายสภาวะไม่ล็อกของระบบได้ (a) บล็อกไดอะแกรมของระบบในโดเมนเวลา (b) รูปจำลองทางคณิตศาสตร์ในระบบในโดเมนความถี่*	16
2-7	*โมเดลคณิตศาสตร์แบบเชิงเส้นของ DPLL สำหรับสภาวะล็อก*	21
2-8	*พิสัยพูล-เอาต์(pull-out range)ของ PD แบบ 4 ตามฟังก์ชันของ ζ*	24
2-9	*โครงสร้างบล็อกฟังก์ชันแบบต่างๆใช้ภายในระบบ ADPLL*	26
2-10	*PD แบบฟลิปฟลอปตัวนับ (a) บล็อกไดอะแกรม (b) การตอบสนองรูปคลื่น*	27
2-11	*PD แบบ Nyquist-rate (a) บล็อกไดอะแกรม (b) การตอบสนอง*	28
2-12	*PD แบบ ซีโร่ครอสซิง แซมปลิงเฟสเออเรอร์ที่ขอบขาขึ้นของสัญญาณอ้างอิงเท่านั้น (a) บล็อกไดอะแกรม (PED = positive edge detector (b) รูปการตอบสนอง*	29
2-13	*PD แบบ ซีโร่ครอสซิง แซมปลิงเฟสเออเรอร์ทั้งสองขอบของสัญญาณอ้างอิง (a) บล็อกไดอะแกรม (PNED = positive and negative edge detecter) (b) การตอบสนองของรูปคลื่น*	31
2-14	*PD แบบ การแปลง ฮิลเบอร์ต (a) บล็อกไดอะแกรม (b) การตอบสนอง*	32
2-15	*PD แบบเฉลี่ยดิจิทัล(digital-averaging)*	33
2-16	*ลูปฟิลเตอร์แบบเคาน์เตอร์ UP-DOWN (a) บล็อกไดอะแกรม (b) การตอบสนอง*	35
2-17	*ลูปฟิลเตอร์แบบเคาน์เตอร์ K (a) บล็อกไดอะแกรม (b) การตอบสนอง*	36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่		หน้า
2-18	*บล็อกไดอะแกรมของลูปฟิลเตอร์แบบ N-before-M*	37
2-19	*บล็อกไดอะแกรมของลูปฟิลเตอร์ส่วนเดียวแบบรีเคอร์ซีฟออร์เดอร์หนึ่ง*	39
2-20	*บล็อกไดอะแกรมของ DCO แบบ หาร N*	40
2-21	*DCO แบบเคาน์เตอร์นับเพิ่มและนับลด (a) บล็อกไดอะแกรม (b) การตอบสนอง*	41
2-22	*DCO แบบสังเคราะห์รูปคลื่น (a) บล็อกไดอะแกรม (b) รูปคลื่นแสดงให้เห็นเป็นซายน์ความถี่ 1 Hz และ 0.5 Hz ตามลำดับ*	42
2-23	*.(a) บล็อกไดอะแกรม (b) การตอบสนองรูปคลื่น (1) u_1^* นำหน้า u_2 (2) u_1^* นำหน้า u_2*	44
2-24	*ระบบดิจิทัล PLL ระบบนี้ถูกทำให้เป็นจริงโดยวงจรรวมเบอร์ 74LS297*	45
2-25	*ระบบ PLL แบบดิจิทัลล้วน ตัวอย่างที่ 3 เป็นระบบทางซอฟต์แวร์ดีที่สุด*	46
3-1	*ส่วนประกอบของระบบควบคุมอย่างง่าย*	48
3-2	*แสดงกราฟการควบคุมแบบเปิดปิด และ เดดแบนด์*	49
3-3	*คอนโทรลเอาต์พุตในโหมด P จะราบเรียบ ภายในข้อกำหนดของเกน และเออเรอร์*	50
3-4	*ในกรณีที่เออเรอร์เป็นศูนย์ แต่เอาต์พุตดีริเวทีฟมีการตอบสนอง*	52
3-5	*เป็นโฟล์วชาร์ตแสดงค่าเออเรอร์ DE ทั้งขนาดและเครื่องหมาย*	56
3-6	*โฟล์วชาร์ตอัลกอลิธึมของโหมดพรอปพรอชันแนล*	58
3-7	*รายละเอียดของโฟล์วชาร์ตของโหมด P แสดงแฟลกเครื่องหมายของ DE และป้องกันโอเวอร์โฟล์วและอันเดอร์โฟล์ว*	60
3-8	*กราฟแสดงตัวอย่างพื้นที่เอุอเรอร์ต่อเวลาโดยประมาณพื้นที่เป็นสี่เหลี่ยม*	62
3-9	*โฟล์วชาร์ตโดยทั่วๆไปของโหมดอินทีกรัล*	64
3-10	*รายละเอียดโฟล์วชาร์ต ของโหมดอินทีกรัล แสดงถึงแอคคิวมูเลเตอร์ SUM และการป้องกันโอเวอร์โฟล์ว อันเดอร์โฟล์ว*	66
3-11	*โฟล์วชาร์ตสำหรับโหมดดีริเวทีฟแสดงถึงการกำหนดขั้วเครื่องหมาย*	69
3-12	*โฟล์วชาร์ตของโหมด PI แสดงถึงการป้องกันโอเวอร์โฟล์ว และอันเดอร์โฟล์ว*	74
4-1	*แสดงบล็อกไดอะแกรมของตัวดิจิทัลคอนโทรลเลอร์*	77
4-2	*วงจร 8031 บอร์ด*	79

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่		หน้า
4-3	*วงจรถอดรหัสแอดเดรดอุปกรณ์ภายนอก*	80
4-4	*บล็อกไดอะแกรมสแกนคีย์บอร์ด*	83
4-5	*วงจรสแกนคีย์บอร์ด*	84
4-6	*แสดงเป็นวงจรขับตัวเลข 7 ส่วนอย่างง่ายแบบทั่วไป*	86
4-7	*การขับตัวเลข 7 ส่วนโดยวิธีตรง*	87
4-8	*แสดงวงจร ADC และ DAC*	88
4-9	*ไดอะแกรมการทำงานแบบเฟสดีเทกเตอร์ไตร-สเตต*	93
4-10	กระบวนการ *Auto-reload ของ TLX จาก THX*	93
4-11	*แสดงบล็อกไดอะแกรมของวงจรสนับสนุนโหมด PLL*	95
4-12	*บล็อกไดอะแกรมของ DPLL ควบคุมความเร็วมอเตอร์*	96
4-13	*ฟังก์ชันถ่ายโอน ของ VCO เปรียบเทียบกับมอเตอร์*	96
4-14	*วงจรดิจิทัลเฟสล็อกลูป*	97
5-1	*แผงควบคุมซึ่งแทนห้องควบคุมแบบดั้งเดิม*	99
5-2	*ห้องควบคุมแบบ และเครื่องมือต่างๆสำหรับการควบคุมแบบใหม่*	100
5-3	*โครงสร้างไฟล์มอดูล*	102
5-4	*แสดงเมนูหลัก FILE และเมนูย่อยต่างๆซึ่งตอบสนองการทำงานของสัปรูทีน DO00 ถึง DO03*	104
5-5	*แสดงเมนูหลัก OPTION และเมนูย่อยต่างๆซึ่งตอบสนองการทำงาน ของสัปรูทีน DO10 ถึง DO13*	105
5-6	*แสดงเมนูหลัก monitor และเมนูย่อยต่างๆซึ่งตอบสนองการทำงาน ของสัปรูทีน DO20 ถึง DO23*	105
5-7	*แสดงเมนูหลัก CONFIG และเมนูย่อยต่างๆซึ่งตอบสนองการทำงาน ของสัปรูทีน DO30 ถึง DO33*	106
5-8	*บล็อกไดอะแกรมของซอฟต์แวร์โปรแกรมมอนิเตอร์ของคอนโทรลเลอร์*	116
5-9	*บล็อกไดอะแกรมของซอฟต์แวร์รูทีนคีย์บอร์ด*	118
5-10	*โฟล์วชาร์ตของการแสดงผลค่าเปอร์เซ็นต์*	120
5-11	*แสดงโฟล์วชาร์ตของการเปลี่ยนค่าเกน*	122

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ หน้า

5-12 แสดงโฟล์วชาร์ตซึ่งสอดคล้องกับรูทีนแอสเซมบลีในการควบคุมแบบพรอปพรอชันแนล ซึ่งรายละเอียดของโฟล์วชาร์ตนี้อธิบายไว้ในบทที่ 3 124

5-13 ไดอะแกรมเวลาของการทำงานในโหมดเฟสล็อก 128

5-14 โฟล์วชาร์ตของเฟสดีเทกเตอร์แบบไตร-สเตต 131

5-15 แสดงหน้าปัดด้านหลังของตัวคอนโทรลเลอร์ 132

5-16 การกำหนดรูปแบบการสื่อสารแบบอนุกรม 140

5-17 ไตเติลเริ่มต้นของโปรแกรม 142

5-18 แสดงเมนูหลัก <FILE> และเมนูย่อย 143

5-19 แสดงการใส่ชื่อไฟล์ของเมนูย่อย<OPEN TAG FILE> 144

5-20 เมนูย่อยของเมนูหลัก<OPTIONS> 146

5-21 แสดงตัวเลือกช่องการติดต่อแบบอนุกรม 146

5-22 แสดงตัวเลือกของเมนูย่อย<BAUD RATE> 147

5-23 แสดงตัวเลือกจำนวนกราฟของเมนูย่อย<NO. GRAPH> 147

5-24 แสดงการใส่ค่าแซมปลิงในเมนูย่อย <SET SAMPLING> 148

5-25 แสดงเมนูย่อยของเมนูหลัก <MONITOR> 150

5-26 แสดงการป้อนพารามิเตอร์ของเมนูย่อย<TAG edit> 151

5-27 เมื่อเลือก trend graph จอภาพจะแสดงดังรูป เพื่อให้ใส่หมายเลข ตัวดิจิทัลคอนโทรลเลอร์ที่ต้องการแสดงผล 154

5-28 แสดงกราฟบันทึกแนวโน้มข้อมูล 4 กราฟ 154

5-29 แสดงกราฟบาร์ของเมนูย่อย<BAR> 156

5-30 แสดงเมนูย่อยของเมนูหลัก<CONFIG> 157

5-31 แสดงค่าโครงแบบของโปรแกรมในเมนูย่อย <list config> 158

6-1 แสดงด้านหน้าของตัวดิจิทัลคอนโทรลเลอร์ต้นแบบ 159

6-2 แสดงด้านหลังของตัวดิจิทัลคอนโทรลเลอร์ต้นแบบ 160

6-3 แสดงอุปกรณ์ภายในของดิจิทัลคอนโทรลเลอร์ 160

6-4 บล็อกไดอะแกรมเพื่อการทดสอบลักษณะเฉพาะของ PD 161

6-5 แสดงสภาวะไตร-สเตตของซอฟต์แวร์เฟสดีเทกเตอร์ ถ้าความถี่ Fr น้อยกว่า F_b จากเครื่องวิเคราะห์ลอจิก 162

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ หน้า

6-6 *แสดงสภาวะไตร-สเตตของซอฟต์แวร์เฟสดีเทกเตอร์ ถ้าความถี่ Fr มากกว่า F_b จากเครื่องวิเคราะห์ลอจิก* 163

6-7 *ลักษณะการต่อแบบลูปปิดของ PLL เพื่อทดสอบการล็อก* 164

6-8 *แสดงความถี่สถาวะล็อกของระบบ PLL ตามเงื่อนไขข้างต้น บน: ความถี่ป้อนกลับจากระบบ ล่าง: ความถี่อ้างอิง* 165

6-9 *การตอบสนองสเต็ปขึ้น 0%-50% เซตพอย์น เงื่อนไข LF = (1+0.39s)/(1+0.72s) มอเตอร์ = 1/(1+0.33s) บน: เอาต์พุตมอเตอร์ ล่าง: เอาต์พุตฟิลเตอร์* 166

6-10 *การตอบสนองสเต็ปลง 0%-50% เซตพอย์น เงื่อนไข LF=(1+0.39s)/(1+0.72s) มอเตอร์ = 1/(1+0.33s) บน: เอาต์พุตมอเตอร์ ล่าง: เอาต์พุตฟิลเตอร์* 167

6-11 *รูป(a) การตอบสนองสเต็ปขึ้น 50%-95% เซตพอย์น รูป(b) การตอบสนองสเต็ปลง 95%-50% เซตพอย์น เงื่อนไข LF=(1+0.39s)/(1+0.72s) มอเตอร์ = 1/(1+0.33s) บน: เอาต์พุตมอเตอร์ ล่าง: เอาต์พุตฟิลเตอร์* 168

6-12 *รูป(a) การตอบสนองสเต็ปขึ้น 0%-100% เซตพอย์น รูป(b) การตอบสนองสเต็ปลง 100%-0% เซตพอย์น เงื่อนไข LF=(1+0.39s)/(1+0.72s) มอเตอร์ = 1/(1+0.33s) บน: เอาต์พุตมอเตอร์ ล่าง: เอาต์พุตฟิลเตอร์* 169

6-13 *เทียบเฟสในขณะล็อกที่เซตพอย์น 0% บน:เอาต์พุตเอนโค๊ดเดอร์ ล่าง: เอาต์พุตเซตพอย์น เงื่อนไข LF=(1+0.39s)/(1+0.72s) มอเตอร์ = 1/(1+0.33s)* 170

6-14 *เทียบเฟสในขณะล็อกที่เซตพอย์น 95% บน:เอาต์พุตเอนโค๊ดเดอร์ ล่าง: เอาต์พุตเซตพอย์น เงื่อนไข LF=(1+0.39s)/(1+0.72s) มอเตอร์ = 1/(1+0.33s)* 171

6-15 *แสดงวงจรทดสอบของตัวดิจิทัลคอนโทรลเลอร์ในโหมดทั่วไป* 172

6-16 *บล็อกไดอะแกรมการควบคุมแบบ P โดยให้พล้านต์เป็นออร์เดอร์หนึ่ง* 172

6-17 *การซิมูเลชันระบบในการควบคุมแบบ P* 172

6-18 *บล็อกไดอะแกรมของการควบคุมแบบ P โดยสังเกตจากโฮส พล้านต์ =1/(1+0.33s) ,KP=8.0* 173

6-19 *บล็อกไดอะแกรมของการควบคุมแบบ P โดยสังเกตจากสโคป พล้านต์ =1/(1+0.33s) ,KP=8.0 บน: PV สาง: CV* 173

6-20 *ผลการซิมูเลชั่นลูปเปิดของ* PI-control 174

6-21 *เป็นการตอบสนองสเต็ป 25% อินพุตการควบคุมแบบ PI โดยวัดจากออสซิลโลสโคป โดย Kp=1.0 ,Ki = 0.2 และ PV = 0%* 174

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ หน้า

6-22 แสดงบล็อกไดอะแกรมของการควบคุมแบบ PI โดยให้พล้านต์เป็นสมการออร์เดอร์หนึ่ง 175

6-23 (a),(b),ผลการซิมูเลชั่นโดยการควบคุมแบบ PI โดยให้พล้านต์เป็นสมการออร์เดอร์หนึ่ง ในขณะค่า Kp,Ki ต่างกัน โดยโปรแกรมคณิตศาสตร์ 175

6-24 ผลการตอบสนองสเต็ปของการควบคุมแบบ PI โดยให้พล้านต์เป็นสมการออร์เดอร์หนึ่ง ในขณะค่า Kp,Ki ต่างกัน โดยสังเกตจากโฮสต์ 176

6-25 ผลการตอบสนอง 50% สเต็ปของการควบคุมแบบ PI โดยให้พล้านต์เป็นสมการออร์เดอร์หนึ่ง ในขณะค่า Kp,Ki ต่างกัน โดยวัดได้จากสโคป 177

6-26 การซิมูเลชั่นของการตอบสนองสเต็ปของ PID ในลูปเปิด 178

6-27 การตอบสนองสเต็ป 25% ของ PID โดยวัดจากสโคป Kp=1 , Ki= 0.2 และ PV = 0 178

6-28 แสดงบล็อกไดอะแกรมของการควบคุมแบบ PID โดยให้พล้านต์เป็นสมการออร์เดอร์หนึ่ง 179

6-29 (a),(b) ผลการซิมูเลชันโดยการควบคุมแบบ PID โดยให้พล้านต์เป็นสมการออร์เดอร์หนึ่ง ในขณะค่า Kp,Ki,Kd ต่างกัน โดยโปรแกรมคณิตศาสตร์ 179

6-30 ผลการตอบสนอง 50 % สเต็ปของ PID ในขณะค่า Kp,Ki ,Kd ต่างกัน โดยสังเกตจากโฮส 180

6-31 ผลการตอบสนอง 50 % สเต็ปของ PID ในขณะค่า Kp,Ki ,Kd ต่างกัน โดยวัดจากสโคป 181

6-32 บล็อกไดอะแกรมของระบบควบคุมแบบความเร็วมอเตอร์ โดยใช้เฟสล็อก 182

6-33 บล็อกไดอะแกรมลดรูปให้ง่ายขึ้น ของรูปที่ 6-32 182

6-34 แสดงแอมปลิจูดพล็อต ของโบดไดอะแกรม 184

6-35 แสดงเฟสพล็อต ของโบดไดอะแกรม 184

6-36 เป็นกราฟฟังก์ชันการแปลงของ A/D ซึ่งมีค่าความเที่ยงตรง(accuracy) 1 บิต 187

6-37 การใช้ซอฟต์แวร์วนลูปเพื่อแซมปลิงความถี่อินพุต ในโหมดเฟสล็อก 188

6-38 ผลการทดสอบโหมด PLL โดยแปรค่า τ_m ของพล้านต์ออร์เดอร์หนึ่งใช้ LF=(1+0.39s)/(1+0.72s) , บน: เอาต์พุตพล้าน, ล่าง:เอาต์พุต LF 189

6-39 ผลการทดสอบโหมด PI โดยแปรค่า τ_m พล้านต์ออร์เดอร์หนึ่ง ใช้ Kp = 4 , Ki = 20 , บน: เอาต์พุตพล้าน(PV), ล่าง:เอาต์พุตควบคุม(CV) 190

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ หน้า

6-40 *กราฟแสดงการเปรียบเทียบข้อมูลโอเวอร์ชูต จากตารางที่ 6-2* 191
6-41 *กราฟแสดงการเปรียบเทียบข้อมูลอันเดอร์ชูต จากตารางที่ 6-3* 192
6-42 *กราฟแสดงการเปรียบเทียบไรส์ไทม์ข้อมูลจากตารางที่ 6-4* 193
6-43 *กราฟแสดงการเปรียบเทียบข้อมูลเซตตลิงไทม์ จากตารางที่ 6-5* 194
6-44 *กราฟแสดงการเปรียบเทียบข้อมูลออฟเซตเออร์เรอร์ จากตารางที่ 6-6* 195

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำย่อและสัญลักษณ์

8031 - ไมโครโปรเซสเซอร์แบบ 8031 ของบริษัท อินเทล(Intel)

<XXX> - สัญญาลักษณ์แสดงแทนเมนูหลักในโปรแกรมพล้านมอนิเตอริ่ง

ADC,A/D - ตัวแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล(Analog toDigital Converter)

ADLL - ระบบดิจิทัลเฟสล็อกลูปล้วน(All Digital Phase Locked Loop)

CV - เอาต์พุตคอนโทรล (Control Variable)

CRT - จอภาพแบบใช้หลอดแคโทด(Cathode Ray Tube)

DAC,D/A - ตัวแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อก(Digital to Analog Converter)

DCO - ดิจิทัลคอนโทรลออสซิลเลเตอร์

DPLL - ดิจิทัลเฟสล็อกลูป(Digital Phase Locked Loop)

INP - อินพุต (Input)

LAN - ระบบเครือข่ายงานบริเวณเฉพาะที่ (Local Area Network)

PD - เฟสดีเทกเตอร์(Phase Detecter) แบ่งเป็น 4 แบบด้วยกัน แบบที่ 1 คือ เป็นตัวคูณสี่ควอดแดรนต์ แบบที่ 2 คือเกทแบบเอกซ์คลูสีฟออร์ แบบที่ 3 คือฟลิบฟลอบแบบ J-K และแบบที่ 4 คือ PFD

PFD - เฟส/ความถี่ดีเทกเตอร์(Phase Frequency Detecter)

PLL - เฟสล็อกลูป(Phase Locked Loop)

PV - Process Variable

PWM - พัลส์วิทด์มอดูเลชัน(Pulse Width Modulation)

RC - หมายถึงวงจรซึ่งประกอบด้วยตัวต้านทานและตัวเก็บประจุ

SP - เซตพอย์น

TLX ,TRX - เรจิสเตอร์พิเศษของไมโครโปรเซสเซอร์เบอร์ 8031 สำหรับการสื่อสารอนุกรม

VCO - โวลต์เตจคอนโทรลออสซิเลเตอร์ (Voltage Control Oscilator)

XOR - เกทแบบเอกซ์คลูสีฟออร์

Z^{-1} - ดีเลย์โอเปอร์เรเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 บทนำ

ในปัจจุบันนี้ระบบต่างๆซับซ้อนมากขึ้น ซึ่งย่อมเกี่ยวพันกับการควบคุม ซึ่งในปัจจุบันมีการควบคุมหลายวิธี เช่นการควบคุมแบบต่อเนื่อง เช่น PID เป็นต้น ในบางงานระบบควบคุมสมัยใหม่จะใช้วิธีเฟสล็อกลูปกันมากขึ้น ซึ่งเป็นแบบอย่างมาจากวงการสื่อสาร สิ่งหนึ่งที่ขาดไม่ได้ก็คือ การนำไมโครโปรเซสเซอร์มาประยุกต์ใช้กับงานต่างที่เกี่ยวกับในระบบควบคุม ดังนั้นในผลงานวิจัยนี้จึงได้นำเอา 2 สิ่งดังกล่าวมาประยุกต์ใช้งาน โดยการนำเอาไมโครคอนโทรลเลอร์ หรือไมโครโปรเซสเซอร์มาพัฒนาเป็นตัวคอนโทรลเลอร์แบบเฟสล็อกลูป โดยทำงานตามลักษณะของไอซีเฟสล็อกลูปแบบสำเร็จรูป มาใช้ประโยชน์ที่ได้ในการซิงโครไนซ์สัญญาณอ้างอิงกับสัญญาณป้อนกลับจากกระบวนการ ซึ่งเป็นลักษณะการเปรียบเทียบรูปคลื่นทางเฟส และความถี่อินพุตความถี่เปรียบเทียบและสัญญาณความถี่อ้างอิง เพื่อให้ได้เอาต์พุตออกมาใช้งานควบคุมตามความต้องการ ซึ่งในส่วนซอฟต์แวร์โปรแกรมแทนเฟสดีเทกเตอร์ จะมีความยืดหยุ่นในการใช้งานดีกว่าการใช้ไอซีสำเร็จรูป โดยพยายามรักษาคุณสมบัติความเที่ยงตรง และความเร็วในการแทรคกิ้งของระบบเฟสล็อกไว้

ตัวดิจิทัลคอนโทรลเลอร์ที่ใช้ไมโครโปรเซสเซอร์เป็นพื้น ซึ่งจะทำงานในการควบคุมเป็นแบบเฟสล็อกโหมด โดยออกแบบซอฟต์แวร์โปรแกรมเลียนแบบการทำงาน ตัวดีเทกเตอร์เฟส/ความถี่ด้วยอัลกอลิธึมลักษณะการทำงานแบบไตร-สเตต

นอกจากนี้ ส่วนของไมโครคอนโทรลเลอร์จะทำงานเป็นตัวคอนโทรลเลอร์แบบเฟสล็อกแล้วยังทำการมีโหมดทั่วไปไว้ให้เลือกใช้งาน คือ โหมดพรอปพรอชันแนล หรือ โหมด P โหมดพรอปพรอชันแนล-อินทีกรัล หรือ PI และ โหมดพรอปพรอชันแนล-อินทีกรัล-ดีริเวทีฟ หรือ PID ซึ่งเป็นโหมดใช้ในระบบควบคุมทั่วๆไป

อีกเพิ่มระบบเครือข่ายโดยใช้ไมโครคอมพิวเตอร์นำมาใช้ช่วยงาน จะเป็นเสมือนเป็นหน่วยหลัก ที่คอยควบคุมดูแลตัวคอนโทรลเลอร์ ในลักษณะของเครือข่าย หรือระบบ LAN โดยให้ไมโครคอมพิวเตอร์สามารถทำหน้าที่เป็น พล้านต์มอนิเตอริง ด้วยการจำลองรูปแบบของพล้านต์ หรือการนำเอากระบวนการมาแสดงบนจอภาพ ซึ่งการแสดงมีหลายรูปแบบ เช่น แสดงรูปจำลองของพล้านต์(plant mimic) ฮอริซอนทอลบาร์(horizontal bar) และ กราฟการแสดงแนวโน้มของข้อมูล(trend recording) เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวคอนโทรลเลอร์ตัวเดียวนั้นสามารถใช้งานได้ในระบบเพียงลูปเดียว โดยจะทำงานแบบอิสระไม่ขึ้นอยู่กับใคร(stand alone) ดังนั้นในงานควบคุมทางอุตสาหกรรมจริงจะเป็นระบบใหญ่ จึงจำเป็นต้องตัวคอนโทรลเลอร์หลายตัวในการควบคุม จึงต้องนำตัวคอนโทรลเลอร์เหล่านั้นมาออน-ไลน์(On-Line) โดยใช้ไมโครคอมพิวเตอร์เป็นหน่วยกลางคอยดูแล โดยใช้งานเหมือนกับการดูแลพล้านต์ หรือเรียกอีกอย่างหนึ่งว่าเป็น "ระบบพล้านต์มอนิเตอริง"

1.2 วัตถุประสงค์ของงานวิจัย

เพื่อหาแนวทางพัฒนาตัวควบคุมแบบเฟสล็อก เพื่อให้ใช้เป็นตัวควบคุมแบบมาตราฐาน ซึ่งใช้ในอุตสาหกรรมทั่วไปเหมือนกับตัวควบคุมแบบปกติ ซึ่งนิยมใช้กันอยู่ทั่วไป โดยพยายามใช้ข้อ-ดีต่างๆของระบบเฟสล็อกลูปมาใช้ปฏิบัติและพัฒนา ที่สำคัญได้ทำเป็นต้นแบบเพื่อไปทดลองใช้งานกับกระบวนการได้

1.3 ขอบเขตของงานวิจัย

ขอบเขตในงานวิจัย ในจุดเริ่มแรก ได้มาถึงระบบเฟสล็อกลูปกำลังเป็นดาวรุ่ง ซึ่งเดิมทีถูกจำกัดอยู่ในแวดวงการสื่อสาร แล้วมีการพัฒนามาใช้ระบบควบคุมความเร็วของมอเตอร์ ซึ่งประสบความสำเร็จอย่างท่วมท้น อย่างไรก็ได้การนำเอาระบบเฟสล็อกลูปมาใช้กับระบบควบคุมนั้น ยังต้องคำนึงถึงความสามารถของเฟสดีเทกเตอร์บางแบบในวงจรสำเร็จรูปอยู่ ซึ่งผลลัพธ์ที่ได้เป็นที่ยอมรับว่า ตัวดีเทกเตอร์เฟส/ความถี่ จะมีคุณสมบัติดีที่สุดในบรรดาตัวดีเทกเตอร์ต่างๆในระบบเฟสล็อก-ลูป

ในปัจจุบันตัวคอนโทรลเลอร์ในแบบการควบคุมทั่วไป ซึ่งถ้าหากใช้ไมโครโปรเซสเซอร์เลียน-แบบอัลกอลิธึมในการทำงานของมันแล้ว จะเรียกอีกอย่างหนึ่งว่า "ดิจิทัลคอนโทรลเลอร์" ซึ่งในทางอุตสาหกรรมเราหันมาใช้ตัวคอนโทรลเลอร์แบบดิจิทัลมากขึ้น เพราะความยืดหยุ่นในการใช้งาน การเปลี่ยนแปลงพารามิเตอร์ และยังมีความสามารถทัดเทียมระบบแอนะล็อกมากขึ้น ซึ่งส่วนหนึ่งเป็นเพราะความสามารถของซีพียู ซึ่งปัจจุบันมีความเร็วสูง

ดังนั้นจึงไม่อาจมองข้ามในการนำไมโครโปรเซสเซอร์ ในการนำไปประยุกต์เกี่ยวกับระบบเฟสล็อกลูปได้ ถึงแม้ว่าระบบเฟสล็อกลูปเดิมซึ่งเป็นวงจรอิเล็กทรอนิกส์นั้นก็มีความสามารถดีเลิศแล้ว การนำไมโครโปรเซสเซอร์มาเลียนแบบอัลกอลิธึมของตัวเฟสดีเทกเตอร์ ปัจจุบันยังต้องมีการพัฒนาให้เพื่อให้มีความสามารถดีเทียบเท่ากับวงจรอิเล็กทรอนิกส์ ซึ่งในผลงานฉบับนี้ได้นำเสนอขั้นตอนการพัฒนาไว้อย่างละเอียด อีกทั้งผลงานที่ได้รับก็เป็นไปตามคาดหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้อ่านบางท่านสงสัยว่าในเมื่อระบบเฟสล็อกลูปในปัจจุบันเป็นขบวนการดิจิทัล ซึ่งใช้เป็นวงจรอิเล็กทรอนิกส์สำเร็จรูปนั้นดีอยู่แล้ว ทำไมจึงต้องไปใช้ไมโครโปรเซสเซอร์เลียนแบบด้วย ทั้งที่จะพัฒนาให้ความสามารถเทียบเท่านั้นเป็นไปได้ยาก โดยคำนึงว่า งานทุกที่กระทำอย่างเราจะหวังจะได้ผลลัพธ์ที่ดีสุด โดยไม่ต้องคำนึงถึงวิธีการปฏิบัติ หรือกล่าวอีกนัยหนึ่งว่า สิ่งที่ได้ทำนั้นดีกว่าเก่าอย่างไร ซึ่งคำตอบที่ได้นั้นคือ ผลลัพธ์ในการเลียนแบบขบวนการของเฟสดีเทกเตอร์โดยซอฟต์แวร์นั้น ไม่อาจดีกว่าวงจรอีเล็กทรอนิกส์ได้ อย่างดีที่สุดก็เทียบเท่า

เปรียบเทียบเหมือนกับ การใช้ซอฟท์แวร์แทนการควบคุมในโหมดทั่วไป เช่น โหมด PID โดยการใช้ไมโครโปรเซสเซอร์ ไปควบคุมลูปของกระบวนการ โดยสมมุติให้ขบวนการมีค่าคงที่เวลาน้อยกว่าอัตราแซมปลิง ผลลัพธ์ในการควบคุมก็ยังเกิดปัญหาบางอย่างขึ้นได้ ดังนั้นการใช้ซอฟต์แวร์แทนโหมดควบคุมปกติไม่อาจเทียบกับการควบคุมโดยใช้วงจรแอนะล็อกได้ อย่างไรก็ดีประโยชน์ได้ที่รับในการนำไมโครโปรเซสเซอร์มาใช้งานคือ ความก้าวหน้าอีกขั้นหนึ่ง

งานวิจัยนี้ได้นำเสนอเฉพาะ การใช้ทดสอบกับพล้านต์ซึ่งแทนระบบควบคุมความเร็วมอเตอร์ โดยพล้านต์จะอยู่ในรูปของดีเลย์ออร์เดอร์เหนึ่ง โดยทดสอบเปรียบเทียบระหว่างโหมดการควบคุมแบบเฟสล็อก กับโหมดโหมดการควบคุมแบบ PI

เราสามารถสรุปขอบเขตของงานวิจัยไว้ดังนี้

1.การใช้ไมโครโปรเซสเซอร์ เลียนแบบอัลกอลิธึมการทำงาน ของเฟส/ความถี่ดีเทกเตอร์ ในระบบเฟสล็อกลูป

2.การนำตัวคอนโทรลเลอร์แบบเฟสล็อกมาใช้งานในการควบคุมระบบจริง เช่น ควบคุมความเร็วของมอเตอร์

3.สร้างต้นแบบของตัวดิจิทัลคอนโทรลเลอร์ซึ่งมีโหมดควบคุมปกติ แบบ P ,PI และ PID และยังเพิ่มโหมดควบคุมแบบเฟสล็อก รวมเข้าด้วย

4.การพัฒนาตัวดิจิทัลคอนโทรลเลอร์ให้ใช้เป็นระบบเครือข่าย เพื่อให้ก้าวทันกับระบบอุตสาหกรรมในปัจจุบัน

5.สร้างระบบพล้านต์มอนิเตอริง ซึ่งใช้คอมพิวเตอร์เป็นศูนย์กลางของข้อมูล และข่าวสารของการปฏิบัติงานตัวดิจิทัลคอนโทรลเลอร์อยู่ในเครือข่ายได้

1.4 ประโยชน์ที่ได้รับจากงานวิจัย

เป็นการพัฒนาตัวดิจิทัลคอนโทรลเลอร์ในโหมดต่างๆ ในตัวเดียวกัน และระบบพล้านต์มอนิเตอริงนั้น จะเป็นประโยชน์มากในระบบอุตสาหกรรม ซึ่งมีขบวนการในการผลิตที่ซับซ้อน เช่น อุตสาหกรรมเคมีภัณฑ์ เป็นต้น โดยปกติแล้ว ระบบพล้านต์มอนิเตอริง นี้จะใช้ในอุตสาหกรรมขนาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใหญ่ หรืออุตสาหกรรมที่มีมูลค่าสูงเท่านั้น เพราะว่าการติดตั้งระบบพล้านต์มอนิเตอริงนั้นต้องสั่งจากต่างประเทศ ซึ่งมีค่าใช้จ่ายมาก จึงไม่นิยมใช้กันในอุตสาหกรรมขนาดกลาง หรือต่ำกว่า อย่างไรก็ตามการที่เราได้พัฒนาได้เองนั้น ทำให้ประหยัดค่าใช้จ่ายอย่างมาก ต่ออุตสาหกรรมขนาดเล็กและขนาดกลางที่ต้องการนำระบบพล้านต์มอนิเตอริงมาใช้

อีกทั้งในโหมดควบคุมแบบเฟสล็อก ยังเป็นตัวเลือกใหม่ในการควบคุม ซึ่งในงานบางอย่างเราสามารถใช้แทนโหมดควบคุมแบบปกติได้ดี เช่น สมมุติถ้าขบวนการมีการเปลี่ยนแปลงฟังก์ชันถ่ายโอน เราจะต้องมีปรับพารามิเตอร์ของโหมดควบคุมปกติอยู่เสมอ แต่เหล่านี้จะไม่เกิดขึ้นโหมดการควบคุมแบบเฟสล็อกในช่วงแปรผันค่าหนึ่ง

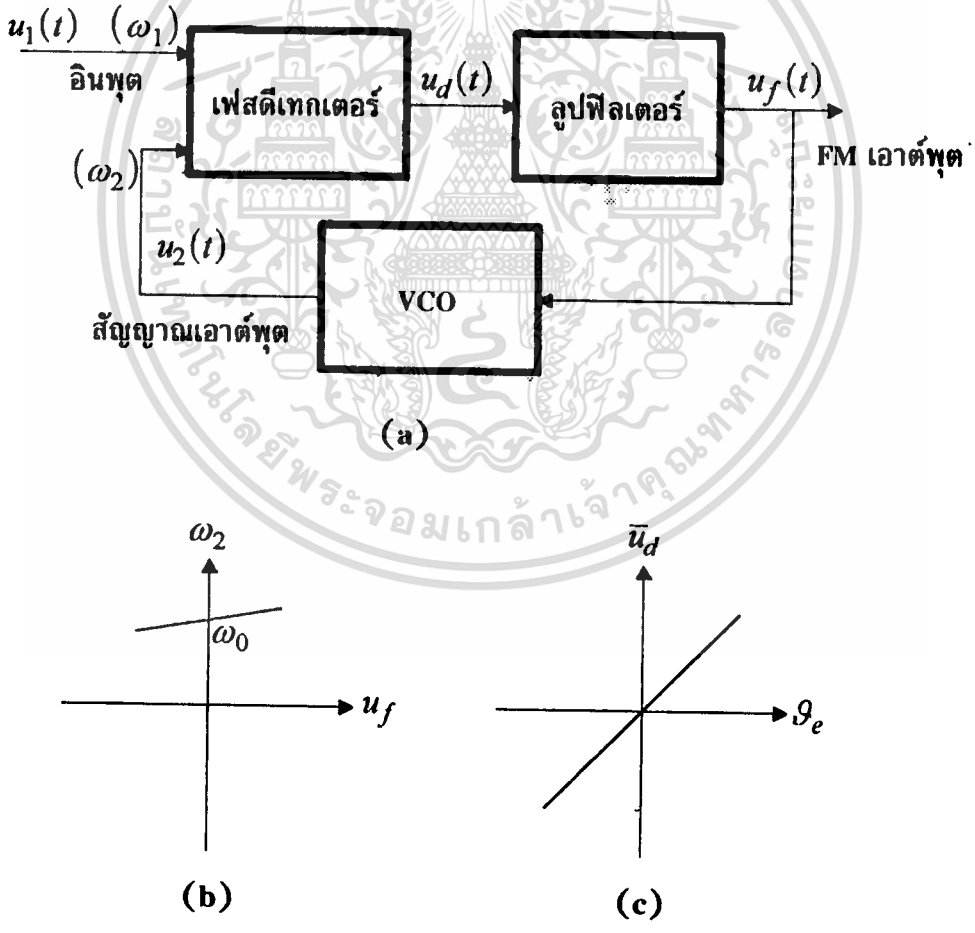
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีดิจิทัลเฟสล็อกลูป

2.1 นิยามของหลักการทำงานเฟสล็อกลูปตามแบบฉบับ

เฟสล็อกลูปคือ วงจรพิเศษที่ซึ่งทำให้ระบบหนึ่งปฏิบัติตามรอยอีกระบบหนึ่งได้ หรืออธิบายให้ชัดเจนคือ เฟสล็อกลูป(PLL)เป็นวงจรที่ทำซิงโครไนซ์สัญญาณเอาต์พุตกับสัญญาณอ้างอิงหรือสัญญาณอินพุตในความถี่และเฟสเดียวกัน ซึ่งการซิงโครไนซ์นี้ถูกเรียกบ่อยๆว่า "สภาวะล็อก (locked state)" โดยที่เฟสเออเรอร์ระหว่างออสซิลเลเตอร์เอาต์พุตกับสัญญาณอ้างอิงนั้นเป็นศูนย์หรือน้อยมาก



รูปที่ 2-1 *(a) บล็อกไดอะแกรมของระบบเฟสล็อกลูป (b) ฟังก์ชันถ่ายโอนของ VCO (c) ฟังก์ชันถ่ายโอนของ PD*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นั่นเป็นหลักการโดยย่อของเฟสล็อกลูป บล็อกไดอะแกรมแสดงตามรูปที่ 2-1 โดยวงจรเฟส-ล็อกลูป จะประกอบด้วยบล็อกฟังก์ชัน 3 ส่วนดังนี้

1. ออสซิลเลเตอร์ควบคุมด้วยแรงดันไฟฟ้า(VCO)
2. เฟสดีเทกเตอร์(PD)
3. ลูปฟิลเตอร์(LF)

สัญญาณภายในวงจรเฟสล็อกลูปที่น่าสนใจกำหนดดังนี้

- สัญญาณอ้างอิง หรืออินพุต $u_1(t)$
- ความถี่เชิงมุม(angular frequency) ω_1 ของสัญญาณอ้างอิง
- สัญญาณเอาต์พุต $u_2(t)$ ของ VCO
- ความถี่เชิงมุม ω_2 ของสัญญาณเอาต์พุต
- สัญญาณเอาต์พุต $u_d(t)$ ของเฟสดีเทกเตอร์
- สัญญาณเอาต์พุต $u_f(t)$ ของลูปฟิลเตอร์
- เฟสเออเรอร์ θ_e กำหนดเป็นความต่างเฟสของสัญญาณ $u_1(t)$ และ $u_2(t)$

เรามาสนใจกับการทำงานทั้งสามบล็อกตามรูป 2-1 ตัว VCO ออสซิลเลตที่ความถี่เชิงมุม ω_2 ซึ่งถูกกำหนดจากสัญญาณเอาต์พุต u_f ของลูปฟิลเตอร์จะออสซิลเลตที่ความถี่เชิงมุม ω_2 กำหนดเป็นสมการดังนี้

$$\omega_2(t) = \omega_0 + K_0 u_f(t) \qquad (2\text{-}1)$$

ที่ซึ่ง ω_0 คือความถี่กลางของ VCO และ K_0 คือเกนของ VCO ในหน่วย $s^{-1}V^{-1}$

ตัว PD หรืออ้างเป็น *เฟสคอมพาราเตอร์* ได้ด้วย จะทำหน้าที่เป็นตัวปรียบเทียบเฟสระหว่างสัญญาณเอาต์พุตกับเฟสของสัญญาณอ้างอิง ซึ่งจะจ่ายสัญญาณเอาต์พุต $u_d(t)$ โดยประมาณเป็นสัดส่วนโดยตรงของเฟสเออเรอร์ θ_e ภายในช่วงจำกัดช่วงหนึ่ง

$$u_d(t) = K_d \theta_e \qquad (2\text{-}2)$$

ที่ซึ่ง K_d แทนด้วยเกนของ PD มีหน่วยเป็นโวลต์ หรือ V/rad สำหรับเหตุผลในการอธิบาย แสดงตามรูปที่ 1-1c เป็นกราฟซึ่งแทนจากสมการที่ 2-2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

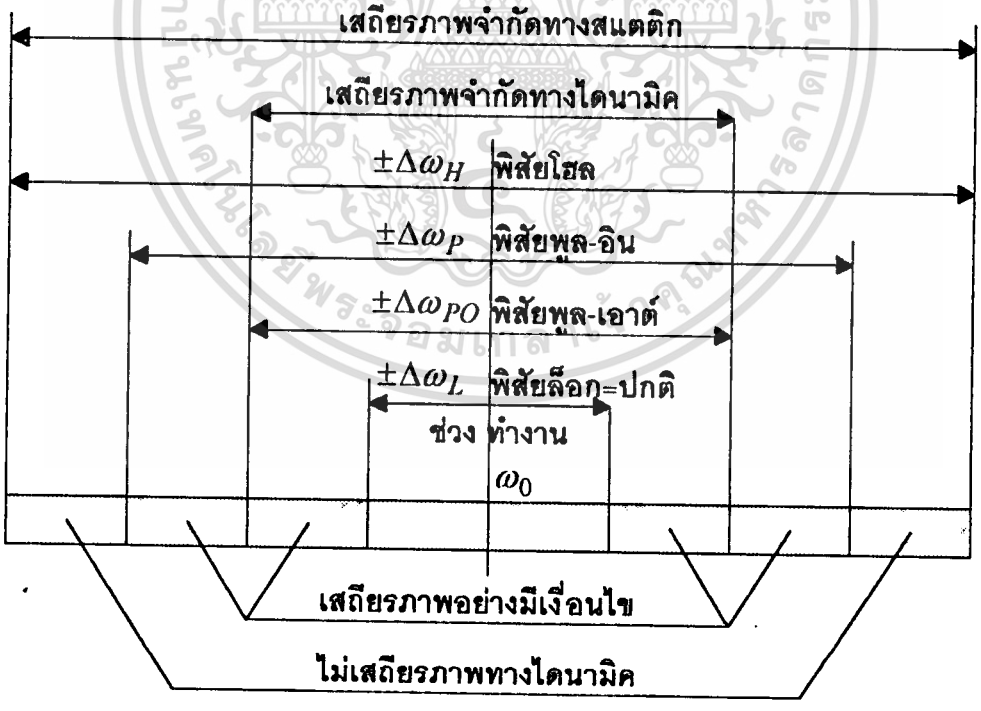
ต่อไป มาดูการทำงานทั้งสามฟังก์ชันบล็อก โดยเริ่มแรกสมมุติให้ความถี่เชิงมุมของสัญญาณอินพุต $u_1(t)$ มีค่าเท่ากับ ความถี่กลาง(centor frequency) ซึ่งแทนด้วย ω_0 แล้วให้ VCO ทำงานที่ความถี่กลาง ดังนั้นจะเห็นว่าเฟสเออเรอร์ θ_e เป็นศูนย์ ซึ่งถ้า θ_e เป็นศูนย์นั้น u_d ของ PD ต้องเป็นศูนย์ด้วย เพราะฉะนั้นสัญญาณเอาต์พุตของลูปฟิลเตอร์ u_f จะเป็นศูนย์ด้วย ซึ่งเงื่อนไขนี้เป็นการให้ VCO ทำงานที่ความถี่กลาง

ถ้าเฟสเออเรอร์ θ_e เริ่มต้นไม่เป็นศูนย์แล้ว PD จะสร้างสัญญาณเอาต์พุต u_d ไม่เป็นศูนย์ หลังจากผ่านลูปฟิลเตอร์เป็นสัญญาณ u_f ด้วยเหตุนี้จะทำให้ VCO เปลี่ยนความถี่ทำงานในทิศทางซึ่งจะทำให้เฟสเออเรอร์หายไป

มีสามเงื่อนไขที่จำเป็นให้ระบบ PLL แบบเชิงเส้นยังคงรักษาการแทรคกิ้ง(tracking)

1. ความถี่เชิงมุมของสัญญาณอ้างอิงต้องอยู่ภายในพิสัยโฮล(hold range)
2. สเต็ปความถี่สูงสุดจ่ายสู่อินพุตอ้างอิงของ PLL ต้องน้อยกว่าพิสัยพูลเอาต์(pull-out range)
3. อัตราการเปลี่ยนของความถี่อ้างอิง $\Delta\omega$ ต้องน้อยกว่า ω_n^2

มีการจัดแบ่งพารามิเตอร์ระบุพิสัยความถี่ในการทำงานของระบบ PLL ซึ่งสามารถสรุปเนื้อหาที่สำคัญได้ดังนี้



รูปที่ 2-2 *โครงสร้างเสถียรภาพทางไดนามิค และสแตติก ของ PLL แบบเชิงเส้น*

1.พิสัยโฮล(hold range) $\Delta\omega_H$ ช่วงความถี่นี้ PLL สามารถรักษาการติดตามอยู่กับเฟสได้ ซึ่ง PLL จะมีเสถียรภาพอย่างมีเงื่อนไขในช่วงนี้เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.พิสัยพูล-เอาต์(pull-out range) $\Delta\omega_{PO}$ เป็นช่วงที่จำกัดไดนามิคสำหรับการทำงาน PLL อย่างเสถียรภาพ ถ้าการแทรคกิ้งสูญเสียไปภายในช่วงนี้ ระบบ PLL จะล็อกอีกครั้ง แต่กระบวนการจะช้าลง ซึ่งเป็นลักษณะของกระบวนการดึงเข้า (pull in process)

3.พิสัยพูล-อิน(pull-in range) $\Delta\omega_P$ คือช่วงที่ระบบ PLL ล็อกเสมอ แต่กระบวนการค่อนข้างช้า

4.พิสัยล็อก(lock range) $\Lambda\omega_L$ คือช่วงความถี่ที่ระบบ PLL จะทำการล็อกภายในหนึ่งบีต(beat) ระหว่างความถี่อ้างอิงกับความถี่เอาต์พุต ซึ่งโดยปกติเป็นช่วงความถี่การทำงานของระบบ PLL ถูกจำกัดภายในช่วงนี้

ดังนั้นในการออกแบบโดยส่วนมาก เราต้องจัดขนาดพิสัยต่างๆไม่เท่ากันดังนี้

$$\Delta\omega_L < \Delta\omega_{PO} < \Delta\omega_P < \Delta\omega_H$$

<u>หมายเหตุ</u> เงื่อนไขประโยชน์ที่ได้รับจะเกี่ยวข้องเฉพาะระบบ PLL แบบเชิงเส้นเท่านั้น กรณีที่เป็นระบบดิจิทัล PLL สภาวะเงื่อนไขนี้จะแตกต่างกันออกไป

ในหนังสืออ้างอิงหลายฉบับจะกล่าวถึง เทอมของ "พิสัยแคปเจอร์(capture range)" ในบางครั้งสามารถใช้ในความหมายเป็น พิสัยพูล-อิน(pull-in range) ซึ่งไม่อาจแยกความแตกต่างกันได้ชัดเจน

2.2 นิยามดิจิทัลเฟสล็อกลูป(Digital Phase-Locked Loop Definition)

ดิจิทัลเฟสล็อกลูปนิยามความหมายได้ยาก เพราะว่าคำว่าดิจิทัลนั้นสามารถประยุกต์ใช้ได้หลายแบบอย่าง ในการกำหนดนิยามของดิจิทัลเฟสล็อกลูปตามแบบดั้งเดิม(classical) จะต้องมีเทอม ซึ่งมีความหมายของสัญญาณอ้างอิง u_1 และสัญญาณเอาต์พุต u_2 เป็นสัญญาณไบนารี โดยเป็นสัญญาณรูปคลื่นสี่เหลี่ยม อย่างไรก็ดี ภายในดิจิทัลเฟสล็อกลูปแบบดั้งเดิม บล็อกแอนะล็อกยังคงถูกใช้อยู่ ดังเช่น ฟิลเตอร์ RC และ VCO ซึ่งเราสามารถกล่าวว่า ระบบ DPLL แบบดั้งเดิม มีคุณลักษณะของสัญญาณแอนะล็อกเป็นตัวกลางอยู่

หลายปีมาแล้วระบบได้มีการพัฒนาสิ่งต่างๆ โดยสร้างจากอุปกรณ์ดิจิทัลอย่างแพร่หลาย ซึ่งจะไม่พบการใช้สัญญาณแอนะล็อก อย่างเช่น ลูปฟิลเตอร์(loop filter) ซึ่งไม่ได้เป็น RC ฟิลเตอร์อย่างที่เป็นมานาน แต่ใช้เป็นดิจิทัลฟิลเตอร์แบบออร์เดอร์หนึ่งหรือออร์เดอร์สองแทน ดังนั้นดิจิทัลเฟสล็อกลูปแบบใหม่จึงต่างจากแบบดิจิทัลแบบดั้งเดิมโดยสิ้นเชิง โดยเรียกรวมทั้งหมดว่า "ระบบเฟสล็อกลูปแบบดิจิทัลล้วน (ADPLL)" จะอธิบายต่อไป ระบบ DPLL ยังได้เพิ่มเติมโดยใช้ซอฟต์แวร์บนไมโครโปรเซสเซอร์ ในขั้นนี้จึงเป็นการง่ายที่จะแยก DPLL แบบดั้งเดิมและ ADPLL ออกจากกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ดิจิทัลเฟสล็อกลูปแบบต่าง ๆกับสัญญาณแอนะล็อกตัวกลาง(Digital PLLs with Intermediate Analog Signals)

ระบบ DPLL แบบนี้ต่างจากระบบ PLL แบบเชิงเส้น ในกรณีการใช้ PD เท่านั้น ซึ่งระบบ PLL แบบเชิงเส้นใช้เฟสดีเทกเตอร์แบบ 1 (ตัวคูณ) อย่างเดียวเท่านั้น โดยที่เกือบทั้งหมดของ DPLL แบบดั้งเดิมใช้เฟสดีเทกเตอร์แบบ 2 (XOR) ,แบบ 3 (J-K) และแบบ 4 ซึ่ง PD แบบ 4 เรียกว่า "เฟส-/ความถี่ดีเทกเตอร์(PFD)" ได้รับความนิยมมากเพราะว่า สามารถกำหนดพิสัยล็อก(lock ranges) ได้กว้างมากตามอำเภอใจ

2.3.1 อิทธิพลของเฟสดีเทกเตอร์แบบต่าง ๆทางด้านไดนามิค (Influence of the Phase-Detector Type on the Dynamic Performance)

สามแบบของดิจิทัลเฟสดีเทกเตอร์คือ

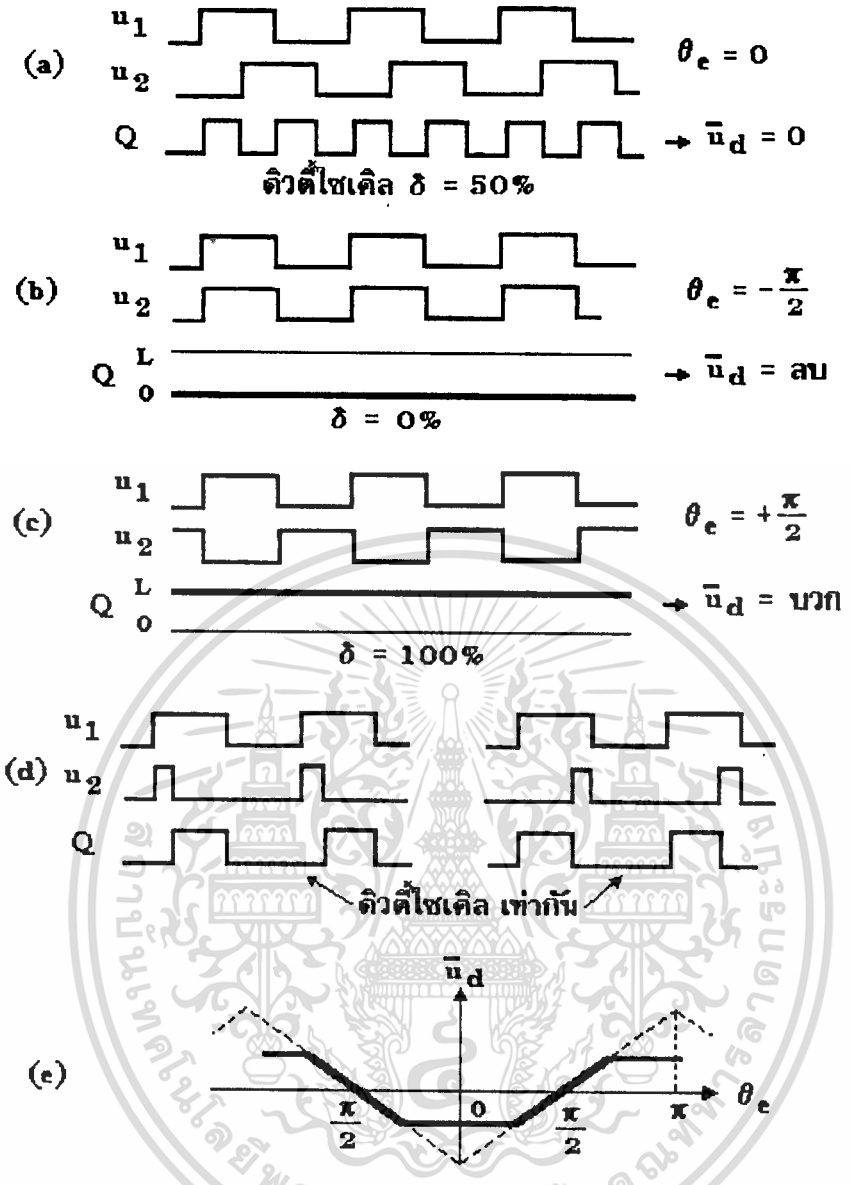
1. เกทเอกซ์คลูสีฟออร์(XOR) หรือ PD แบบ 2
2. ฟลิปฟลอป J-K แบบกระตุ้นขอบ หรือ PD แบบ 3
3. เฟส/ความถี่ดีเทกเตอร์ หริ PD แบบ 4 หรือ PFD

1) เกทเอกซ์คลูสีฟออร์(XOR) หรือ PD แบบ 2

เริ่มด้วย เฟสดีเทกเตอร์แบบที่ง่ายที่สุด คือเกทเอกซ์คลูสีฟออร์(รูปที่ 2-3) วงจรนั้นง่ายมาก และราคาประหยัดแต่ความสามารถไม่ดี เพราะว่าเอาต์พุตของมันจะแปรผันตามค่าดิวตี้ไซเคิส(duty cycle) ของสัญญาณ $u_1(t)$ และ $u_2(t)$ ด้วย ในการใช้งานจริงนั้นจะให้เอาต์พุตเป็นรูปคลื่นสมมาตรเสมอ หากว่ารูปคลื่นอินพุตเหล่านั้นมีเฟสแตกต่างกันเท่ากับ $\pi/2$ ระหว่างสัญญาณอ้างอิง และสัญญาณเอาต์พุตป้อนกลับ โดยที่ทั้งคู่ทำงานที่ความถี่กึ่งกลาง ω_0 ในตำแหน่งนี้จะกำหนดเฟสเออเรอร์เป็นศูนย์คล้ายในเฟสล็อกลูปแบบเชิงเส้น สัญญาณเอาต์พุต $u_d(t)$ ของ PD จะเป็นรูปคลื่นแบบสมมาตรโดยมีความถี่เป็นสองเท่าของสัญญาณอ้างอิง สัญญาณเฉลี่ย $\overline{u}_d$ ให้เป็นศูนย์ซึ่งสัมพันธ์กับครึ่งหนึ่งของแหล่งจ่ายไฟ

ถ้าเฟสแตกต่างกันน้อยกว่า 90 องศาระหว่างสัญญาณ $u_1(t)$ และ $u_2(t)$ ทำให้ดิวตี้ไซเคิลของสัญญาณ $u_d(t)$ จะน้อยกว่า 50 เปอร์เซ็นต์ ซึ่งหมายความว่าสัญญาณเฉลี่ย $\overline{u}_d$ เป็นลบ รูปที่ 2-3 เป็นกรณีที่สัญญาณ $u_1(t)$ และ $u_2(t)$ มีเฟสเดียวกันซึ่งเฟสเออเรอร์ตรงนี้ จะถูกกำหนดเป็น $-\pi/2$ ซึ่งเป็นค่าน้อยที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-3 *การกระทำของ PD แบบ 2*

(a)-(e) แสดงการตอบสนองของ PD เมื่อเฟสเออเรอร์มีค่าต่างๆ

รูปที่ 2-3c ขณะนี้เฟสเออเรอร์จะถูกกำหนดเป็น $+\pi/2$ เป็นค่ามากที่สุดคือและ $\overline{u}_d$ เป็นค่าเฉลี่ยมากที่สุด

ตัวอย่างเช่น PLL ใช้ PD แบบ XOR การทำงานเหมือนกับ PLL แบบเชิงเส้นด้วยเฟสเออเรอร์ θ_e อยู่ในช่วง $-\pi/2 < \theta_e < +\pi/2$

เอาต์พุตเฉลี่ยของ PD ถูกพล็อตเทียบกับเฟสเออเรอร์ θ_e เส้นโค้งที่ได้จะเป็นรูปสามเหลี่ยม ถ้ารูปคลื่น $u_1(t)$ และ $u_2(t)$ สมมาตรกัน กราฟเอาต์พุตค่าเฉลี่ยของ $u_d(t)$ จะได้ตามเส้นโค้งที่เป็นเส้นประตามรูป 2-3e และถ้าสมมุติว่ารูปคลื่น $u_1(t)$ และ $u_2(t)$ ไม่สมมาตรกันแล้วค่าเฉลี่ย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของ $\overline{u}_d$ จะได้ตามเส้นทึบตามรูปที่ 2-3e สังเกตว่าถ้าเป็นรูปคลื่นแบบไม่สมมาตรแล้วจะทำให้ค่าเกน K_d ของ PD ลดลง มีผลทำให้พิสัยโฮล (hold ranges) และพิสัยล็อก(lock ranges)ลดลง ในกรณีนี้คล้ายกับการใช้ PD แบบ 1 ดังนั้นการใช้ PD แบบ 1 จะใช้ประยุกต์กับเฉพาะรูปคลื่นสมมาตรเท่านั้น ถ้าในกรณีรูปคลื่นไม่สมมาตร PD แบบที่ 2 จะปฏิบัติได้เป็นเชิงเส้นกว่า PD แบบที่ 1 ซึ่งสมการสำหรับพารามิเตอร์ต่างๆเช่น พิสัยโฮล พิสัยล็อก พิสัยพูล-อิน(pull in ranges) และ อื่นๆก็ไม่ได้แตกต่างกันมากมายนัก

2) ฟลิปฟลอป J-K แบบกระตุ้นขอบ หรือ PD แบบ 3

ต่อไปพิจารณาถึง PD แบบที่ 3 จากรูปที่ 2-4 ซึ่ง PD แบบนี้ จะใช้การทริกขอบ(edge trigged)ของฟลิปฟลอปแบบ J-K ขอบขาลงของสัญญาณ $u_1(t)$ จะเซตฟลิปฟลอปเข้าสู่สถานะ(state)"1" และสัญญาณขอบขาลงของ $u_2(t)$ จะรีเซตฟลิปฟลอป ดังนี้เพราะว่าฟลิปฟลอปจะตรวจสอบขอบสัญญาณเป็นหลัก จึงไม่มีผลกับดิวตี้ไซเคิล(duty cycles)ของสัญญาณ $u_1(t)$ และ $u_2(t)$ ถ้า PLL ใช้ PD แบบที่ 3 ในการทำงานที่ความถี่กลาง(center freguency) ω_0 สัญญาณ $u_1(t)$ และ $u_2(t)$ จะมีเฟสต่างกัน π(180 องศา) ซึ่งตรงนี้ถูกกำหนดเฟสเออเรอร์ θ_e เป็นศูนย์(ดูรูป 2-4b) สัญญาณเอาต์พุต $u_d(t)$ เป็นรูปคลื่นสี่เหลี่ยมมีดิวตี้ไซเคิล 50 เปอร์เซนต์มีระดับค่าเฉลี่ยแรงดันเท่ากับครึ่งหนึ่ง ซึ่งถูกกำหนดเป็น $\overline{u}_d(t) = 0$

รูปที่ 2-4a เป็นรูปที่แสดงในกรณีที่ $u_1(t)$ และ $u_2(t)$ มีเฟสใกล้เคียงกันอย่างมาก เป็นการเปรียบเทียบรูปคลื่นทั้งสองซึ่งมีเฟสเออเรอร์มีประมาณ $-\pi$ มีดิวตี้ไซเคิลที่เอาต์พุตเกือบจะเป็นศูนย์ ดังนั้นค่าเฉลี่ย $\overline{u}_d(t)$ เป็นค่าลบ

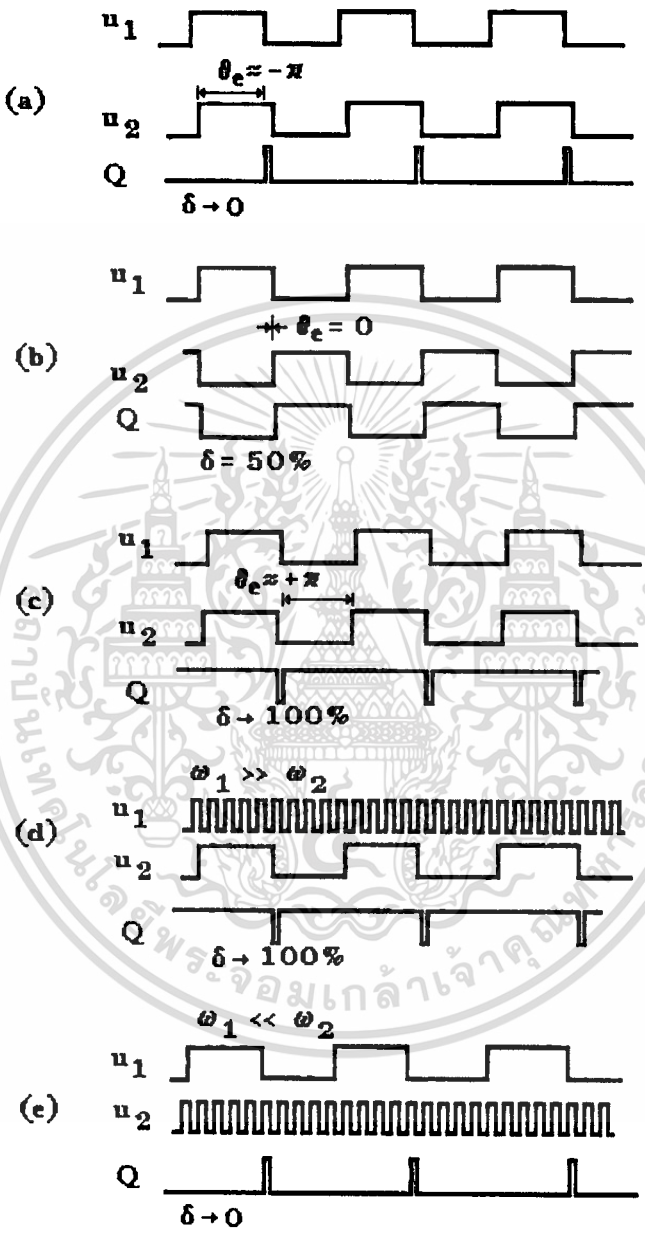
รูปที่ 2-4c เฟสเออเรอร์เป็น π มีอัตราส่วนดิวตี้ไซเคิลเกือบจะ 100 เปอร์เซ็นต์ ดังนั้นค่าเฉลี่ย $\overline{u}_d$ เป็นค่าบวกอย่างมาก ผลการวาดกราฟค่าเฉลี่ยเอาต์พุตของ PD เทียบกับ เฟสเออเรอร์ θ_e จะได้ผลลัพธ์เป็นเส้นโค้งรูปฟันเลื่อย(sawtooth) สำหรับ PD แบบ 3 มีช่วงที่ใช้ประโยชน์ของเฟสเออเรอร์คือ $-\pi < \theta_e < \pi$ ซึ่งมากกว่า PD แบบ 1 และแบบ 2 ถึงสองเท่าดังที่ได้กล่าวมาแล้ว ต่อจากนี้นำมาพิจารณาสถานะที่ซึ่ง PLL ยังไม่มีการล็อกคือ $\omega_1 \neq \omega_2$

ในรูปที่ 2-4d เป็นตัวอย่างแสดงถึงสภาวะที่ $\omega_1 >> \omega_2$ ซึ่งความถี่อ้างอิงมากกว่าความถี่ของ VCO ดังนั้นในหนึ่งสเตต(state)ฟลิปฟลอปถูกเซตมากครั้งกว่าถูกรีเซต สัญญาณเอาต์พุตจะมีดิวตี้ไซเคิลเกือบ 100 เปอร์เซ็นต์ ในลักษณะนี้จะทำให้ความถี่ของ VCO ถูกดึงให้สูงขึ้นตาม ซึ่งเป็นสิ่งที่ต้องการอย่างยิ่ง

อีกกรณีหนึ่งคือ $\omega_1 << \omega_2$ ซึ่งตรงข้ามกับความจริง(ตามรูปที่ 2-4e) ตอนนี้เอาต์พุต PD จะมีดิวตี้ไซเคิลเกือบเป็นศูนย์และความถี่ของ VCO จะถูกดึงลงมาในทิศทางที่ต้องการ เหตุนี้ PD

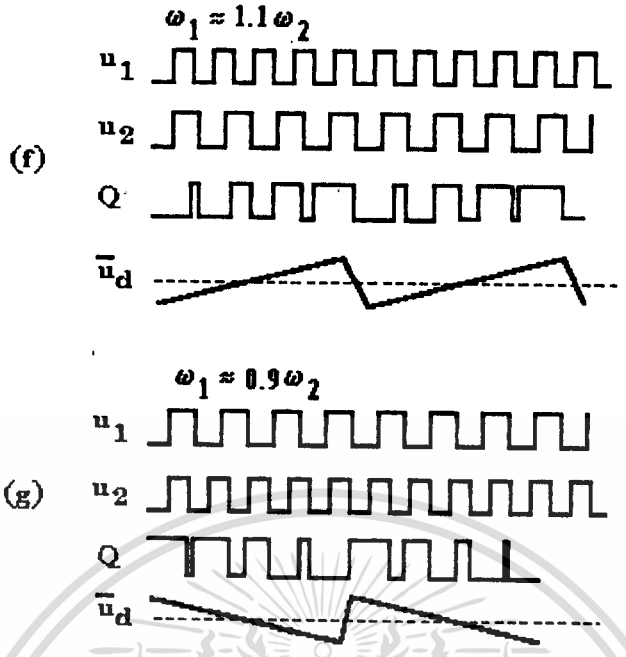
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบ 3 จึงตอบสนองได้ทั้งกระทำเฟสและกระทำความถี่ได้ แต่โชคไม่ดีคุณสมบัติเข้าคู่(duality)ยังไม่สมบูรณ์ดี ซึ่งจะได้แสดงให้เห็นต่อไป



รูปที่ 2-4 *ผลการกระทำของเฟสดีเทกเตอร์แบบ 3 (a)-(g) ความแตกต่าง 7 แบบของออฟเซตของเฟสและความถี่*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-4 *ต่อ*

รูปที่ 2-4f เป็นกรณีที่สองความถี่ ω_1 และ ω_2 ใกล้เคียงกันมาก $\omega_1 = 1.1\omega_2$ อัตราส่วนดิวตี้ไซเคิลของสัญญาณ $u_d(t)$ จะกวาดตั้งแต่ 0 ถึง 100 เปอร์เซ็นต์อย่างต่อเนื่อง และค่าเฉลี่ยของ $\bar{u}_d(t)$ แสดงในรูปคลื่นฟันเลื่อยมีความถี่เท่ากับ $\omega_1 - \omega_2$ สำหรับ $\omega_1 = 0.9\omega_2$ (รูปที่2-3g) มีสถานะคล้ายคลึงกันแต่อัตราส่วนดิวตี้ไซเคิลจะถูกกวาดไปในทิศทางตรงกันข้าม

ค่าเฉลี่ย $\bar{u}_d(t)$ เป็นรูปฟันเลื่อยเช่นกันมีความถี่ประมาณ $\omega_2 - \omega_1$ ในทั้งสองกรณีจะมีค่าเฉลี่ย $\bar{u}_d(t)$ เข้าสู่ศูนย์ที่ซึ่งชี้ได้ว่าพฤติกรรมความไวของความถี่เกือบจะสูญเสียไปหมด ถ้าความถี่ ω_1 และ ω_2 ใกล้กันมาก สำหรับ $\omega_1 = \omega_2$ ของ PD แบบ 3 จะมีพฤติกรรมใกล้เคียงกับ PD แบบที่ 1 มากที่ ที่ซึ่งผลของการดึงเข้าเพื่อล็อก(พูล-อิน) จะมีฮาร์โมนิคไม่พึงประสงค์เกิดขึ้น และไปมอดูเลตกับ VCO ได้

3) เฟส/ความถี่ดีเทกเตอร์ หรือ PD แบบ 4 หรือ PFD

PD แบบ 4 จะเอาใจใส่กับปฏิบัติในทั้งสองอย่าง คือทั้งความถี่ และเฟส โดยไม่ขึ้นอยู่กับอัตราส่วนดิวตี้ไซเคิลรูปคลื่น $u_1(t)$ และ $u_2(t)$ ยิ่งกว่านั้นยังมีความไว(sensitive)สูงเมื่อออฟเซตของความถี่ $(\omega_1 - \omega_2)$มีค่าน้อยมาก เพราะว่าเนื่องจากวงจรของ PD เป็นเหมือนกับตัวตรวจสอบทั้งเฟส และความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดูเหมือนว่าไดอะแกรมรูปแบบใช้อุปกรณ์พื้นฐานเช่นฟลิปฟลอป R-S และตัวแลทช์สองตัวรวมกันซึ่งออกแบบเป็นส่วน UP ส่วน DOWN ตามลำดับ เราต้องจำไว้ว่าแลทช์ทั้งสองแอกทีฟที่ลอจิกต่ำ(low)มากกว่าดังนั้นตัวแลทช์ทั้งสองถูกเซต(หรือรีเซต)โดยขอบขาลงว $u_1(t)$ และ $u_2(t)$ ซึ่งมีการทำงานของ PD แบบ 4 ตามลำดับดังต่อไปนี้

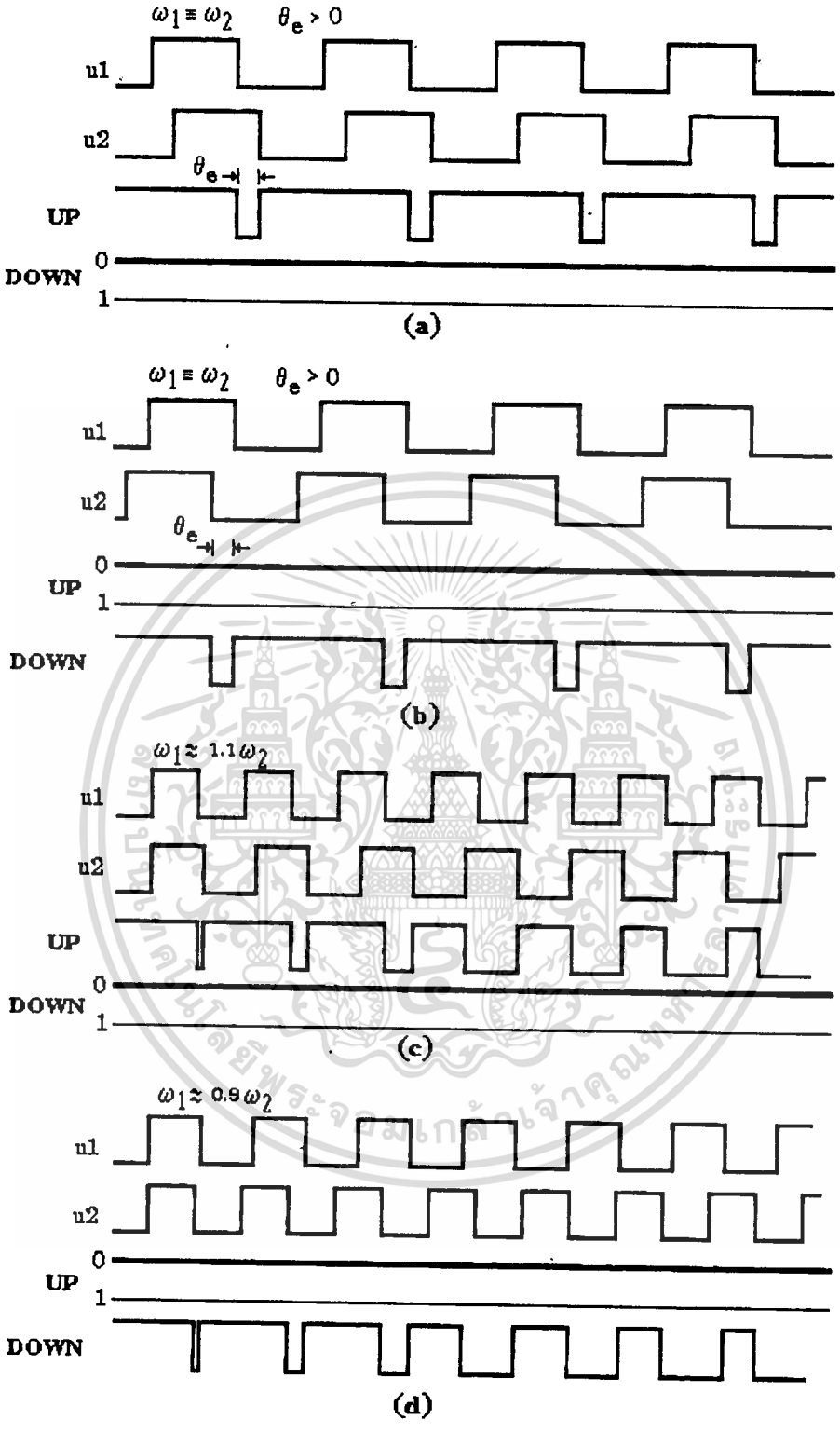
1. ถ้าขอบขาขึ้นของ $u_1(t)$ เกิดขึ้นในขณะที่ $u_2(t)$ เป็นลอจิกต่ำ(low) ขอบขาลงต่อไปของ $u_1(t)$ จะเซตตัวแลทช์ส่วน UP เป็นลอจิกต่ำและขอบขาลงต่อไปของ $u_2(t)$ จะรีเซตตัวแลทช์ตัวนี้เป็นลอจิกสูง(high)
2. ถ้าขอบขาขึ้นของ $u_1(t)$ เกิดขึ้นในขณะที่ $u_2(t)$ เป็นลอจิกสูง แล้วขอบขาลงต่อไปของ $u_1(t)$ จะเซตตัวแลทช์ส่วน DOWN เป็นลอจิกต่ำ และขอบขาลงต่อไปของ $u_2(t)$ จะรีเซตตัวแลทช์ตัวนี้เป็นลอจิกสูง

รูปที่ 2-5 อธิบายการปฏิบัติความไวของความถี่และเฟสของ PD แบบ 4 ในรายละเอียดที่มากขึ้น มีกรณีแตกต่างสี่กรณีถูกพิจารณาคือดังนี้ สองอย่างแรกถูกสมมุติว่า $\omega_1 = \omega_2$ ถ้า $u_2(t)$ มีความถี่เท่ากันกับ $u_1(t)$ แต่ล้าหลัง $u_1(t)$ (ดูรูป 2-5a) เอาต์พุตของตัวแลทช์ UP ถูกทำให้เป็นพัลส์ ซึ่งเป็นสัดส่วนกันเฟสเออเรอร์ θ_e ตัวแลทช์ DOWN จะมีการกระทำสภาวะไม่ทำงาน เหตุนี้ทำให้ความถี่ของ VCO ถูกเพิ่มขึ้นทันที ในขณะนั้นเพื่อว่าเฟสเออเรอร์จะให้ลดลงสู่ศูนย์ ในการปฏิบัติปกติมีเฟสจิตเตอร์(phase jitter)เล็กน้อยอยู่เสมอระหว่าง $u_1(t)$ และ $u_2(t)$ และความถี่ของ VCO ถูกปรับขึ้นลงเพื่อให้ถูกต้องอยู่เสมอ

รูป 2-5b แสดงกรณีที่ $u_2(t)$ กำลังถูกนำหน้า $u_1(t)$ ขณะนี้เอาต์พุตของตัวแลทช์ DOWN ถูกสร้างเป็นพัลส์โดยเป็นสัดส่วนกับเฟสเออเรอร์ θ_e ขณะเดียวกันความถี่ของ VCO จะถูกทำให้ลดลงจนกระทั่งเฟสเออเรอร์เข้าสู่ศูนย์

อีกสองกรณีนั้นแสดงข้อปฏิบัติความไวความถี่(frequency-sensitive) รูปที่ 2-5c แสดงสภาวะที่ซึ่ง $\omega_1 = 1.1\omega_2$ เห็นได้ชัดว่าตัวแลทช์ UP ก็ยังสร้างพัลส์ในตอนนี้และตัวแลทช์ DOWN ยังคงไม่กระทำ และถ้า ω_1 น้อยกว่า ω_2 เหมือนกับ $\omega_1 = 0.9\omega_2$ ซึ่งแสดงไว้ดังรูปที่ 2-5d ถ้าเอาต์พุตของตัวแลทช์ UP และ DOWN ถูกใช้ขับวงจรรวมชาร์จ(charge-IC)หรือเรียกอีกอย่างหนึ่งว่า "วงจรชาร์จปั๊มพ์(charge-pump)" แล้วความถี่ของ VCO จะถูกดึงให้อยู่ในทิศทางที่ถูกต้องเสมอ พิสัยพูล-อิน(pull-in range) พิสัยโฮล(hold range)ของ PLL ซึ่งใช้เป็นช่วงการตรวจสอบเฟสความถี่ต้องมีค่าไม่จำกัดตามทฤษฎี เหตุนี้กระมังจึงไม่เคลือบแคลงสงสัยเลยว่า PD แบบนี้จึงถูกอ้างถึงในการประยุกต์ใช้ PLL แบบดิจิทัลเกือบทั้งหมด สัญญาณเอาต์พุตเฉลี่ย $\overline{u}_d(t)$ ของ PD แบบ 4 นี้ถูกกำหนดโดยค่าเฉลี่ยของอัตราส่วนดิวตี้ไซเคิลของตัวแลทช์ UP และ DOWN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



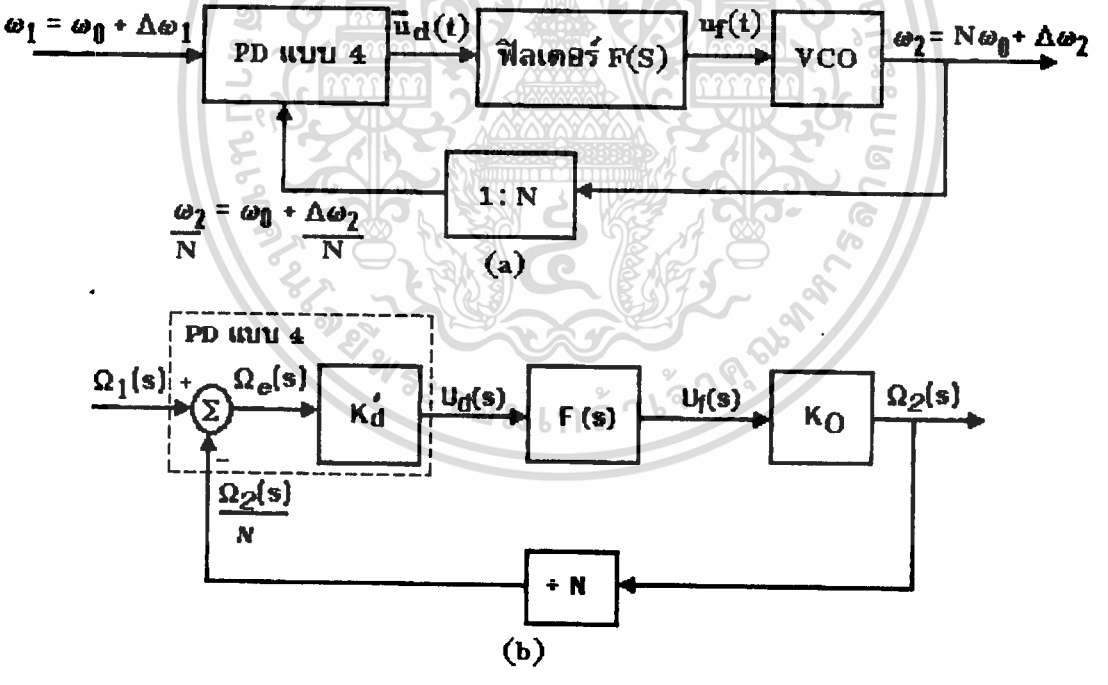
รูปที่ 2-5 *ผลการกระทำของ PD แบบ 4 โดย (a)-(d) แบ่งเป็น 4 กรณีของออฟเซต*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 กระบวนการล็อกของเฟสล็อกลูป(The Lock-In Process of the PLL)

รูปที่ 2-6a แสดงบล็อกไดอะแกรมทั่วไปของ PLL ซึ่งใช้ PD แบบ 4 โดยอาศัยสามฟังก์ชันบล็อกซึ่งรู้จักดี และตัวหารความถี่ถูกรวมเข้าไว้ด้วย ระบบ PLL ตามบล็อกนี้ถูกใช้มากในการสังเคราะห์ความถี่(frequency synthesis) PLL มีการผลิตสัญญาณเอาต์พุตเป็นความถี่ N เท่าของความถี่อ้างอิง ซึ่ง N คือสเกลแฟกเตอร์(scalling factor) ของตัวหารความถี่

ในวงจรรูป 2-6a ไม่เปรียบเทียบโดยตรงของสัญญาณอ้างอิง $u_1(t)$ กับสัญญาณเอาต์พุต $u_2(t)$ แต่จะเปรียบเทียบสเกลที่ลงลงของเอาต์พุตแทน ดังนั้นในขณะล็อกสัญญาณอ้างอิงและความถี่เอาต์พุตไม่จำเป็นต้องเป็นตัวเดียวกันอีก แต่จะแตกต่างกันโดยแฟกเตอร์ N ถ้าเราจะพูดถึงพิสัยโฮล พิสัยล็อก พิสัยพูล-เอาต์ของระบบ อย่างแรกต้องตัดสินใจว่าเมื่อไรพารามิเตอร์เหล่านี้มาสำพันธ์กันอินพุตอ้างอิง หรือเอาต์พุตของ PLL ซึ่งต่อไปการอธิบายจะต้องระบุพารามิเตอร์เหล่านี้ให้สำพันธ์กันอินพุตอ้างอิง



รูปที่ 2-6 ถ้า DPLL ใช้เฟสดีเทกเตอร์แบบ 4 รูปจำลองเชิงเส้นสามารถใช้อธิบายสภาวะไม่ล็อกของระบบได้ (a) บล็อกไดอะแกรมของระบบในโดเมนเวลา (b) รูปจำลองทางคณิตศาสตร์ในระบบในโดเมนความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ที่จริงแล้วสูตรของ PLL แบบดิจิทัลของ PD แบบ 2 และ 3 ไม่ได้แตกต่างไปจาก PLL แบบเชิงเส้นมากมายนัก แต่ PD แบบ 4 มีข้อปฏิบัติทางไดนามิคแตกต่างกันโดยสิ้นเชิง คุณสมบัติที่เป็นหนึ่งเดียวของวงจรเฟสดีเทกเตอร์จะต้องสาธิตให้เห็นและพิจารณาตาม กลับมาที่รูป 2-6a บล็อกไดอะแกรมของ DPLL โดยใช้ PD แบบ 4 เราจะมองเป็นโมเดลทางคณิตศาสตร์เชิงเส้นได้ ก็ต่อเมื่อเป็นสภาวะไม่ล็อก(unlocked) อย่างแรกต้องกำหนดตัวแปรดังนี้

$$\omega_1 = \omega_0 + \Delta\omega_1 \quad = \text{ความถี่อ้างอิง} \tag{2-1a}$$

ซึ่ง ω_O คือความถี่กลาง(center frequency)อ้างเทียบกับอินพุต

$$\omega_2 = N\omega_0 + \Delta\omega_2 \quad = \text{ความถี่เอาต์พุต} \tag{2-2b}$$

ซึ่ง $N\omega_0$ คือความถี่กลางอ้างเทียบกับเอาต์พุต

$$\omega_e = \omega_1 + \omega_2 / N \quad = \text{เออเรอร์ความถี่(frequency error)} \tag{2-3c}$$

ซึ่งอ้างเทียบกับอินพุต

นอกเหนือจากนี้ เราจะต้องกำหนดการแปลงลาปาส(Laplace transform) ของตัวแปรเหล่านั้น ดังนี้

$$\Omega_1(s) = L\{\Delta\omega_1(t)\} \tag{2-2a}$$

$$\Omega_2(s) = L\{\Delta\omega_2(t)\} \tag{2-2b}$$

$$\Omega_e(s) = L\{\Delta\omega_e(t)\} \tag{2-2c}$$

ในสภาวะล็อก($\omega_1 = \omega_2 / N$) สัญญาณเอาต์พุตเฉลี่ย $\overline{u}_d(t)$ ของ PD ขึ้นอยู่กับเฟสเออเรอร์ θ_e

$$\overline{u}_d = K_d\theta_e \tag{2-3}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
20168

คล้องจอง ความสำพันธ์นี้จำกัดอยู่ในช่วง $-2\pi < \theta_e < +2\pi$ ซึ่ง K_d คือ ดีเทกเตอร์เกน K_d เป็นค่าที่มีค่าแน่นอนมากถ้าอ้างถึงตัวเดียวกันดิเทคเตอร์เกนในสภาวะล็อก

ในขณะสภาวะไม่ล็อก $\overline{u}_d(t)$ ถูกแสดงค่าโดยสำพันธ์อยู่กับเออเรอร์ความถี่(frequency error) ω_e สำหรับการประมาณค่าแบบหยาบเราถือว่า $\overline{u}_d$ เป็นสัดส่วนโดยตรงกับเออเรอร์ความถี่ ω_e

$$\overline{u}_d \approx K'_d\omega_e = K'_d\left(\omega_1 - \frac{\omega_2}{N}\right) \qquad (2\text{-}4)$$

นี้คือสมการซึ่งถูกปรับปรุงแล้วอธิบายใน ภาคผนวก ก.

สัมประสิทธิ์ K'_d ในสมการที่ 2-4 ถูกเรียกว่าเป็นเฟสดีเทกเตอร์เกนในสภาวะไม่ล็อก และไม่ได้ปฏิเสธเสียทีเดียวกับสัมประสิทธิ์ K_d ข้างต้น แต่ K'_d มีหน่วยหลายแบบเช่น โวลต์ต่อเฮิรตซ์ โวลต์ต่อวินาที และอื่นๆ ขณะที่ K_d มีหน่วยเป็นโวลต์อย่างเดียว

สามสมการที่ระบุการปฏิบัติไดนามิค(dynamic performance)ของ DPLL คือ

.สำหรับ PD:

$$U_d(s) = K'_d\Omega_e(s) \qquad (2\text{-}5a)$$

.สำหรับลูปฟิลเตอร์:

$$U_f(s) = U_d(s) \bullet F(s) \qquad (2\text{-}5b)$$

.สำหรับ VCO:

$$\Omega_2(s) = K_0U_f(s) \qquad (2\text{-}5c)$$

หลังจากขจัดพจน์ $U_d(s)$ และพจน์ $U_f(s)$ เราได้

$$\Omega_e(s) = \frac{N}{N + K_0K'_dF(s)}\Omega_1(s) \qquad (2\text{-}6)$$

บล็อกไดอะแกรมของโมเดลทางคณิตศาสตร์ของ DPLL ถูกแสดงในรูป 2-6b โมเดลทางคณิตศาสตร์นี้ยอมให้เราคำนวณเออเรอร์ความถี่ ω_e เหมือนเป็นฟังก์ชันของเวลาเพื่อกระจายฟังก์ชันไปประยุกต์ใช้กับอินพุตของ PLL การปฏิบัติในกรณีที่สัญญาณความถี่อ้างอิง ω_1 เป็นค่า-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คงที่แต่ถูกทำเป็นออฟเซตโดยใช้จำนวน $\Delta\omega$ จากความถี่กลาง ω_0 โดยสมมุติได้เลยว่า PLL ไม่ได้ล็อกในขณะเริ่มต้น และความถี่ของ VCO ขณะนั้นเป็น ω_0 ที่ $t = 0$ ดังนั้นเราได้ $\omega_1(t)$ คือ

$$\omega_1 = \omega_0 + \Delta\omega_1 \tag{2-7}$$

สำหรับการแปลงลาปลาสเราได้

$$\Omega_1(s) = \frac{\Delta\omega_1}{s} \tag{2-8}$$

สำหรับลูปฟิลเตอร์แบบนี้ได้อ้างบ่อยๆ และเพื่อความสะดวกจะรวมวงจรชาร์จปั๊มพ์กับ PD แบบ 4 สมการฟิลเตอร์เป็นดังนี้

$$F(s) = \frac{1+s\tau_2}{s\tau_1} \tag{2-9}$$

รวมสมการ 2-6 สมการ 2-8 และสมการ 2-9

$$\Omega_e(s) = \frac{N\tau_1 / K_0K_d'}{1+s\left[\tau_2 + N\tau_1 / K_0K_d'\right]} \frac{\Delta\omega_1}{s} \tag{2-10}$$

การตอบสนองของ $\omega_e(t)$ ของ DPLL หาได้โดยแปลงอินเวอร์สลาปลาสให้อยู่ในรูปโดเมนของเวลา ดังนี้

$$\omega_e(t) = \frac{\Delta\omega_1 N\tau_1}{K_0K_d'} e^{-t/T_P} \tag{2-11}$$

$$T_P = \tau_2 + \frac{N\tau_1}{K_0K_d'} \tag{2-12}$$

สมการที่ 2-11 บอกเราว่า PLL แบบดิจิทัลโดยใช้ PD แบบ 4 ในที่สุดจะสามารถตามล็อกได้ในช่วงออฟเซตความถี่(frequency offset) $\Delta\omega_1$ ของสัญญาณอ้างอิง

สำหรับพิสัยโฮล(hold range)และพิสัยล็อก(lock range)ตามทฤษฎีเราได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\Delta\omega_H = \infty \tag{2-13}$$

$$\Delta\omega_P = \infty \tag{2-14}$$

นี่คือสิ่งที่เราคาดหวังไว้ในตอนต้นสำหรับตัวดีเทกเตอร์เฟส/ความถี่

ในกรณีของ PLL แบบเชิงเส้นค่าพิสัยล็อกเป็นออฟเซตความถี่สูงที่สุดที่ยอมให้ PLL ล็อก ซึ่งจะถูกล็อกภายในหนึ่งซิงเกิลบีต(single-beat) ถ้า PLL แบบเชิงเส้นไม่ล็อกแสดงว่าความถี่ของ VCO ถูกมอดูเลตในทิศทางทั้งลบและบวก(negative and positive) รอบความถี่กลาง ω_0 ถ้าหากค่ายอด(peak)ของความถี่ของ VCO ที่เบี่ยงเบนไปนั้นมีค่ามากเพียงพอแล้ว ระบบจะกลับมาล็อกได้อย่างรวดเร็ว

สภาพการณ์แตกต่างจากกรณีของ PLL แบบดิจิทัล ดังที่วิเคราะห์ไว้แล้วนั้นสัญญาณเอาต์พุตเฉลี่ย $\overline{u}_d$ ไม่เป็นสัญญาณเอซี(AC) ถ้า $\omega_1 \neq \omega_2 / N$ แต่เป็นค่าไฟขั้วบวกเสมอสำหรับ $\omega_1 > \omega_2 / N$ และเป็นค่าไฟขั้วลบเพราะว่า $\omega_1 < \omega_2$ เหตุนี้เองที่ซึ่งความถี่ของ VCO จึงไม่เคยเกิดมอดูเลต รอบความถี่กลางแต่จะพยายามดึงเข้าสู่ความถี่อ้างอิงในทิศทางเดียวเสมอ กล่าวได้อย่างเน้นหนักว่าเป็นไปไม่ได้ที่จะกำหนดพิสัยล็อกของ DPLL ซึ่งกระบวนการล็อกนั้นจะมีได้เสมอ

การออกแบบ DPLL ไม่ได้แตกต่างมากนักกับการออกแบบ PLL แบบเชิงเส้นเพราะว่าพารามิเตอร์เช่น พิสัยโฮล(hold range) พิสัยพูล(pull range) คำนวณง่ายมาก ถ้าเราเลือก PD แบบที่ 2 หรือแบบ 3 จะไม่เผชิญกับบางปัญหาเช่นข้อมูลตัวเลขของพารามิเตอร์ของ K_O และ K_d ซึ่งหาได้ในดาต้าชีต(data sheet) และพารามิเตอร์ K'_d หาไม่พบในดาต้าชีต เพราะว่า K'_d เป็นพารามิเตอร์ใหม่ซึ่งจะกำหนดครั้งแรกในบทความนี้ สมการของ K'_d อย่างง่ายเป็นดังนี้

$$K'_d \approx \frac{U_B}{\omega_0} \tag{2-15}$$

ในสมการนี้ U_B คือแหล่งจ่ายแรงดัน(voltage supply)ของแบบ 4 และ ω_0 เป็นความถี่กลางสำพันธ์กับอินพุต

การบรรยายที่ได้ดำเนินผ่านมานั้น แสดงให้เห็นว่า กระบวนการตอบสนองในการล็อกของ PLL แบบดิจิทัลนั้น ง่ายที่จะทำความเข้าใจกว่า PLL แบบเชิงเส้น การออกแบบวงจรนั้นมีรายละเอียดเพียงพอสำหรับพิสัยโฮล พิสัยพูลและค่าคงที่เวลาตามสมการที่ 2-12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

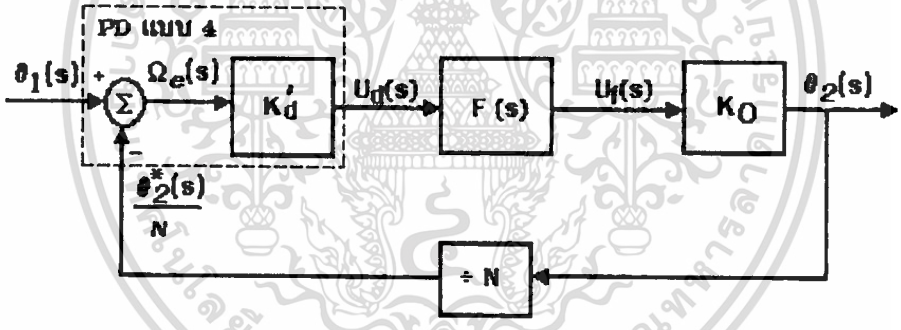
2.3.3 การแทรคกิ้งของ DPLL ในสภาวะล็อก(Tracking Performance of the DPLL on Lock State)

หัวข้อที่แล้วได้เสนอกระบวนการล็อกซึ่งยังคงเหลือการแทรคกิ้งยังไม่ได้พิจารณา ในกระบวนการล็อกนั้นเรามี $\omega_1 = \omega_2 / N$ สมการที่ 2-5a เราสามารถวาดโมเดลทางคณิตศาสตร์ตามรูปที่ 2-7 สำหรับสัญญาณเฟส โมเดลนี้เกือบจะเหมือนกับ PLL แบบเชิงเส้นยกเว้นที่มีตัวหารความถี่ ถ้า $\theta_2(t)$ กำหนดเป็นสัญญาณเอาต์พุตจาก VCO และ $\theta_2^* = \theta_2(t)/N$ คือสัญญาณเฟสที่ออกจากตัวหารความถี่

การแปลงลาปลาสของ $\theta_2(t)$ คือ

$$\Theta_2(s) = L\{\theta_2(t)\}$$

ขณะที่การแปลงลาปลาสของ $\theta_2^*(t)$ โดยอุปมานคือ



รูปที่ 2-7 *โมเดลคณิตศาสตร์แบบเชิงเส้นของ DPLL สำหรับสภาวะล็อก*

$$\Theta_2^*(s) = L\{\theta_2^*(s)\} = \frac{1}{N} L\{\theta_2(s)\}$$

เฟสเออเรอร์(phase error) $\theta_e(t)$ กำหนดในขณะนี้ว่า

$$\theta_e(t) = \theta_1(t) - \frac{\theta_2(t)}{N} \qquad (2\text{-}16a)$$

ขณะที่แปลงลาปลาสเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\Theta_e(t) = \Theta_1(t) - \frac{\Theta_2(t)}{N} \quad (2\text{-}16b)$$

ฟังก์ชันถ่ายโอนของเฟสของ PLL ถูกกำหนดโดย

$$H^*(s) = \frac{\Theta_2(s)}{\Theta_1(s)} \quad (2\text{-}17)$$

ฟังก์ชันถ่ายโอนของเออเรอร์คือ

$$H_e^*(s) = \frac{\Theta_e(s)}{\Theta_1(s)} \quad (2\text{-}18)$$

บนหลักการที่กำหนดไว้ของโมเดลทางคณิตศาสตร์ของรูป 2-7 ใช้สำหรับ $H^*(s)$ และ $H_e^*(s)$ เป็นดังนี้

$$H^*(s) = \frac{K_0 K_d F(s)}{Ns + K_0 K_d F(s)} \quad (2\text{-}19)$$

$$H_e^*(s) = \frac{Ns}{Ns + K_0 K_d F(s)} \quad (2\text{-}20)$$

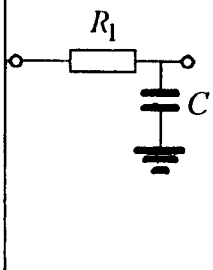
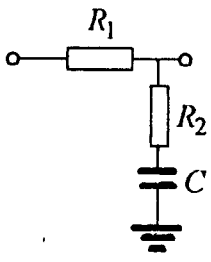
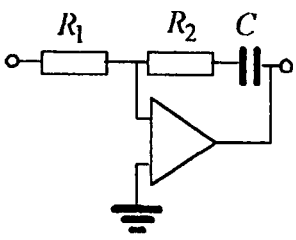
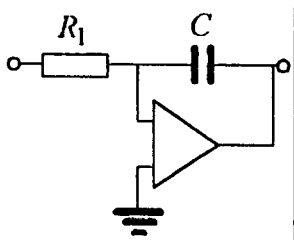
ในกรณีของ DPLL ใช้ PD แบบ 4 ใช้ลูปฟิลเตอร์(loop filter) แบบ 3 (จากตารางที่2-1) โดยปกติเรียกบ่อยๆว่า "ชาร์จปั๊มพ์" สำหรับฟังก์ชันถ่ายโอนของลูปฟิลเตอร์นี้คือ

$$F(s) = \frac{1 + s\tau_2}{s\tau_2}$$

แทนสมการนี้ลงไปในสมการที่ 2-19 และสมการที่ 2-20 เป็นดังนี้

$$H^*(s) = \frac{(K_0 K_d / N\tau_1)(1 + s\tau_2)}{s^2 + s(\tau_2 K_0 K_d / N\tau_1) + (K_0 K_d / N\tau_1)} \quad (2\text{-}21)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พาสซีฟ		แอคทีฟ	
1	2	1	2
			
$\frac{1}{1+s\tau_1}$	$\frac{1+s\tau_2}{1+s(\tau_1+\tau_2)}$	$\frac{1+s\tau_2}{s\tau_1}$	$\frac{1}{s\tau_1}$

ตารางที่ 2-1 โลว์พาสฟิลเตอร์ออร์เดอร์หนึ่งแบบต่างๆ

$$H_e^*(s) = \frac{s^2}{s^2 + s(\tau_2 K_0 K_d / N\tau_1) + (K_0 K_d / N\tau_1)} \qquad (2\text{-}22)$$

เรานำทั้งสองสมการมาหารูปนอร์มอลไลซ์(normalized form) โดยการแทนซึ่งกันและกันเป็นดังนี้

$$\omega_n = \sqrt{\frac{K_O K_d}{N\tau_1}} \qquad (2\text{-}23)$$

$$\zeta = \frac{\tau_2}{2}\sqrt{\frac{K_O K_d}{N\tau_1}} \qquad (2\text{-}24)$$

สำหรับ $H^*(s)$ และ $H_e^*(s)$ แสดงใหม่เป็น

$$H^*(s) = \frac{2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \qquad (2\text{-}25)$$

$$H_e^*(s) = \frac{s^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \qquad (2\text{-}26)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นี่หมายความว่าการแทรคกิ้งของ PLL แบบดิจิทัลเป็นเหมือนกับ PLL แบบเชิงเส้น การตอบสนองไดนามิคของ DPLL ซึ่งสัญญาณอ้างอิงถูกมอดูเลตความถี่ หรือเฟสตามสมการ

ดังนั้นสามารถหาสมการแสดงสำหรับพิสัยพูล-เอาต์(pull-out range) $\Delta\omega_{PO}$ ได้ด้วยเหมือนกับ PLL แบบเชิงเส้น สัญญาณเอาต์พุตเฉลี่ย $u_d(t)$ เป็นฟังก์ชันของเฟสเออเรอร์ θ_e ของ PD แบบที่ 2 คล้ายคลึงกันมากกับแบบที่ 1 เหตุนี้เราสามารถใช้วิธีสมการประมาณค่าคล้ายกัน เพื่อหาพิสัยพูล-เอาต์(pull-out range) ได้



รูปที่ 2-8 พิสัยพูล-เอาต์(pull-out range)ของ PD แบบ 4 ตามฟังก์ชันของ ζ

สำหรับ PD แบบ 3 สัญญาณเอาต์พุตเฉลี่ย $\overline{u}_d(t)$ เป็นฟังก์ชันเชิงเส้นของ θ_e อยู่ในช่วง $-\pi < \theta_e < +\pi$ ระบบจะไม่ทำการล็อกเมื่อไรก็ตามที่เฟสเออเรอร์เกิน π เพราะฉะนั้นพิสัยพูล-เอาต์(pull-out range) สามารถถูกคำนวณได้อย่างเที่ยงตรง ซึ่ง PD แบบ 4 ก็เหมือนกันต่างกันตรงที่เฟสเออเรอร์เป็น 2π แทนที่จะเป็น π

พิสัยพูลเอาต์(pull-out range) โดยใช้ PD แบบ 4 ขึ้นอยู่กับตัวประกอบการหน่วง (damping factor) ζ อย่างแน่นอน รูปที่ 2-8 แสดงพิสัยพูล-เอาต์ของ DPLL กับฟังก์ชันของ ζ จากสมการ

$$\frac{\Delta\omega_{PO}}{\omega_n} = 2\pi \bullet \exp\left[\frac{\zeta}{\sqrt{\zeta^2-1}}\tanh^{-1}\frac{\sqrt{\zeta^2-1}}{\zeta}\right] \quad : \text{for } \zeta > 1 \text{ อ้างจาก [1]}$$

ถ้าเราใช้ PLL สำหรับสังเคราะห์ความถี่ เราจะต้องรู้ความจริงว่าตัวประกอบการหน่วง ζ เป็นฟังก์ชันของอัตราส่วนการหาร(divider ratio) N_O เหมือนสมการที่ 2-23 ถ้าเราใช้อัตราส่วนการ

เอกสารนี้ใช้สำหรับเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หารแปรค่าอยู่ในช่วง $N_{MIN} < N < N_{MAX}$ ดังนั้นอัตราส่วนตัวประกอบการหน่วงระหว่างค่าต่ำสุดกับสูงสุดเป็น

$$\frac{\zeta_{MAX}}{\zeta_{MIN}} = \sqrt{\frac{N_{MAX}}{N_{MIN}}} \qquad (2\text{-}26)$$

ถ้าการเปลี่ยนค่าของ N อยู่ในแฟกเตอร์ 100 แล้วตัวประกอบการหน่วง ζ จะแปรผันด้วยแฟกเตอร์ 10 ที่ซึ่งไม่อาจทนอยู่ได้โดยมีเสถียรภาพ ระบบ PLL จะเป็นอันเดอร์แดมป์(underdamped) หรือโอเวอร์แดมป์ที่ทำให้ช่วงความถี่จำกัด เช่นในการประยุกต์ต่างๆ ค่าคงที่ของเวลาของลูปฟิลเตอร์ต้องสวิทช์ให้คล้องจองกับอัตราส่วนตัวหาร(divider ratio)อย่างทันท่วงที และเพื่อที่จะให้ประสบความสำเร็จต้องประมาณค่าของ ζ อยู่ในช่องระหว่าง 0.5 ถึง 1

2.4 ระบบเฟสล็อกลูปแบบดิจิทัลล้วนและแบบใช้ซอฟต์แวร์เป็นหลัก(All-Digital and Solfware-Based PLLs)

ระบบ ADPLL หรือระบบ PLL แบบดิจิทัลล้วน ซึ่งประกอบด้วยบล็อกของดิจิทัลทั้งหมดโดยไม่รวมบล็อกของแอนะล็อกอีก เทอมคำว่า "ดิจิทัล" ยังคงคลุมเครืออยู่ในแง่การอธิบายประกอบบางตอน ซึ่งไม่ได้มีความหมายว่า เป็นบล็อกฟังก์ชันสัญญาณอินพุต หรือเอาต์พุตอย่างใดอย่างหนึ่งโดยเฉพาะ เช่นเป็นสัญญาณลอจิกสูง(high) หรือลอจิกต่ำ(low)เท่านั้น แต่อีกนัยหนึ่งเป็นสัญญาณแบบเฉพาะ เช่น ไบต์ หรือดิจิทัลเวิร์ดแบบ n บิต(n บิต digital word) ซึ่งแทนผลต่างสภาวะทั้งหมดเป็น 2^n ในหัวข้อต่อไปเราจะเผยให้เห็นลักษณะต่างๆของบล็อกโครงสร้างในระบบ ADPLL

2.4.1 โครงสร้างของบล็อกฟังก์ชันใน ADPLL (Structure of Funtional Blocks in ADPLL)

บล็อกที่สร้างขึ้นของ ADPLLs สามารถมีหลายรูปแบบแตกต่างกัน(ดูรูปที่ 2-9) สังเกตรูปที่ 2-9a สัญญาณไบนารีอินพุตเป็นหนึ่งบิต หรือมากกว่า โดยเอาต์พุตต้องเป็นสัญญาณไบนารีเหมือนกัน ตัวอย่าง เช่น บล็อก PD แบบ XOR ประกอบด้วยสัญญาณไบนารีอินพุตสองอินพุต และสัญญาณไบนารีเอาต์พุตหนึ่งเอาต์พุต บล็อกฟังก์ชันของอินพุตสามารถใช้เป็นไบต์ หรือเป็น n บิตต่อเวิร์ด(word) ดูรูปที่ 2-9b โดยที่สัญญาณเอาต์พุตยังคงเป็นสัญญาณไบนารีหนึ่งบิต หรือ มากกว่า ตามรูปที่ 2-9 สัญญาณไบนารีแทนด้วยเส้นทึบ สัญญาณ n บิตแทนด้วยเส้นคู่

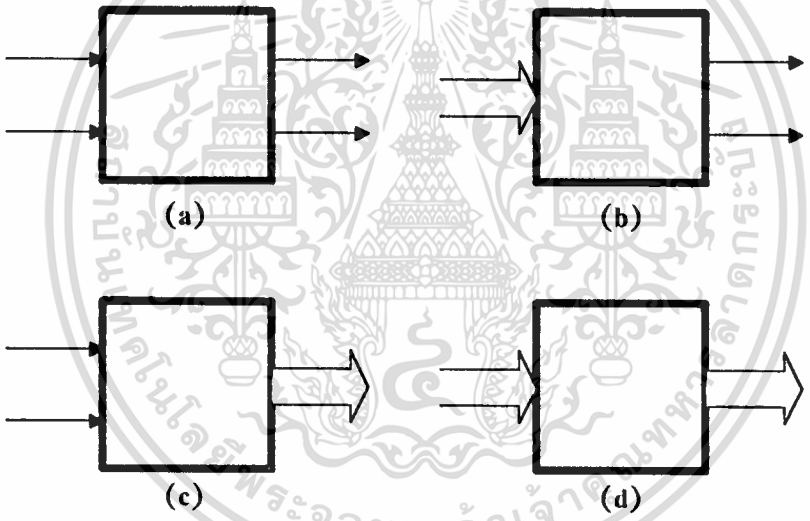
เหมือนแสดงรูป 2-9c แสดงสถานะกลับกันโดยที่อินพุตเป็นสัญญาณไบนารี และเอาต์พุตเป็นเวิร์ด n บิต มีตัวอย่างเหล่านี้เช่นตัวนับ(counter)แบบไบนารี n บิต สัญญาณนาฬิกาที่นับ เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณอินพุตไบนารี คือสัญญาณ UP หรือ สัญญาณ DOWN ดังนั้นความจุของตัวนับคือ เอาต์พุตเวิร์ดแบบ n บิต รูปสุดท้ายสัญญาณอินพุตและเอาต์พุตทั้งสองเป็นสัญญาณ n บิต เวิร์ดดู รูป 2-8c) ตัวอย่างรูปแบบนี้คือ ดิจิทัลฟิลเตอร์ ซึ่งจะอธิบายในหัวข้อต่อไป

การกระทำลอจิก หรือการกระทำคณิตศาสตร์ภายในบล็อกที่สร้างขึ้นตามรูปที่ 2-9 สามารถจะกระทำได้โดยใช้ทั้งฮาร์ดแวร์ และซอฟต์แวร์ ภายใต้การควบคุมของไมโครโปรเซสเซอร์ ตัวอย่างเช่น การคูณสามารถปฏิบัติได้ทั้งฮาร์ดแวร์การคูณ และในอีกทางหนึ่งสามารถปฏิบัติทางซอฟต์แวร์การคูณได้

เนื่องจากอิทธิพลของไมโครโปรเซสเซอร์ ทำให้แนวโน้มของการใช้ซอฟต์แวร์นั้น ได้รับความสนใจมากขึ้น แม้ว่าความเร็วของไมโครโปรเซสเซอร์มากขึ้น แต่มันยังคงมีอุปสรรค์หลักในระบบพื้นฐานของซอฟต์แวร์



รูปที่ 2-9 *โครงสร้างบล็อกฟังก์ชันแบบต่างๆใช้ภายในระบบ ADPLL*

ตัวคูณแบบดิจิทัลโดยใช้ฮาร์ดแวร์ล้วนๆ 16 บิตคูณ 16 บิต ซึ่งถ้าใช้ไอซีสำเร็จรูปอย่างเช่น TTL การปฏิบัติในการคูณนั้นจะใช้เวลาต่ำกว่า 100 ηS เมื่อการทำงานอย่างเดียวกันบนไมโครโปรเซสเซอร์แบบชิปเดียว ตัวโปรแกรมจะใช้คำสั่งประมาณหนึ่งโหลที่จะรันโปรแกรม ถ้าแต่ละคำสั่งใช้เวลาประมาณ 2-3 ηS ดังนั้นจะใช้เวลาทั้งสิ้น 100 μS หรือมากกว่า ดังนั้นการพัฒนา ความเร็วของไมโครโปรเซสเซอร์จะมีผลต่อการเพิ่มความสำคัญในซอฟต์แวร์โปรแกรมของเฟสล็อกลูป

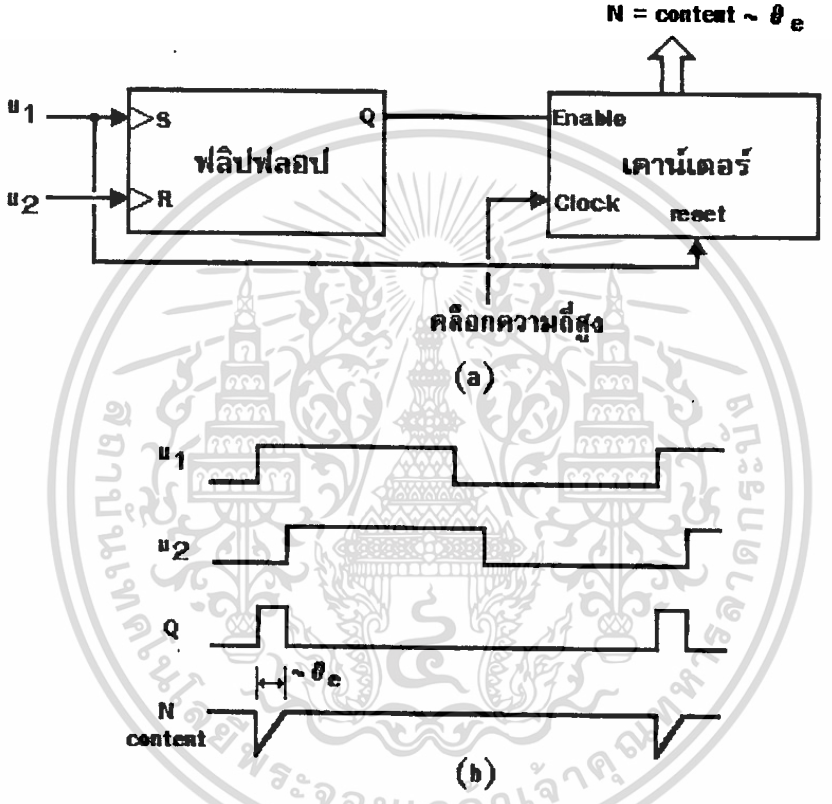
2.4.2 เฟสดีเทกเตอร์แบบดิจิทัลล้วน(All Digital PDs)

เราจำได้ว่าเราได้พิจารณาเฟสดีเทกเตอร์แบบ XOR หรือแบบ 2 แบบฟลิปฟลอป J-K แบบ 3 และดีเทกเตอร์ของเฟสและความถี่แบบ 4 ผ่านมาแล้วคงยังจำได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าเรายอมรับเอารูปแบบสัญญาณอินพุต และสัญญาณเอาต์พุตเป็นแบบเวิร์ด(word) n บิต มาใช้ โดยคิดแผนการสร้างรูปแบบต่างๆของเฟสดีเทกเตอร์ ดังนั้นการออกแบบการสร้างจึงไม่จำกัดการใช้ว่าจะเป็นฮาร์ดแวร์อย่างเดียวหรือซอฟต์แวร์อย่างเดียว

วิวัฒนาการของฟลิปฟลอป J-K ของ PD แบบ 3 ได้เป็นดีเทกเตอร์แบบตัวนับเฟส(FF-counter-phase) แสดงในรูปที่ 2-10a คล้องจองกับรูปคลื่นในรูปที่ 2-10b



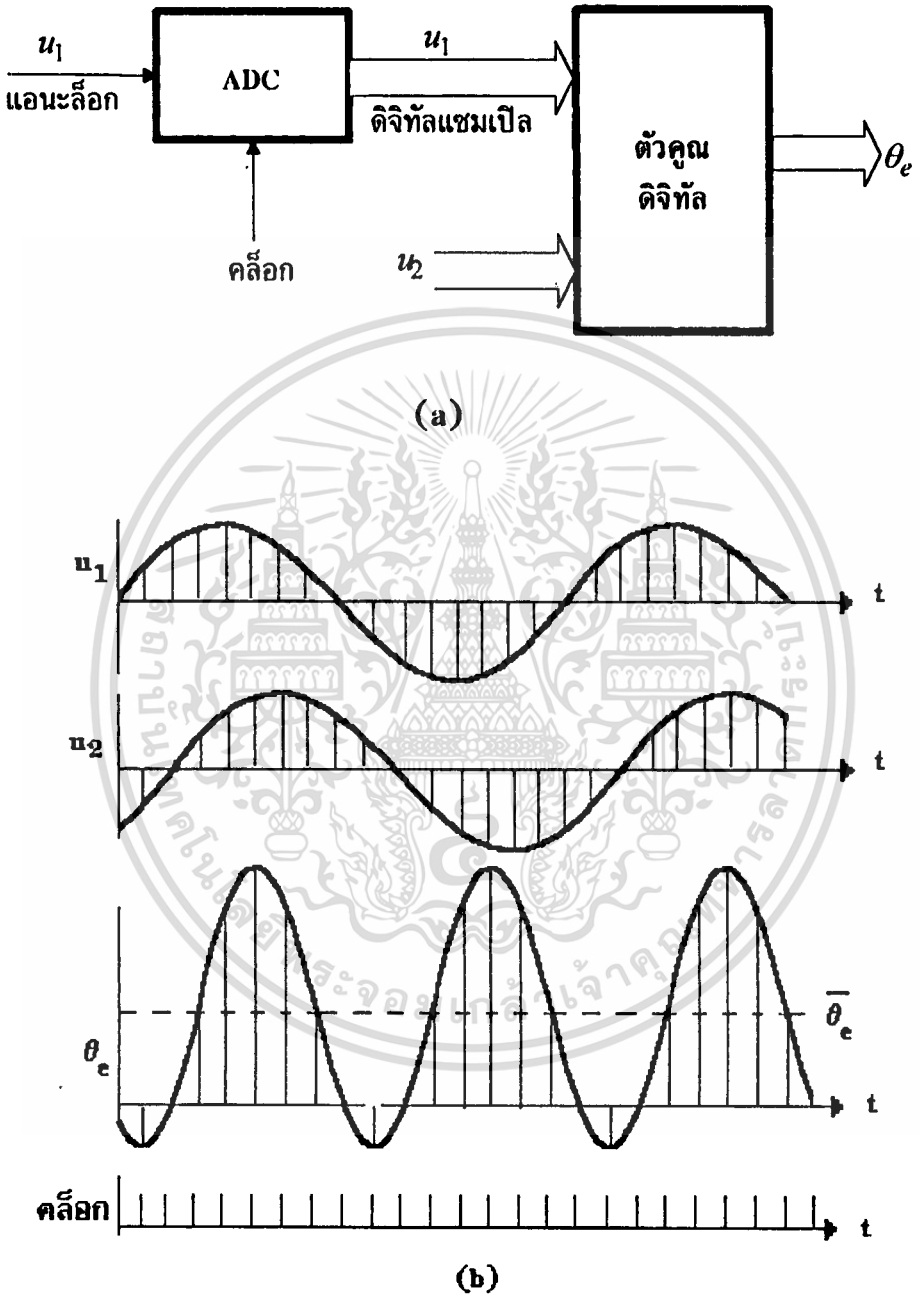
รูปที่ 2-10 *PD แบบฟลิปฟลอปตัวนับ (a) บล็อกไดอะแกรม (b) การตอบสนองรูปคลื่น*

สัญญาณอินพุตอ้างอิง u_1 และสัญญาณเอาต์พุต u_2 ของ VCO หรือ DCO เป็นสัญญาณไบนารี โดยจะเซตหรือรีเซตของฟลิปฟลอปแบบทริกขอบบ คาบเวลาของเอาต์พุต Q เป็นผลจากฟลิปฟลอปซึ่งให้ลอจิกสูง จะเป็นสัดส่วนโดยตรงกับเฟสเออเรอร์ θ_e สัญญาณ Q จะใช้อินาเบิลสัญญาณคล็อกความถี่สูงเข้าสู่ตัวนับแบบนับขึ้น สังเกตได้ว่าตัวนับถูกรีเซตทุกๆขอบขาขึ้นของสัญญาณ u_1

ความจุของ N ของตัวนับเป็นสัดส่วนกับเฟสเออเรอร์ด้วยซึ่ง N ก็คือ เอาต์พุต n บิตของเฟสดีเทกเตอร์ ความถี่ของคล็อกความถี่สูงโดยปกติเป็น $2^M f_O$ ที่ซึ่ง f_O คือความถี่ของสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อ้างอิง และ M คือจำนวนจริงบวก แล้ว 2^M คือจำนวนระดับควอนไตซ์(quantization levels) สำหรับเฟสเออเรอร์ครอบคลุมช่วงของ $0 < \theta_e < 2\pi$

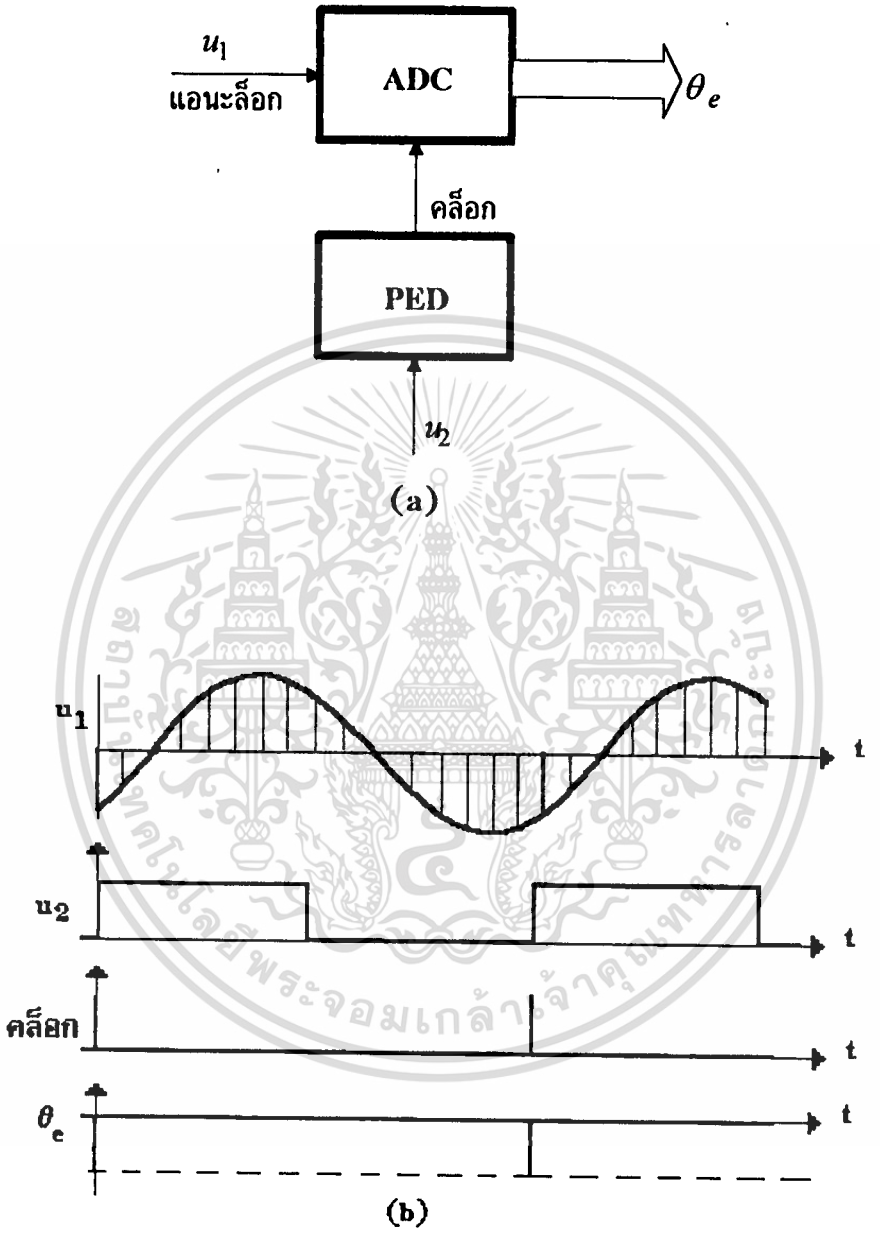


รูปที่ 2-11 *PD แบบ Nyquist-rate (a) บล็อกไดอะแกรม (b) การตอบสนอง*

วงจรเฟสตัวดีเทกเตอร์ดิจิทัลล้วนอีกแบบหนึ่ง คือวงจรนายค์วิสต์เรตเฟสดีเทกเตอร์ (Nyquist-rate NRPD) ชื่อนี้มีกำเนิดมาจากทฤษฎีของนายค์วิสต์(Nyquist) ที่ซึ่ง สเตต(state)จะเกิดขึ้นได้ ก็ต่อเมื่อสัญญาณแซมปลิงต้องมีความถี่เป็นความถี่เป็นสองเท่า ของความถี่สูงสุด ของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความถี่ที่เป็นส่วนประกอบของสัญญาณ บล็อกไดอะแกรมของ NRPD แสดงในรูป 2-11a และคล้องจองกับรูปคลื่นในรูปที่ 2-11b



รูปที่ 2-12 *PD แบบซีโร่ครอสชิง โดยใช้แซมปลิงเฟสเออเรอร์ที่ขอบขาขึ้นของสัญญาณอ้างอิงเท่านั้น (a) บล็อกไดอะแกรม (PED = positive edge detector (b) รูปการตอบสนอง*

สัญญาณอินพุตควรจะเป็นสัญญาณแอนะล็อก ซึ่งสัญญาณถูกส่งมาบนสายข้อมูล มันจะถูกแซมปลิงเป็นคาบ และดิจิไตซ์ในอัตราความถี่ของคล็อก อย่างเช่น อัตราคล็อกถูกเลือกเป็น 16

เอกสารนี้เป็นการเผยแพร่เพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เท่าของสัญญาณความถี่ ซึ่งเป็น 8 เท่าของนายค์วิสต์เรต(Nyquist-rate) สัญญาณ u_2 เป็นสัญญาณเวิอร์ดแบบ n บิต สร้างโดย DCO สัญญาณที่ดิจิไตซ์ได้จาก u_1 และ u_2 ถูกนำมาคูณเข้าด้วยกันโดยใช้โปรแกรมรูทีนการคูณ ดังนั้นการทำงานของ NRPD จะคล้ายคลึงกับแบบเชิงเส้น

ผลลัพธ์ของสัญญาณเฟสเออเรอร์ θ_e แสดงในรูป 2-11b ด้วย ค่าเฉลี่ยของเฟสเออเรอร์ $\overline{\theta}_e$ ถูกฟิลเตอร์ออกไปในดิจิทัลฟิลเตอร์ของกระบวนการต่อไป ซึ่งรายละเอียดจะได้นำเสนอในตอนหน้า

วิธีการวัดเฟสเออเรอร์ยังคงมีอีก เช่น เทคนิคของ ซีโร่ครอสซิง(zero-crossing) ตัวดีเทก-เตอร์แบบซีโร่ครอสซิง(zero-crossing) แบบง่ายที่สุด แสดงในรูปที่ 2-12a สัญญาณอ้างอิง u_1 ถูกสมมุติเป็นสัญญาณแอนะล็อก และ u_2 เป็นสัญญาณไบนารี ขอบขาขึ้นของสัญญาณ u_2 เป็นสัญญาณคล็อกของตัวแปลงแอนะล็อกเป็นดิจิทัล(ADC) ดังนั้น u_1 จะถูกแซมปลิงในทุกๆคาบเวลาอ้างอิง สัญญาณเอาต์พุตของ ADC จะเป็นสัดส่วนโดยตรงกับเฟสเออเรอร์ โดยปกติสัญญาณเอาต์พุตดิจิทัลจะถูกเก็บในเรจิสเตอร์ กระทั่งการแปลงค่าครั้งต่อไปสำเร็จสมบูรณ์ ดังนั้นเฟสเออ-เรอร์ของ θ_e เป็นสัญญาณกึ่งต่อเนื่อง(quasicontinuos) ซึ่งแสดงเป็นเส้นประตามรูปที่ 2-12b

การปรับปรุงรุ่นของเฟสดีเทกเตอร์แบบซีโร่ครอสซิง แสดงในรูป 2-13a กับรูปคลื่นที่คล้องจองในรูป 2-13b ในความแตกต่างระหว่างแบบเดิมโดยสัญญาณ u_2 จะแซมปลิงสัญญาณได้เป็นจำนวนสองเท่าของแบบเดิม และสัญญาณ u_1 ยังคงเป็นแบบเดิม รูปคลื่นถูกแซมปลิงด้วยสัญญาณจากขอบขาลงของ u_2 จะต้องทำคอมพลีเมนต์(หรือกลับขั้ว) ซึ่งจะกระทำโดยวงจรคอม-พลีเมนต์ในรูป 2-13a ด้วยการเพิ่มจำนวนความถี่แซมปลิงเพิ่มขึ้นสองเท่านี้ วงจรสามารถตรวจสอบการเปลี่ยนแปลงของเฟสได้เร็วขึ้น

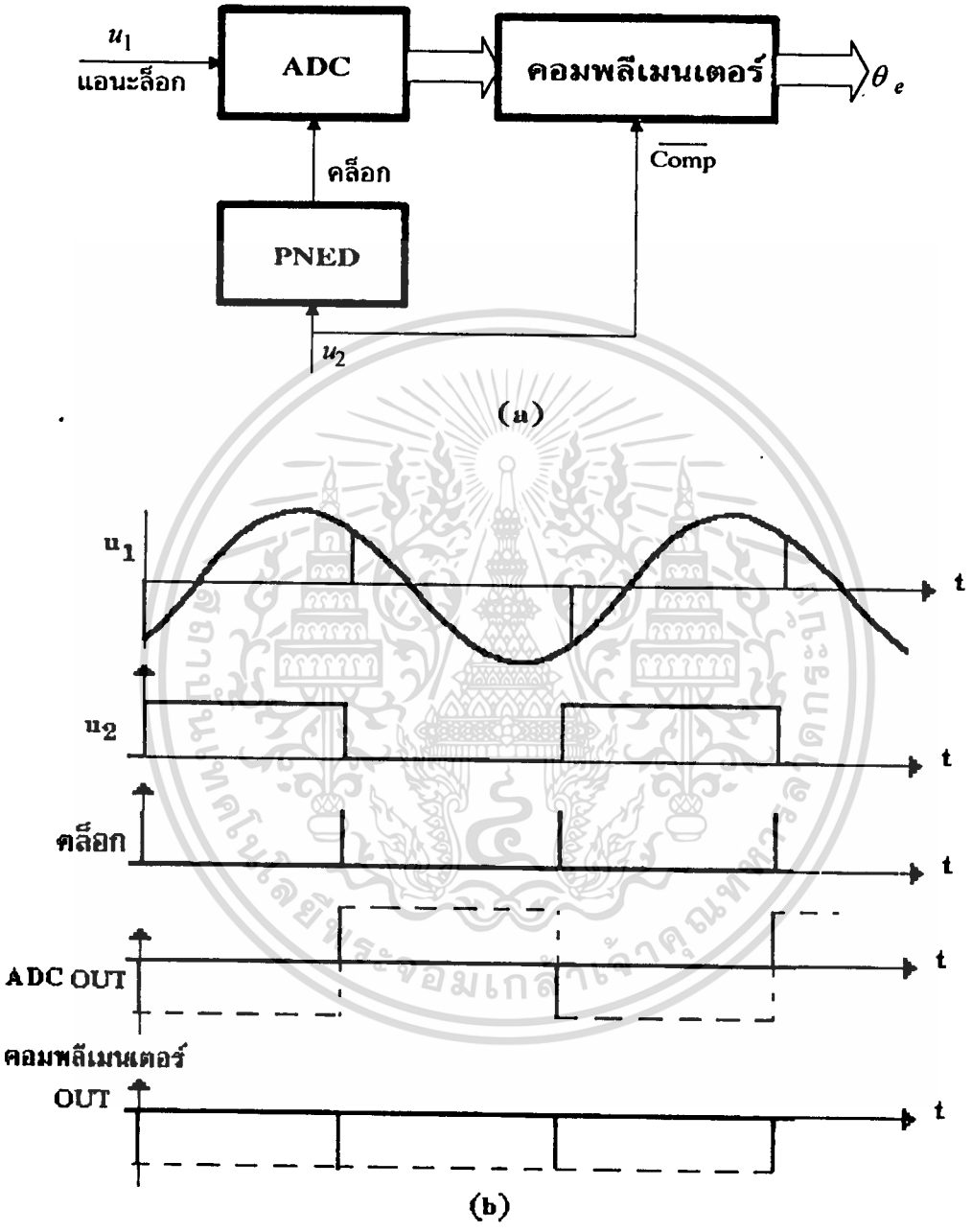
อีกวิธีหนึ่งของการหาเฟสเออเรอร์ ซึ่งได้นำมาใช้โดยเฉพาะเป็น วิธีการแปลงฮิลเบอร์ต (Hibert transform) สมมุติว่าสัญญาณอ้างอิง u_1 ให้เป็นดังนี้

$$u_1 \cos(\omega + \theta_e)$$

ที่ซึ่ง θ_e คือเฟสเออเรอร์ซึ่งไม่ทราบค่า นอกเหนือจากนี้ DCO ที่ใช้ในระบบของเรา ซึ่งถูกสมมุติว่าสร้างสัญญาณเอาต์พุตได้สองช่องเป็น สัญญาณอินเฟส(in-phase) $I = \cos\omega_0 t$ และสัญญาณควอดราเจอร์(quadrature) $Q = \sin\omega_0 t$ รูปการแปลงฮิลเบอร์ตเป็นโพรซีเดอร์(procedure) ทางคณิตศาสตร์อย่างง่ายๆ ซึ่งคำนวณโดยตรงหาค่าเฟสเออเรอร์ θ_e จาก u_1 , I , Q ในการปฏิ-บัติเราต้องการสัญญาณช่วยเหลือ $\hat{u}_1(t)$ ซึ่งคือสัญญาณซึ่งดัดแปลงมาจากสัญญาณ u_1 แต่เฟสเลื่อนไป -90 องศา($-\pi/2$)

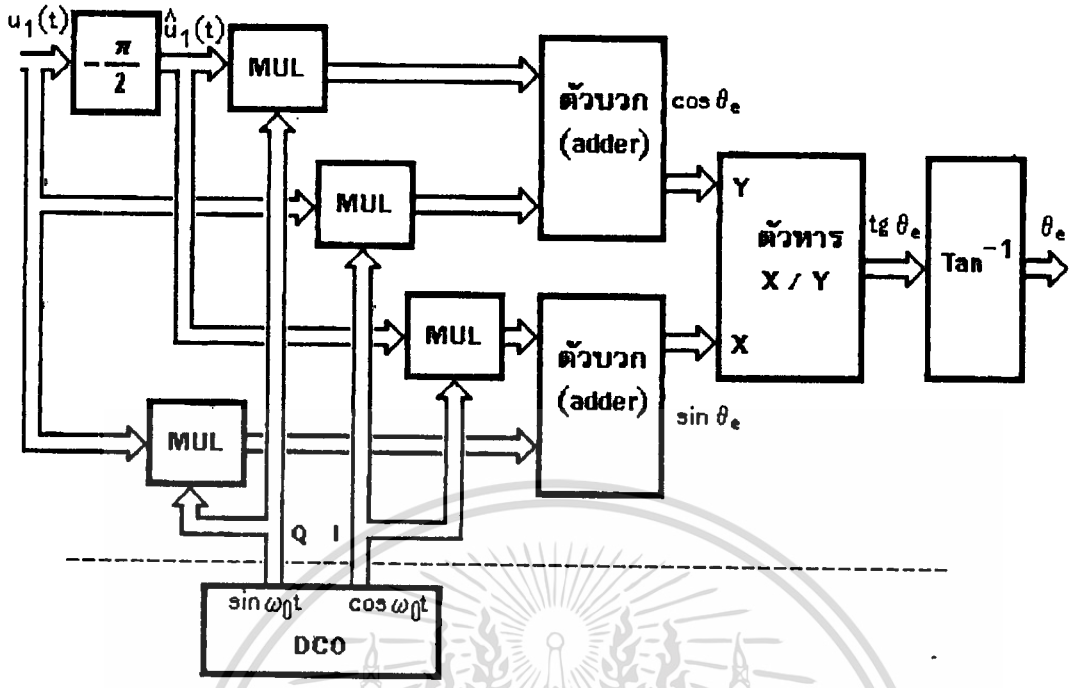
เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

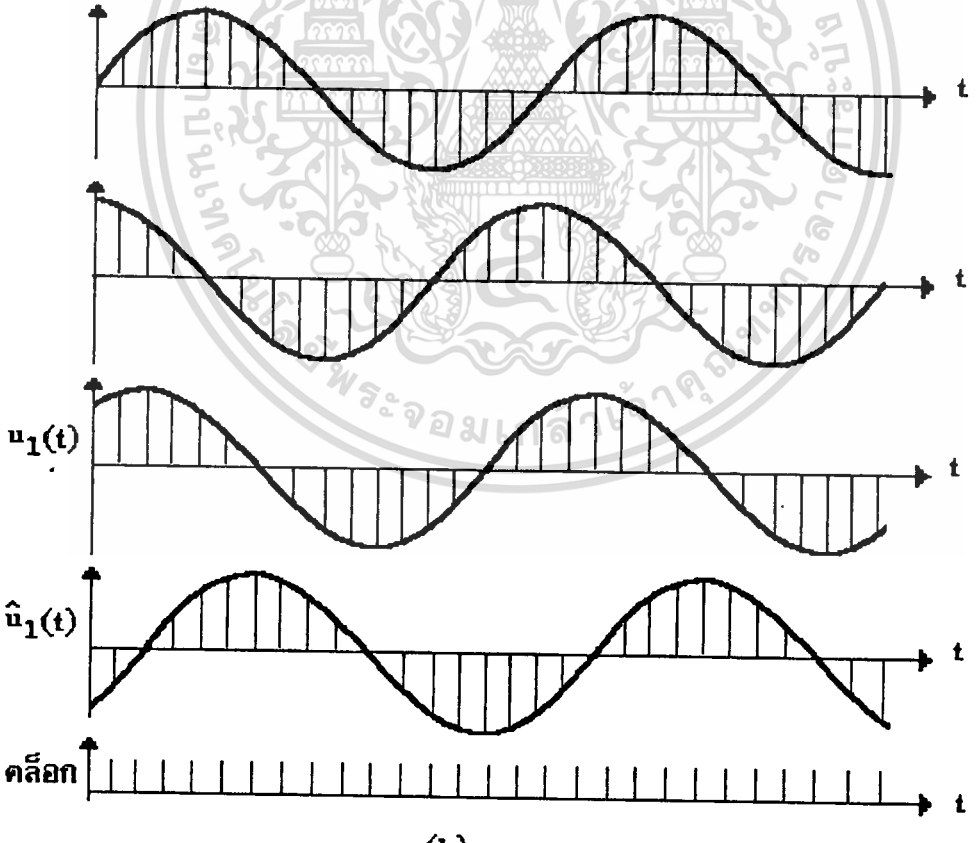
$$\hat{u}_1(t) = u_1 \angle -\pi/2 = \cos(\omega_0 t + \theta_e - \pi/2)$$



รูปที่ 2-13 *PD แบบ ซีโร่ครอสซิง ใช้แซมปลิงเฟสเออเรอร์ทั้งสองขอบของสัญญาณอ้างอิง (a) บล็อกไดอะแกรม (PNED = positive and negative edge detecter) (b) การตอบสนองของรูปคลื่น*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





(b)

รูปที่ 2-14 *PD แบบ การแปลง ฮิลเบอร์ต (a) บล็อกไดอะแกรม (b) การตอบสนอง*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราใช้วิธีทางตรีโกณมิติในการใช้หาเฟสเลื่อนให้ชัดเจนยิ่งขึ้น

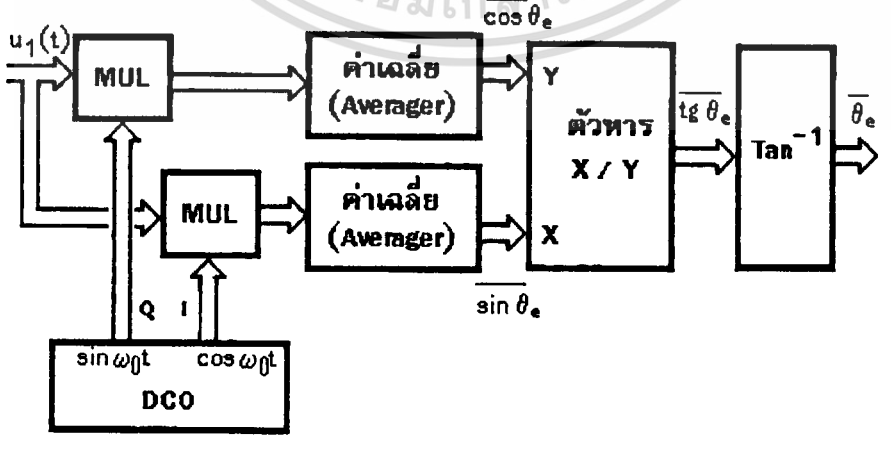
$$\cos\theta_e = Iu_1 + Q\hat{u}_1$$

$$\sin\theta_e = I\hat{u}_1 + Qu_1$$

รูปที่ 2-14 แสดงการกระทำทางคณิตศาสตร์เหล่านั้น สังเกตว่าสัญญาณทั้งหมดภายในระบบถูกสมมุติเป็นข้อมูลเวิอร์ดแบบ n บิต

ตัวหารเราใช้เป็นแทนเจนต์(tangent) ของมุม θ_e แล้วทำอินเวอร์สของแทนเจนต์ ของเฟสเออเรอร์ รูปคลื่นของการแปลงฮิลเบอร์ต(Hilbert transform)ของเฟสดีเทกเตอร์ แสดงตามรูปที่ 2-14b เหมือนใน NRPD การคำนวณทางคณิตศาสตร์ถูกปฏิบัติภายใต้สัญญาณนาฬิกาที่ ซึ่งโดยปกติเป็น 2^M ของสัญญาณความถี่ $f_0 = \omega_0/2\pi$ โดยเฟสเลื่อนที่ต้องการเป็น $-\pi/2$ ที่ซึ่งประมาณค่าโดยการหน่วงสัญญาณ $u_1(t)$ ซึ่งใช้ชิฟท์เรจิสเตอร์(shift register) อย่างเช่นรูป 2-14b สัญญาณนาฬิกาถูกเลือกเท่ากับ 16 เท่าของความถี่สัญญาณ ด้วยเหตุนี้การเลื่อนเฟส $\pi/2$ จะต้องใช้สัญญาณนาฬิกาจำนวน 4 ลูก

มีวิธีซึ่งคล้ายคลึงกันแต่เป็นหนทางที่สะดวกกว่า ในการคำนวณเฟสเออเรอร์ โดยวิธีเฟสดีเทกเตอร์แบบเฉลี่ยดิจิทัล(Digital-averaging phase detector)ตามรูป 2-15



รูปที่ 2-15 *PD แบบเฉลี่ยดิจิทัล(digital-averaging)*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังเช่นกับที่บรรยายผ่านมาโดยต้องการ DCO ให้สร้างสัญญาณอินเฟส I และสัญญาณควอดราเจอร์ Q ตามลำดับ เทอมเหล่านี้มาคูณกับสัญญาณอ้างอิง $u_1(t)$ อีกครั้ง แต่สัญญาณ $\cos\theta_e$ และ $\sin\theta_e$ ถูกใช้ โดยการเฉลี่ยหรืออินทีกรัลสัญญาณเอาต์พุตของตัวคูณครอบคลุมคาบเวลาที่เหมาะสม สังเกตว่าการจัดการแบบนี้พร้อมที่จะรวมฟิลเตอร์ฟังก์ชัน ซึ่งกำหนดโดยฟังก์ชันถ่ายโอนแบบอิมพัลส์(im-pluse transfer function) ของฟิลเตอร์ที่ใช้ วิธีนี้เป็นวิธีที่ยึมหลักของซอฟต์แวร์มาใช้โดยเฉพาะ

รูปแบบทั้งหมดของ PD แบบดิจิทัลที่ซึ่งสามารถออกแบบได้นั้น ได้เปิดเผยให้เห็นเกี่ยวกับหลักสำคัญทั้งหมดอย่างกระทัดรัด แต่ก็ยังไม่สมบูรณ์มากนัก

2.4.3 ดิจิทัลลูปฟิลเตอร์(Digital Loop Filters)

ดังที่กล่าวมาในหัวข้อที่แล้ว PD ต่างๆจะผลิตสัญญาณเอาต์พุตแบบ n บิต โดยที่ซึ่งถ้าเป็น PD แบบง่ายเช่น XOR (PD แบบ 2) หรือเฟส/ความถี่ดีเทกเตอร์(แบบ 4) ซึ่งให้สัญญาณเอาต์พุตเป็นไบนารีหนึ่งหรือสองสัญญาณ เห็นได้แน่ชัดว่าลูปฟิลเตอร์ไม่จำเป็นต้องสอดคล้องกับทุกแบบของ PD แบบดิจิทัลทั้งหมด ซึ่งเราจะพิจารณาเฉพาะลูปฟิลเตอร์ ในแบบที่สามารถเข้ากันได้กับ PD บางแบบ ดังจะอธิบายต่อไป

บางระบบของ ADPLL จะไม่ใช้ลูปฟิลเตอร์(loop filter) ที่ซึ่งโดยปกติจะเกิดขึ้นในระบบ PLL แบบดิจิทัลแบบออร์เดอร์หนึ่ง อย่างไรก็ดีในกรณีเฟสเออเรอร์ค่อยๆเปลี่ยนแปลง ซึ่งจะไม่ต้องการใช้ลูปฟิลเตอร์แบบออร์เดอร์หนึ่งอีก แต่ถ้าเป็น DCO ซึ่งเป็นผลกับเฟสเออเรอร์ได้ดีกว่าได้นำมาใช้กับในระบบ ADPPL เกือบทุกกรณี

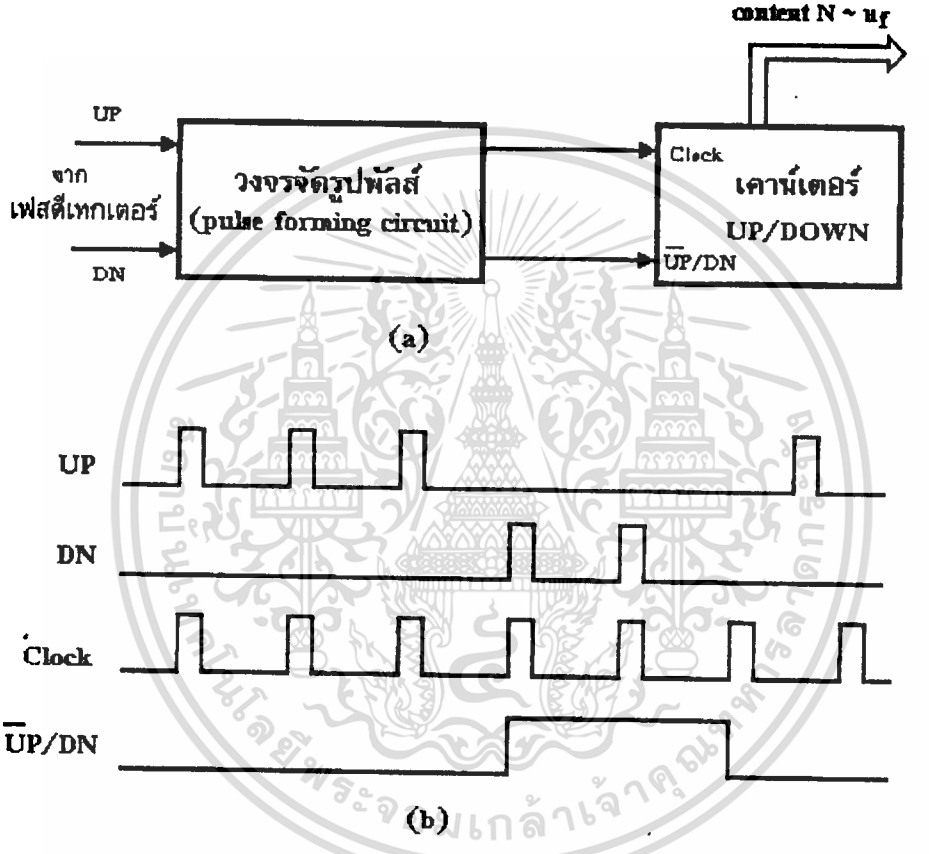
ลูปฟิลเตอร์อย่างง่ายที่สุดสร้างจากตัวนับแบบขึ้นลง(UP/DOWN counter) แสดงดังรูปที่ 2-16 ลูปฟิลเตอร์แบบตัวนับ ซึ่งทำงานโดยอ้างอิงจากขบวนพัลส์ซึ่งส่งมาโดยเฟสดีเทกเตอร์ดังเช่น PD แบบ 4 เป็นต้น อย่างไรก็ดีสามารถปรับปรุงให้ใช้กับ PD แบบ XOR หรือฟลิปฟลอป J-K อย่างง่ายดาย ดังรูป 2-16a ขบวนพัลส์อย่างแรกต้องเปลี่ยนพัลส์ UP และ พัลส์ DOWN เป็นสัญญาณนาฬิกาและสัญญาณกำหนดทิศ(ขึ้น-ลง)จากรูปที่ 2-16b

ในแต่ละพัลส์ UP ซึ่งถูกสร้างโดยเฟสดีเทกเตอร์ทำให้ตัวนับมีความจุ N บิต ถูกเพิ่มขึ้นครั้งละ หนึ่ง เช่นเดียวกับพัลส์ DOWN จะลดความจุ N ความจุ N ถูกส่งออกจากตัวนับเป็นสัญญาณเอาต์พุตแบบขนาน n บิต คือ u_f ของลูปฟิลเตอร์ เพราะว่าความจุ N คือน้ำหนักของการนับจากพัลส์ UP และ DOWN ถ้าพิจารณาแบบหยาบๆลูปฟิลเตอร์ ก็คือฟังก์ชันถ่ายโอนในเทอมอินทีกรัลนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$H(s) = \frac{1}{sT_i}$$

ซึ่ง T_i ค่าคงที่เวลาของอินทีกรัล อย่างไรก็ดี นี่คือการประมาณค่าอย่างคร่าวๆซึ่งพัลส์ UP และ พัลส์ DOWN ไม่ได้ถือเป็นข้อมูลของเฟสเออเรอร์ที่แท้จริง แต่เป็นเพียงเพื่อให้รู้ว่าสัญญาณ $u_1(t)$ นำหน้าหรือตาม หลังสัญญาณ $u_2(t)$



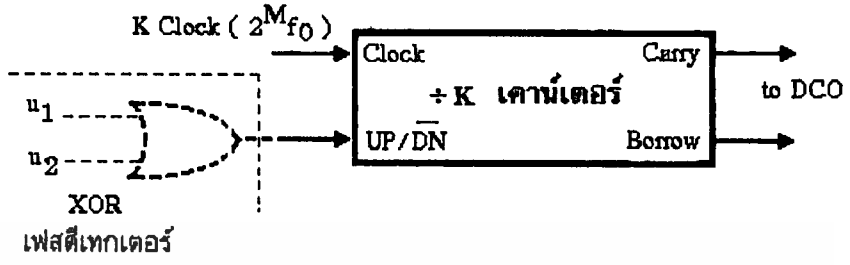
รูปที่ 2-16 *ลูปฟิลเตอร์แบบตัวนับ UP-DOWN (a) บล็อกไดอะแกรม (b) การตอบสนอง*

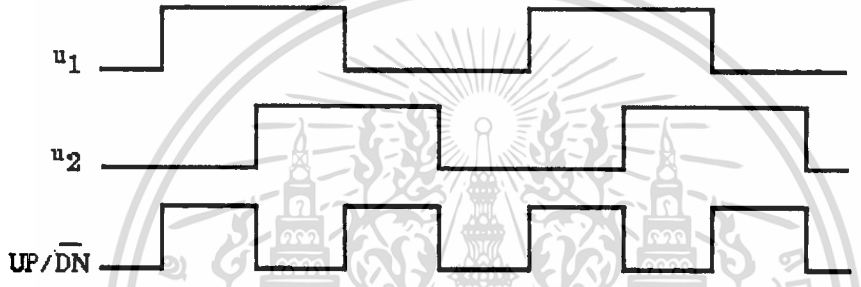
ซึ่งคล้ายคลึงกับลูปฟิลเตอร์แบบดิจิทัลตามรูป 2-17 เป็นตัวนับ K (K counter) ซึ่งสอดคล้องกับ PD แบบ XOR และฟลิปฟลอปแบบ J-K โดยปกติตัวนับ K จะนับยอนกลับซึ่งค่า K คือสเกลแฟกเตอร์(scale factor)สามารถเซตจากสัญญาณควบคุม(K modulus control) สัญญาณอินพุตนาฬิกาโดยปกติจะเป็นรูปคลื่นสี่เหลี่ยมมีความถี่ $2^M f_O$ ซึ่งความถี่ f_O คือความถี่ของสัญญาณอ้างอิงอย่างเช่น PD แบบ XOR จะใช้ควบคุมทิศทางการนับเป็นต้น

ถ้าระบบการทำงานปราศจากเฟสเออเรอร์ 90 องศาจากสัญญาณ u_1 และสัญญาณ u_2 สัญญาณเอาต์พุตจะเป็นรูปคลื่นแบบสมมาตร ด้วยเหตุนี้ ตัวนับจะนับขึ้นและนับลง ในระหว่างช่วง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เวลาที่เท่ากัน ถ้า K เซตไว้มากเพียงพอตัวนับไม่เกิดโอเวอร์โฟลว์และอันเดอร์โฟลว์ เนื่องจากถ้านับขึ้นต้องเตรียมความจุในการสตาร์ตเป็นศูนย์เป็นต้น





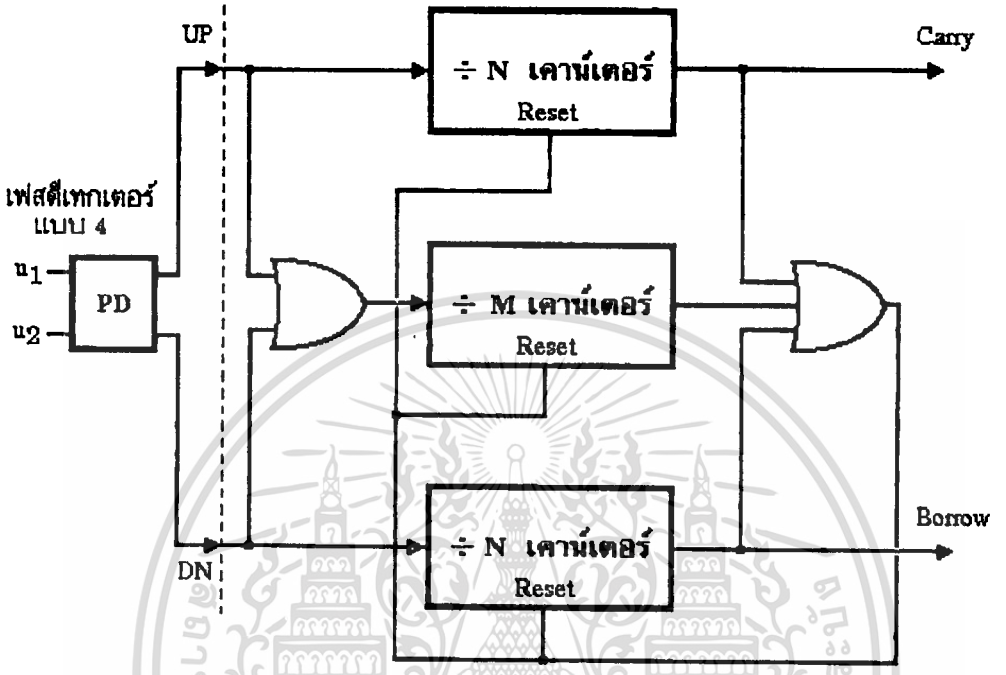
รูปที่ 2-17 *ลูปฟิลเตอร์แบบตัวนับ K (a) บล็อกไดอะแกรม (b) การตอบสนอง*

ถ้าสัญญาณ u_1 สตาร์ตนำหน้าสัญญาณ u_2 สัญญาณเอาต์พุตจาก XOR จะกลายเป็นไม่สมมาตร และตัวนับ(counter)มีช่วงเวลานับขึ้นมากกว่านับลง ซึ่งผลลัพธ์ตัวนับจะโอเวอร์โฟลว์ ซึ่งขณะนั้นจะสร้างพัลส์ทด(carry pulse) ในทางตรงกันข้ามกันกรณีที่ u_1 ล้าหลัง u_2 ตัวนับจะผลิตพัลส์ยืม(borrow pulse) ซึ่งพัลส์ทดและพัลส์ยืมสามารถใช้เป็นสัญญาณควบคุม DCO โดยการบวกหรือการลบพัลส์จากความถี่ของออสซิลเลเตอร์ทุกๆพัลส์ยืมและพัลส์ทด อีกทางเลือกหนึ่งพัลส์ยืมและพัลส์ทดยังสามารถใช้เพิ่มขึ้น และลดลงของค่าสเกลแฟกเตอร์(scale factor)ซึ่งตัวหารความถี่ได้ด้วย โดยที่อินพุตยังคงเป็นสัญญาณนาฬิกาคงที่ด้วย

ไม่ใช่เป็นสิ่งที่ง่ายในการกำหนดฟังก์ชันถ่ายโอน $H(s)$ สำหรับลูปฟิลเตอร์แบบนี้เพราะมันทำงานไม่เป็นเชิงเส้น เฟสเออเรอร์เท่ากับศูนย์มันจะไม่ผลิตสัญญาณเอาต์พุตเลย สำหรับเฟสเออเรอร์เปลี่ยนแปลงคงที่ก็จะผลิตพัลส์ทด(carry) และพัลส์ยืม(borrow)เป็นความถี่คงที่ ถ้าพัลส์เหล่านี้ถูกใช้สร้างพัลส์บวกและพัลส์ลบโดยออสซิสเลเตอร์ความถี่คงที่ ซึ่งตัวนับ K จะปรากฏเหมือนกับบล็อกค่าเกนคงที่มากกว่าที่จะเป็นฟิลเตอร์ ถ้าพัลส์ทดและพัลส์ยืมถูกใช้เป็นสเกลแฟกเตอร์ของตัวนับลง(down counter)กับสัญญาณนาฬิกาคงที่แล้ว การทำงานจะเป็นแบบอินทีกรัลหยาบๆ ถ้ากล่าวทางคณิตศาสตร์แล้วอุปกรณ์นี้สามารถเพิจารณาเป็นฟิลเตอร์ได้ในทุกกรณี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอาต์พุตของฟิลเตอร์แบบ N-before-M สามารถถูกใช้คล้ายคลึงในทางเดียวกับ DCO ซึ่งใช้ K แฟกเตอร์



รูปที่ 2-18 บล็อกไดอะแกรมของลูปฟิลเตอร์แบบ N-before-M

อุปกรณ์อีกแบบหนึ่งคือเป็นลักษณะซึ่งเรียกว่าฟิลเตอร์แบบ N-before-M ตามรูปที่ 2-18 การปฏิบัติของฟิลเตอร์แบบนี้จะไม่เป็นเชิงเส้นมากๆ จากรูปที่ 2-18 แนะนำได้ว่าการทำงานของฟิลเตอร์แบบ N-before-M ในรอยต่อระหว่างเฟสดีเทกเตอร์ซึ่งกำลังส่งพัลส์ UP กับพัลส์ DOWN เหมือนกับกรณีของเฟสดีเทกเตอร์แบบ 4 ฟิลเตอร์แบบ N-before-M ใช้สองความถี่เพื่อให้ตัวนับนับสัญญาณอินพุตแบบนับลงโดยใช้แฟกเตอร์ N และตัวนับอีกตัวนับสัญญาณอินพุตแบบนับลงเช่นเดียวกัน แต่ใช้แฟกเตอร์ M ที่ซึ่ง M > N เสมอ ตัวนับที่ใช้แฟกเตอร์ M นั้นจะนับสัญญาณพัลส์ UP และ DN แสดงดังรูป 2-18 ตามบล็อกไดอะแกรมตัวนับแฟกเตอร์ N ตัวบนจะผลิตพัลส์ทด(CARRY)ตัวหนึ่งเมื่อรับพัลส์อินพุตมาจำนวน N พัลส์ แต่มันจะผลิตพัลส์ทด(CARRY) ก็ต่อเมื่อตัวนับแฟกเตอร์ M รับพัลส์มาไม่มากกว่า M พัลส์ มิฉะนั้นตัวนับแฟกเตอร์ N จะรีเซต เราสามารถกล่าวได้ว่าตัวนับแฟกเตอร์ N ตัวบนจะผลิตพัลส์ทด(Carry pulse) เมื่อไรก็ตามที่พัลส์ N เข้ามากกว่าพัลส์ M ของพัลส์ UP หรือเมื่อไรก็ตามพัลส์ซึ่งนำเข้ามาเป็นหลัก คือพัลส์ UP คล้ายคลึงกันสภาวะการณ์ของรูป 2-18 ที่ทำให้ตัวนับแฟกเตอร์ N มีค่าต่ำกว่า M ที่ซึ่งเอาต์พุตจะได้เป็นพัลส์ยืม(borrow pulse)เท่านั้น หรือเมื่อพัลส์นำเข้ามาเป็นหลักคือพัลส์ DN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อไปนี้เราได้เกี่ยวข้องกับลูปฟิลเตอร์สอดคล้องกับเอาต์พุตแบบ N บิตแบบขนานหรือแบบเวิร์ด การแก้ปัญหาเพื่อหาคำตอบของกรณีลูปฟิลเตอร์แบบดิจิทัลนี้ ซึ่งทำงานด้วยตัวมันเองกับอินพุต N บิต และสัญญาณเอาต์พุต N บิต ซึ่งในบางกรณีลูปฟิลเตอร์แบบดิจิทัลตอ้งการฟังก์ชันถ่ายโอนแบบลูปฟิลเตอร์แบบเชิงเส้น อย่างเช่นฟังก์ชันถ่ายโอนแบบหนึ่งโพล(single-pole) ของโลว์พาสฟิลเตอร์(low pass filter)สามารถเขียนเป็นดังนี้

$$F(s) = \frac{1}{1+sT_1}$$

ถ้าฟิลเตอร์เช่นนี้ได้สร้างเป็นดิจิทัลฟิลเตอร์ ซึ่งเป็นอิมพัลส์ฟังก์ชันถ่ายโอนหรือการแปลง Z (z-transform)สามารถถูกพบก่อน ในกรณีของโลว์พาสฟิลเตอร์แบบออร์เดอร์หนึ่งสามารถหาได้เป็น

$$F(z) = \frac{T/T_1}{1-z^{-1}e^{-T/T_1}} \qquad (2\text{-}27)$$

ซึ่ง z^{-1} คือดีเลย์โอเปอร์เรเตอร์(delay operator)คล้องจองกับ

$$z^{-1} = e^{-sT}$$

และ T คือคาบเวลาแซมปลิงใช้สำหรับดิจิทัลฟิลเตอร์

การใช้การแปลง z (z-transform) $U_f(z)$ ในสมการที่ 2-27 ของสัญญาณเอาต์พุตของลูปฟิลเตอร์ $u_f(t)$ ได้แก่

$$U_f(z) = U_d(z)\frac{T/T_1}{1-z^{-1}e^{-T/T_1}} \qquad (2\text{-}28)$$

$$= U_d(z)\frac{a}{1-bz^{-1}}$$

ที่ซึ่ง $U_d(z)$ คือ z-transform ของสัญญาณเอาต์พุตของเฟสดีเทกเตอร์ $u_d(t)$ สัมประสิทธิ์ฟิลเตอร์ถูกแสดงโดย a และ b เพื่อทำให้ง่ายขึ้น ที่ซึ่ง $a = T/T_1$ และ $b = \exp(-T/T_1)$

แปลงสมการที่ 2-28 กลับเข้าสู่โดเมนเวลาได้แก่

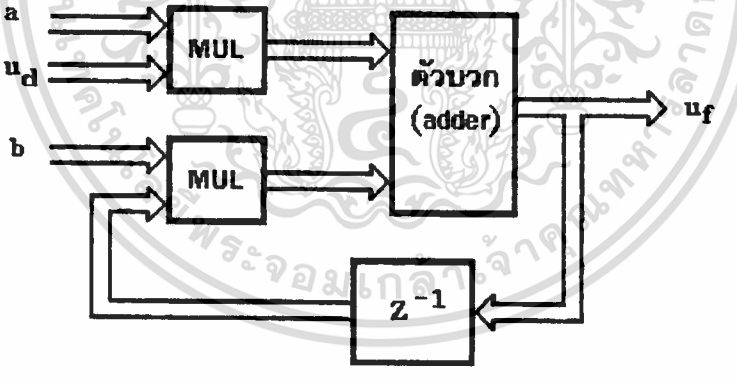
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$u_f(nT) = au_d(nT) + bu_f[(n-1)T] \quad (2\text{-}29)$$

ที่ซึ่ง $u_f(nT)$ แสดงความหมายว่าเป็นแซมเปิลของ $u_f(t)$ ณ เวลาดิสครีตที่ $t = nT$

สมการ 2-29 คือสมการของรีเคอร์ซีฟออร์เดอร์หนึ่ง(recursive fist-order)ของโลว์พาสฟิลเตอร์ มันจะบอกเป็นสัญญาณเอาต์พุตของลูปฟิลเตอร์ที่เวลา $t = nT$ ถูกคำนวณจากแซมเปิลของสัญญาณอินพุต $u_d(nT)$ และได้จากเอาต์พุตแซมเปิล $u_f = [(n-1)T]$ ซึ่งคำนวณได้จากช่วงเวลาแซมปลิงก่อนหน้านี้ โครงสร้างของดิจิทัลฟิลเตอร์สามารถวาดได้ตามรูปที่ 2-19 โอเปอร์เรเตอร์ z^{-1} หมายถึงดีเลย์(delay)เท่ากับหนึ่งช่วงเวลาแซมปลิง T

ดิจิทัลฟิลเตอร์สามารถถูกเพิ่มเติมโดยฮาร์ดแวร์หรือซอฟต์แวร์ได้ ถ้าเป็นฮาร์ดแวร์ตัวดีเลย์ z^{-1} ถูกสร้างขึ้นโดยเรจิสเตอร์เป็นบัฟเฟอร์ ถ้าการเพิ่มเติมเป็นซอฟต์แวร์แล้ว ค่าแซมเปิลดีเลย์เป็น $u_f = [(n-1)T]$ จะถูกเก็บไว้ในเรจิสเตอร์ภายในของ ซีพียู หรือในหน่วยความจำ ท้ายสุดของการเพิ่มเติมโดยฮาร์ดแวร์หรือซอฟต์แวร์ขึ้นอยู่กับความเร็วในการทำงานที่ต้องการ เหมือนกับที่ชี้แจงไว้ในตอนต้น



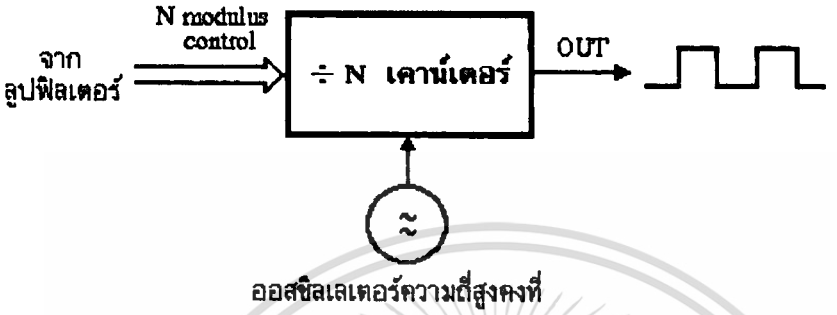
รูปที่ 2-19 *บล็อกไดอะแกรมของลูปฟิลเตอร์ส่วนเดียวแบบรีเคอร์ซีฟออร์เดอร์หนึ่ง*

ในบางกรณีอัตราการแซมปลิงของฟิลเตอร์แบบดิจิทัลต้องเลือกให้เหมาะสม ซึ่งคล้องจองกับทฤษฎีของ Nyquist คืออัตราแซมปลิงมีแบนวิทด์อย่างน้อยเป็นสองเท่าของสัญญาณ ถ้าให้เกิด ผลข้างเคียง (aliasing effect)น้อยที่สุด อัตราการแซมปลิงโดยปกติต้องเลือกอย่างน้อยที่สุดเป็น 10 เท่าของความถี่ f_O ของสัญญาณ u_1 และ u_2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.4 ดิจิทัลคอนโทรลออสซิลเลเตอร์(DCO)

ตัว DCO แบบต่างๆ สามารถออกแบบได้หลากหลายโดยได้ทั้งฮาร์ดแวร์และซอฟต์แวร์ ซึ่งเราจะได้พิจารณาโดยให้ชัดเจนจากนี้

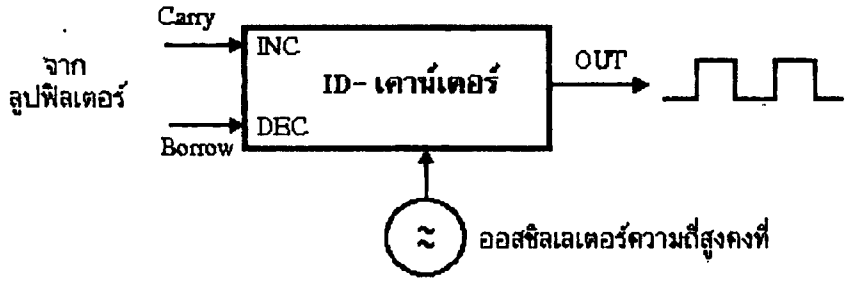


รูปที่ 2-20 บล็อกไดอะแกรมของ DCO แบบ หาร N

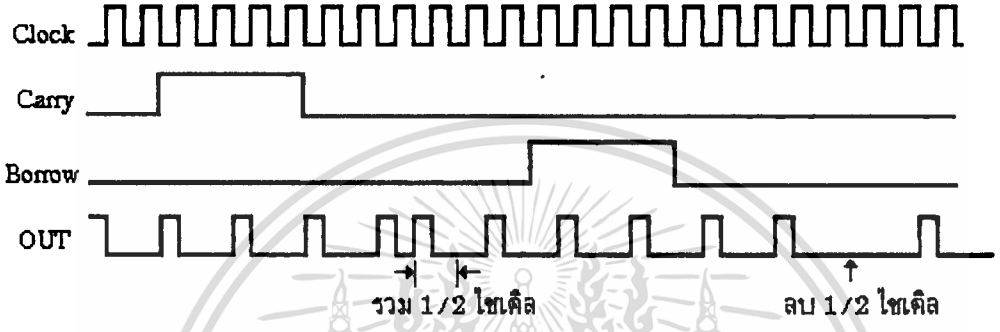
การแก้ปัญหาที่ง่ายที่สุดน่าจะเป็น DCO แบบตัวนับแฟกเตอร์ N (N counter DCO) ตามรูปที่ 2-20 จุดมุ่งหมายของ DCO ต้องทำงานร่วมกับลูปฟิลเตอร์ซึ่งสร้างพัลส์ทดและพัลส์ยืมอย่างเช่น ฟิลเตอร์แบบตัวนับ K หรือฟิลเตอร์แบบตัวนับ N-before-M ดังที่อธิบายไว้ในหัวข้อที่แล้ว การทำงานของตัวนับ I/D เป็นไปตามรูปคลื่นซึ่งแสดงไว้ในรูปที่ 2-21b ตัวนับ I/D จะหารความถี่อินพุตโดยแฟกเตอร์ 2 ซึ่งจะปราศจากเอาต์พุตเป็นพัลส์ทด(carry)และพัลส์ยืม(borrow) แต่เมื่อไรก็ตามที่พัลส์ทดมาปรากฏที่อินพุต INC ซึ่งไซเคิลของสัญญาณนาฬิกาจะถูกรวมเข้า โดยลอจิกภายในของตัวนับ I/D ซึ่งไซเคิลของสัญญาณนาฬิกาจะถูกลบออกในกรณีตรงกันข้ามกัน คือเมื่อไรก็ตามที่พัลส์ยืมมาปรากฏที่อินพุต DEC ดังนั้นความถี่เอาต์พุตของ DCO สามารถถูกควบคุมภายในช่วงซึ่งกำหนดโดยความถี่สูงสุดของพัลส์ทดและพัลส์ยืม รายละเอียดในการวิเคราะห์ของวงจรนี้หาได้ในเอกสารอ้างอิง 8

สองวงจร DCO ซึ่งได้อธิบายในข้างต้นเหมาะสมดีสำหรับฮาร์ดแวร์มากกว่าซอฟต์แวร์ DCO สำหรับ DCO แบบที่สามหรือเป็นแบบสุดท้าย ซึ่งเป็นแบบในอุดมคติในการใช้ซอฟต์แวร์ เรียกว่า "ตัวซินธีไซเซอร์รูปคลื่น DCO (waveform synthesizer)" ซึ่งตัว DCO แบบนี้จะสร้างรูปคลื่นซายน์ และ,หรือ รูปคลื่นโคซายน์โดยใช้วิธีการเปิดตาราง(looking up table)ซึ่งข้อมูลถูกเก็บไว้ใน ROM บล็อกไดอะแกรมของตัวซินธีไซเซอร์รูปคลื่น DCO แสดงในรูปที่ 2-22a

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(a)

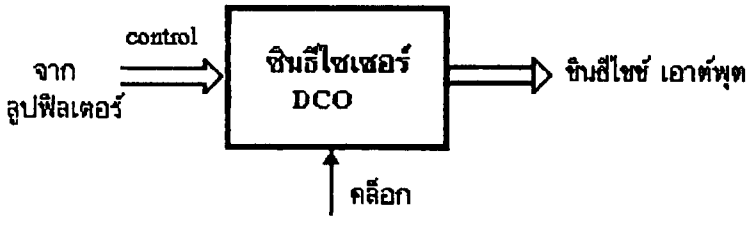


(b)

รูปที่ 2-21 *DCO แบบตัวนับนับเพิ่มและนับลด (a) บล็อกไดอะแกรม (b) การตอบสนอง*

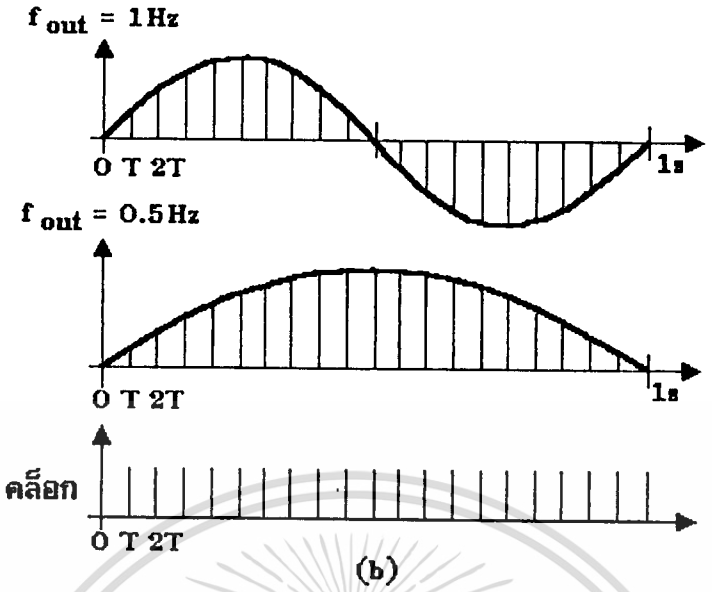
รูปคลื่นในรูป 2-22b แสดงให้เห็นว่าตัวซินธีไซเซอร์สามารถสร้างรูปคลื่นซายน์ ซึ่งมีความถี่ต่างกันได้อย่างไร อย่างเช่น 1 Hz และ 0.5 Hz มันทำงานที่อัตราความถี่คงที่คำนวณตามเวลาแซมปลิง ซึ่งความถี่ต่ำจะมีความละเอียดมากกว่าความถี่สูง

ในตัวอย่างในรูปที่ 2-22b ถ้าสมมุติตามอำเภอใจให้คาบเวลาการแซมปลิงเป็น 20 mS ซึ่งแน่นอนการทำงานตัวซินธีไซเซอร์ของจริงนั้นเร็วกว่ามาก รูปคลื่นซายน์ 1 Hz ถูกสร้างด้วยความละเอียด 20 แซมเปิลเพื่อจะให้เต็มคาบ ในกรณีรูปคลื่น 0.5 Hz ของซายน์จะมีแซมเปิลมากกว่า 2 เท่า เมื่อสร้างรูปคลื่นซายน์ 1 Hz ตัวซินธีไซเซอร์คำนวณฟังก์ชันซายน์สำหรับมุมเฟส ϕ = 0 เริ่มต้น ,$2\pi/20$, $2(2\pi/20)$, $3(2\pi/20)$,..... และโดยมุมเฟส ϕ ถูกเพิ่มขึ้นตามจำนวน $\Delta\phi$ = $2\pi/20$ ที่ทุกคาบของสัญญาณนาฬิกา



(a)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-22 *DCO แบบสังเคราะห์รูปคลื่น (a) บล็อกไดอะแกรม (b) รูปคลื่นแสดงให้เห็นเป็นซายน์ความถี่ 1 Hz และ 0.5 Hz ตามลำดับ*

ถ้ากำหนดความถี่ f ใดๆจะถูกสร้างขึ้นแล้วคาบเวลาการเพิ่มขึ้นเป็น $\Delta\phi$ จะเป็นดังนี้

$$\Delta\phi = 2\pi fT$$

ซึ่ง T เป็นคาบเวลาการแซมปลิง

ตัวซินธีไซเซอร์คำนวณ $\Delta\phi$ จากสัญญาณขนาน N บิตและสัญญาณ f-control ถูกส่งออกโดยลูปฟิลเตอร์แสดงดังรูป 2-22a ตัวซินธีไซเซอร์มีความสามารถสร้างสัญญาณเหมาะสมสู่อินพุต f-control ซึ่งไม่มีปัญหาเพื่อต้องการส่งรูปคลื่นแบบทันทีทันใดไม่ว่าทั้งรูปคลื่นซายน์และโคซายน์ เมื่อใช้เฟสดีเทกเตอร์แบบการแปลงฮิลเบอร์ต (Hilbert) กับระบบ ADPLL อ้างจากรูป 2-14a

ตัวซินธีไซเซอร์หรือตัวสังเคราะห์รูปคลื่นแบบดิจิทัล สามารถใช้งานโดยไมโครโปรเซสเซอร์แบบชิป-เดียว ความเร็วในการคำนวณของตรีโกณมิติสามารถนั้นจะใช้เทคนิคเปิดตาราง(lookup table) มากกว่าที่จะคำนวณฟังก์ชันซายน์ โดยใช้อนุกรมไทเลอร์(Tayler series)หรือ โพลิโนเมียลของ Chebysheff เหมือนในตัวอย่างการประยุกต์ใช้ของตัวซินธีไซเซอร์ในหัวข้อต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.5 ตัวอย่างการใช้ ADPLL เพิ่มเติม(Example of Implemented ADPLLs)

พื้นฐานของ PD ,ลูปฟิลเตอร์, และ DCO มีอยู่มากมายหลายแบบ ซึ่งถ้านำฟังก์ชันบล็อกแบบต่างมาประกอบกันเป็น ADPLL ได้เป็นจำนวนมาก ซึ่งเกินความจำเป็นที่จะบรรยายทั้งหมดโดยจะพิจารณาระบบ ADPLL เพียงสามแบบ

สองแบบแรกเป็นแบบแต่งเติมขึ้นโดยฮาร์ดแวร์ ในตัวอย่างตอนท้ายเป็นการอธิบาย PLL แบบใช้ซอฟต์แวร์ที่ซึ่งเต็มไปด้วยการกระทำทางคณิตศาสตร์ ซึ่งในการใช้เป็นฮาร์ดแวร์นั้นจะไม่ยุ่งยากซับซ้อนมากนัก

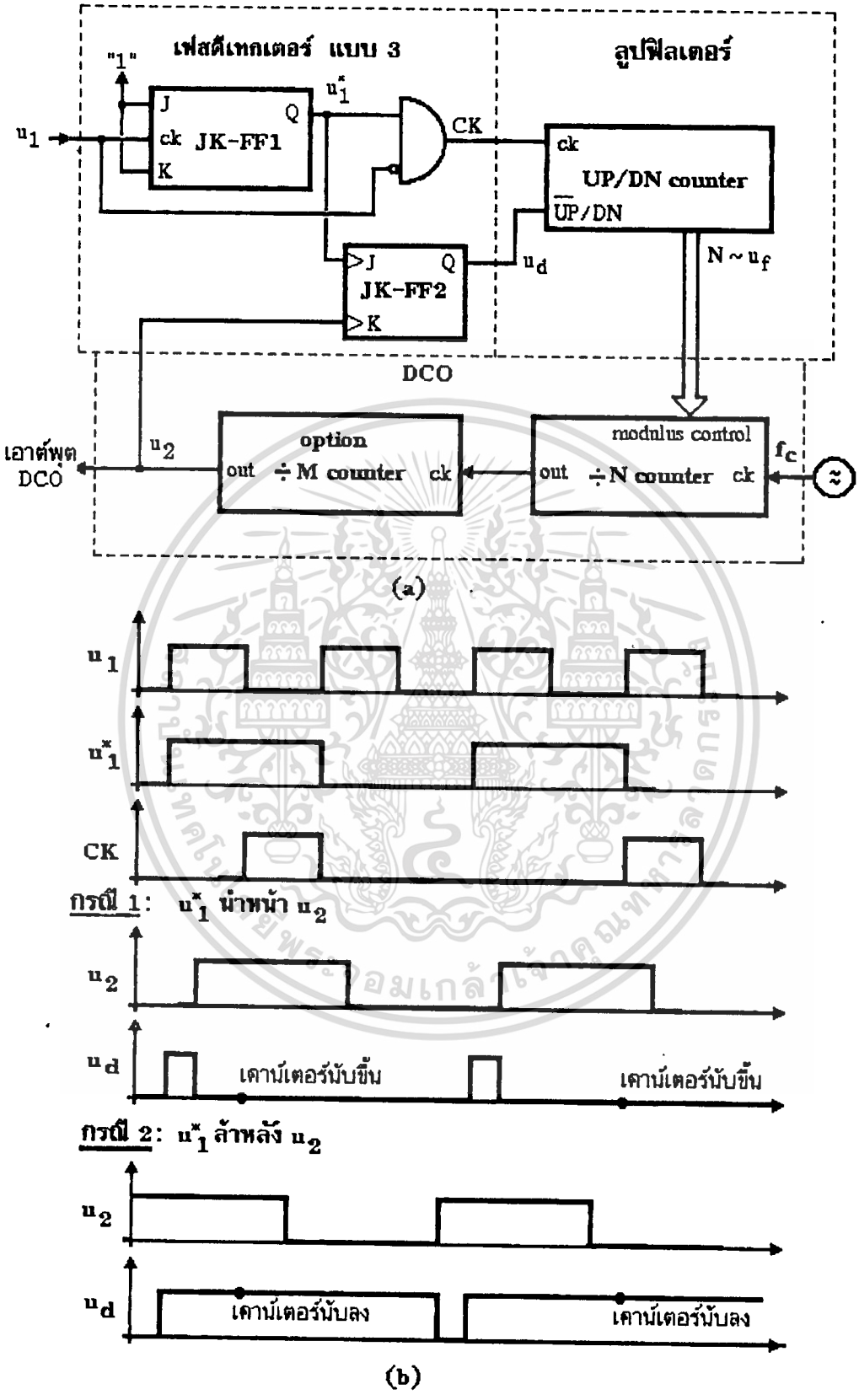
ตัวอย่างแรกของ ADPLL ถูกแสดงตามรูป 2-23a ระบบนี้ถูกสร้างจากดีเทกเตอร์แบบฟลิปฟลอบคล้ายกับแบบสาม,ลูปฟิลเตอร์ใช้ตัวนับแบบ UP/DOWN ซึ่งแสดงไว้ในหัวข้อที่แล้ว,และ DCO แทนด้วยตัวนับแบบแฟกเตอร์ N (แสดงไว้แล้วในรูป 2-20) วงจรภายใน PD ประกอบด้วยตัวแรกเป็นฟลิปฟลอปแบบ J-K และแอนด์เกท(AND gate) ซึ่งวงจรภายในจะผลิตรูปคลื่นสัญญาณ CK ป้อนให้กับตัวนับแบบ UP/DOWN ภายในบล็อกของลูปฟิลเตอร์ซึ่งถูกควบคุมด้วยฟลิปฟลอบตัวที่สองที่อยู่ภายใน

รูปคลื่นในรูป 2-23b ได้อธิบายความหมายหลักการของ PD ตัวฟลิปฟลอปแบบ JK-FF1 จะลดความถี่ของสัญญาณอินพุตด้วยแฟกเตอร์ 2 (หรือหารสอง) เรากำหนดสัญญาณอินพุต ซึ่งถูกลดความถี่ลงเป็น $u_1^*(t)$ สัญญาณ u_2 ที่ถูกสร้างขึ้นใหม่จะเหมือนกับสัญญาณ $u_1^*(t)$ ไม่ใช่ u_1 รูปคลื่นในรูป 2-23 นำมาวาดได้สองกรณีคือ

1. u_1^* นำหน้า u_2
2. u_1^* ล้าหลัง u_2

ตัว JK-FF2 เป็นฟลิปฟลอปของทริกขอบตามสัญญาลักษณ์ในรูปที่ 2-23 และสามารถใช้ฟลิปฟลอปแบบ R-S แบบทริกขอบได้เหมือนกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

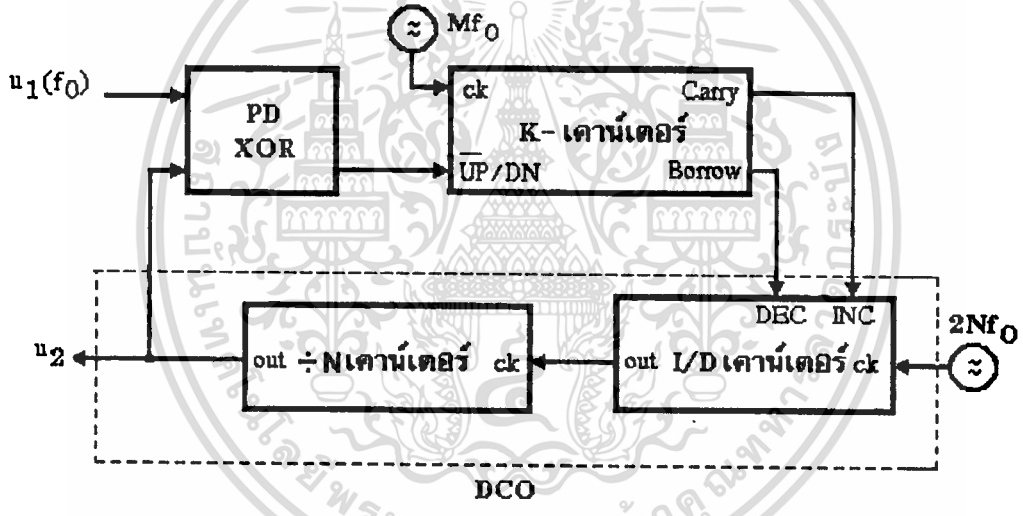


รูปที่ 2-23 *(a) บล็อกไดอะแกรม (b) การตอบสนองรูปคลื่น (1)* u_1^* *นำหน้า* u_2 *(2)* u_1^* *นำหน้า* u_2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของขาขึ้นของ u_1^* จะไปเซตฟลิปฟลอปตัวนี้ ในขณะที่สัญญาณ u_2 จะไปรีเซตมัน ที่ซึ่งจากไดอะแกรมรูปคลื่นที่สัญญาณคล็อกนับ CK เพื่อทำให้เกิดการนับเมื่อสัญญาณเอาต์พุต u_d หรือ JK-FF2 เป็น ลอจิกสูง เมื่อ u_1^* ล้าหลัง u_2 หรือเป็น ลอจิกต่ำ เมื่อ u_1^* นำหน้า u_2 ตามลำดับ ซึ่งสัญญาณเอาต์พุตเฟสดีเทกเตอร์ u_d ถูกใช้เหมือนตัวกำหนดทิศทางของตัวนับแบบ UP/DOWN ถ้าสมมุติว่าความถี่ของ u_1^* และ u_2 ไม่เท่ากันตัวนับจะนับทั้งนับไปข้างหน้า และนับถอยหลังจนกระทั่งถึงค่า N ซึ่งทำให้ตัวนับแฟกเตอร์ N สร้างสัญญาณเอาต์พุตที่ถูกต้องออกมา

เพราะว่า N สามารถใช้แปรค่าได้ที่ละหนึ่งสเต็ป ซึ่งสัญญาณความถี่ของ u_2 โดยปกติจะค่อยเพิ่มขึ้นหรือลดลงไม่มาก ซึ่งเหตุนี้จะเกิดจิตเตอร์(jitter)รอบค่าของ N และ N+1 อย่างต่อเนื่อง ถ้าสัญญาณอ้างอิงคงที่ ที่จุดสมดุล f_1 จะถูกทำให้เท่ากับ $f_c/(M \bullet N)$ เมื่อ M เป็นสเกลแฟกเตอร์(scale factor)



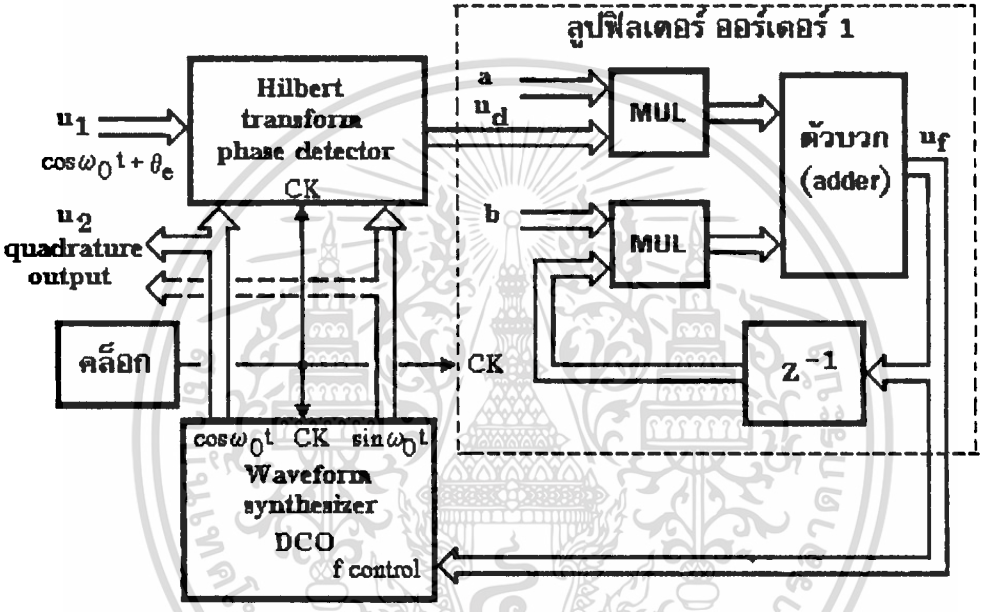
รูปที่ 2-24 *ระบบดิจิทัล PLL ระบบนี้ถูกทำให้เป็นจริงโดยวงจรรวมเบอร์ 74LS297*

ระบบ ADPLL แบบสองแสดงในรูป 2-24 ระบบนี้จะรวมอยู่ในไอซี TTL เบอร์ SN74LS297 และยังสามารถกำหนดลักษณะได้หลากหลายจากรูป 2-24 คือหนึ่งในหลายรูปแบบดังกล่าว นอกจากนี้บนชิปเดียวกัน ยังมี PD สองอย่างคือ แบบ XOR และแบบ JK-FF ส่วนลูปฟิลเตอร์ใน ADPLL นี้ใช้ลูปฟิลเตอร์แบบตัวนับ K ดังที่เคยบรรยายไว้ก่อนหน้านี้ดูจากรูป 2-17 และ DCO เป็นตัวนับ I/D ซึ่งเคยกล่าวไว้เช่นกัน

ในระบบของรูป 2-24 ถูกสมมุติว่าทำงานที่ความถี่กลาง(center frequency) f_O (โดยเทียบกับอินพุต u_1) ตัวนับ K และตัวนับ I/D จะถูกสัญญาณคล็อกขับซึ่งมีความถี่เท่ากับ M เท่าหรือ 2N เท่าของความถี่กลางตามลำดับ โดยปกติทั้ง M และ N จะถูกรวมเป็นเทอมของกำลังสอง เพื่อให้ง่ายกับการแก้ปัญหาทางคณิตศาสตร์ที่ความถี่เดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าอินพุตมีความถี่ f_O ตัวนับ I/D จะหารความถี่ด้วยแฟกเตอร์ 2 ที่แน่นอนและความถี่ u_1 และ u_2 จะมีความถี่เหมือนกันทุกประการ ตัวนับ K จะไม่ผลิตพัลส์ยืม(borrow) หรือพัลส์ทด(carry) ซึ่งนั่นคือระบบจะอยู่ในสภาวะที่จุดสมดุล ถ้าความถี่อินพุตถูกกำหนดเพิ่มขึ้นตัวนับ K จะเริ่มผลิตพัลส์ทด(carry)ที่ซึ่งทำให้ตัวนับ I/D เกิดการรวมพัลส์ในช่วงเวลา จำนวนพัลส์เฉลี่ยได้รวมเข้าทุกวินาทีเป็นจำนวน N เท่าของความถี่เบี่ยงเบน Δf ของความถี่อินพุตจากความถี่กลาง f_O ดูรูป 2-24 ประกอบรายละเอียด



รูปที่ 2-25 *ระบบ PLL แบบดิจิทัลล้วน ตัวอย่างที่ 3 เป็นระบบทางซอฟต์แวร์ดีที่สุด*

ตัวอย่างสุดท้ายของ ADPLL ถูกแสดงดังรูป 2-25 เป็นระบบใช้ซอฟต์แวร์เป็นหลัก โดยที่เป็นซอฟต์แวร์บนโปรเซสเซอร์เบอร์ TMS320 ซึ่งเป็นดิจิทัลซิกแนลโปรเซสซิ่ง(DSP) ซึ่งระบบตามรูปจะสร้างเลียนแบบบล็อกฟังก์ชันต่างๆดังนี้

1. เฟสดีเทกเตอร์แบบการแปลงฮิลเบอร์ต(Hilbert-transform)
2. ลูปฟิลเตอร์แบบออร์เดอร์หนึ่ง
3. DCO หรือซินธีไซเซอร์รูปคลื่น

ดังที่กำหนดไว้ในบล็อกไดอะแกรมตามรูป 2-25 การกระทำทางลอจิกหรือคณิตศาสตร์ภายในบล็อกของฟังก์ชันถูกปฏิบัติภายใต้สัญญาณนาฬิกา นี่ก็หมายความว่ารูทีน(routine)ทั้งหมดในการคำนวณหาค่าเอาต์-พุทบล็อกจะต้องดำเนินการเป็นคาบ ตัว DCO จะสร้างสัญญาณอินเฟสและควอดราเจอร์ I และ Q ตามลำดับ ด้วยการแปลงฮิลเบอร์ต(Hillbert transform) ตัว PD จะคำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ดัดแปลง เผยแพร่ หรือแสวงหาประโยชน์ทางการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นวณค่าเฟสเออเรอร์ u_d ~ θ_e สัญญาณเอาต์พุต u_d จะถูกกรองสัญญาณแบบดิจิทัลด้วยตัว-ลูปฟิลเตอร์จะทำหน้าที่นี้

$$u_f(nT) = au_d(nT) + bu_f\left[(n-1)T\right] \quad (2\text{-}29)$$

ที่ซึ่ง a และ b คือค่าคงที่ของฟิลเตอร์ และได้รับการกำหนดมาจากความถี่จุดตัดที่ต้องการ(cutoff frequency)และอัตราการแซมปลิงเหมือนกับแสดงดังสมการ 2-28

หนึ่งในข้อได้เปรียบของการใช้ซอฟต์แวร์ คือจะง่ายต่อการเปลี่ยนแปลงโครงสร้างของระบบ ADPLL ข้อได้เปรียบต่อมาคือ การดัดแปลงโปรแกรมลูปฟิลเตอร์แบบออร์เดอร์หนึ่ง เป็นลูปฟิลเตอร์แบบออร์เดอร์สองเป็นต้น โดยเหตุนี้ก็จะทำให้ ADPLL จะเป็นฟังก์ชันถ่ายโอนออร์เดอร์สาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

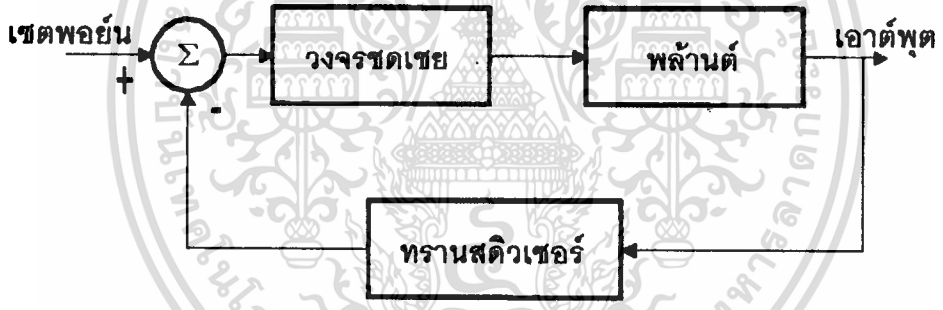
บทที่ 3

โหมดการควบคุมของคอนโทรลเลอร์

3.1 นิยามระบบควบคุม

การนิยามของระบบควบคุม นามมาแล้วการพัฒนาการผลิตในเชิงอัตโนมัติ(automation)ได้ถูกเข้ามาสนใจมากขึ้น ซึ่งเรียกอย่างเดียวกันว่า "ระบบควบคุม" ซึ่งเทอมนี้นั้นมีความหมายหลายแนวความคิด โดยมีจุดประสงค์แตกต่างกันออกไป ทำให้มีเทคนิคต่างๆที่สำคัญเป็นประโยชน์ต่องานวิศวกรรม โดยมีแนวทางปฏิบัติให้เป็นจริง

ระบบควบคุมสามารถจัดแบ่งเป็นแบบ เซลฟเรกูเลชัน(self regulation) ตามรูปที่ 3-1 เป็นบล็อกไดอะแกรมของส่วนประกอบต่างๆ ซึ่งเกี่ยวข้องกับเทอมต่างในระบบ



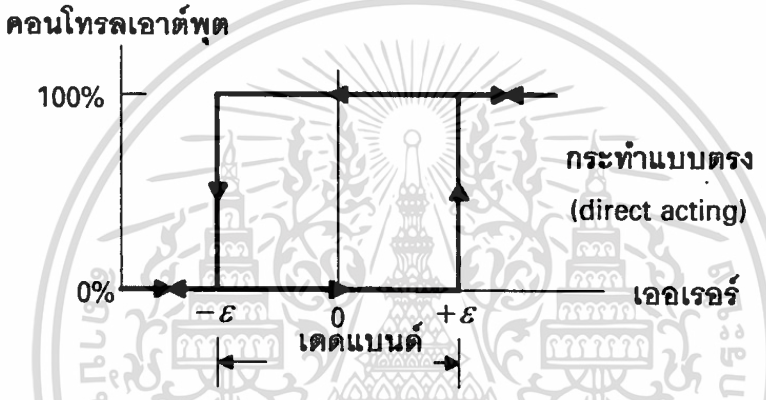
รูปที่ 3-1 *ส่วนประกอบของระบบควบคุมอย่างง่าย*

ระบบต้องการอินพุตคือค่าเซตพอย์น เป็นการเตรียมจุดหมายให้ระบบเซลฟเรกูเลชัน การบรรลุถึงจุดหมายนั้น จะต้องใช้ความรู้ในการปฏิบัติคือ การใช้จุดต่อร่วม(summing junction)เพื่อสร้างเออเรอร์จากการป้อนกลับของเอาต์พุต โดยเตรียมการป้อนกลับจากตัวทรานสดิวเซอร์ ถ้าต้องการค่าเออร์เรอร์ลดน้อยลงแล้วค่าป้อนกลับต้องมีค่าผกผันซึ่งเรียกว่า "ป้อนกลับลบหรือ เนกาตีฟฟีดแบค(negative feedback)" เออเรอร์เหล่านี้จะป้อนให้กับวงจรชดเชย(compensate network) ซึ่งประกอบด้วยตัวขยาย ตัวขับฟิลเตอร์ และอื่นๆ ซึ่งเรียกว่าการควบคุม(control action) สัญญาณที่ได้จากวงจรนี้จะถูกส่งให้กับพล้านต์(plant)ต่อไป และนี้คือความหมายของระบบควบคุมโดยสมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การควบคุมแบบเปิด-ปิด(ON-OFF Control Action)

เป็นหนึ่งของการควบคุมแบบธรรมดาที่สุดและง่ายที่สุด โหมดนี้มีตัวอย่างตามรูปที่ 3-2 ซึ่งการควบคุมแบบนี้มีฟังก์ชันถ่ายโอนกับสภาวะเอาต์พุตที่เป็นไปได้สองสถานะคือ 0% กับ 100% ถ้าค่าเออเรอร์เพิ่มขึ้นเกินค่าที่กำหนด เอาต์พุตจะเปลี่ยนจาก 0% เป็น 100% แต่ถ้าเออเรอร์ต่ำกว่าค่าที่กำหนดแล้ว เอาต์พุตก็จะลดลงจาก 100% เป็น 0% ดังนั้นค่าเดดแบนด์(dead band) เป็นตัวกำหนดการเปลี่ยนแปลงโดยอ้างที่จุดศูนย์ ตัวอย่างในระบบควบคุมซึ่งใช้การควบคุมแบบเปิด/ปิด เช่น ระบบปรับอากาศเป็นต้น ดังนั้นสมการการควบคุมแบบนี้คือ



รูปที่ 3-2 *กราฟแสดงการควบคุมแบบเปิด-ปิด และเดดแบนด์*

$$c = \begin{cases} 0\% & e < -\varepsilon \\ 100\% & e > +\varepsilon \end{cases} \qquad (3.1)$$

3.3 การควบคุมแบบต่อเนื่อง(Continuous Control Action)

เป็นการควบคุม ที่ใช้มากที่สุดในระบบควบคุมกระบวนการแบบต่อเนื่อง เพราะว่ามีเรกูเลชันดีกว่าการควบคุมแบบเปิด-ปิด การควบคุมแบบนี้จัดแบ่งเป็นโหมดต่างๆดังนี้

3.3.1 โหมดการควบคุมแบบพรอปพรอชันแนล(Proprotional Control Mode)

การควบคุมแบบพรอปพรอชันแนลเรียกโดยย่อว่าการควบคุมแบบ "P" การทำงานของโหมดนี้ในการควบคุมกระบวนการ เอาต์พุตของการควบคุมแบบพรอปพรอชันแนลจะเป็นสัดส่วนโดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กับค่าเออเรอร์ของตัวเองคูณด้วยค่าคงที่ โดยที่ค่าเออเรอร์เกิดจากการแก้ไขค่าป้อนกลับนั่นเอง ซึ่งสามารถเขียนสมการได้ดังนี้

$$c_p = K_p e_p + c_0$$

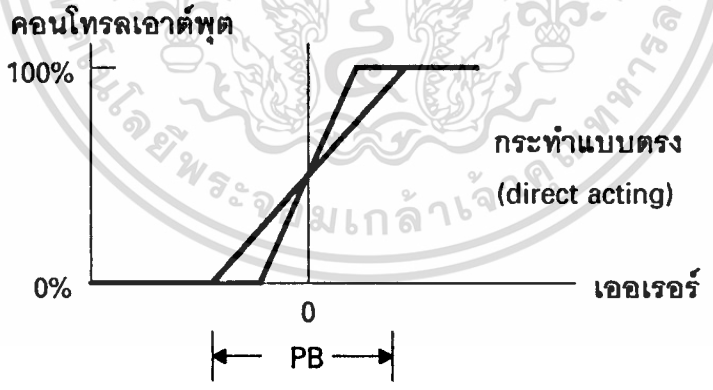
ที่ซึ่ง c_P = เอาต์พุตของคอนโทรลเลอร์ในเปอร์เซ็นต์

K_p = พรอปพรอชันแนลเกนใน เปอร์เซ็นต์ของเอาต์พุต/เปอร์เซ็นต์ของเออเรอร์

e_P = เออเรอร์ในเปอร์เซ็นต์ของช่วง

c_O = เอาต์พุตของเออเรอร์ศูนย์

การของโหมดนี้ถูกแสดงในรูปที่ 3-2 แสดงค่าเกนสองค่าที่แตกต่างกัน สังเกตว่ามีค่าอิ่มตัว (saturation value)ของเออเรอร์เมื่อเอาต์พุตถึง 100% โดยที่ค่าเออเรอร์จะมากขึ้นไปกว่านี้นั้น จะไม่มีผลิตเอาต์พุตที่เพิ่มขึ้นมากไปกว่านี้ ซึ่งจะมีผลเหมือนกันเมื่อเอาต์พุตตกลงที่ 0% แถบเออเรอร์ภายในเอาต์พุตระหว่าง 0% ถึง 100% ถูกเรียกว่า "พรอปพรอชันแนลแบนด์(proportional band)หรือ PB" ที่ค่าเกนมีค่าสูงพรอปพรอชันแนลแบนด์จะมีค่าต่ำ



รูปที่ 3-3 *คอนโทรลเอาต์พุตในโหมด P จะราบเรียบภายในข้อกำหนด ของเกนและเออเรอร์*

ออฟเซตเออเรอร์(offset error) เป็นปัญหาอย่างหนึ่งกับการควบคุมแบบพรอปพรอชันแนล นั่นคือความจริงที่ว่า **เออเรอร์ศูนย์**(zero error) เป็นค่าคงที่ ค่า c_O ถูกเลือกมาในตอนต้นให้เหมาะสม ค่า c_O เป็นค่านอกเหนือจากการควบคุม โดยเกี่ยวพันกับการเปลี่ยนแปรในกระบวนการ(process) การแปรเปลี่ยนแบบทันใดของตัวแปรอื่นๆเหล่านั้น จะเป็นเหตุให้ตัวแปรควบคุมเปลี่ยนด้วย ซึ่งแก้ไขให้ถูกต้องโดยเปลี่ยนค่า c_P คล้องจองกับสมการควบคุม อย่างไรก็ดีถ้าตัวแปร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินพุตเปลี่ยนแปลงแบบที่เรียกว่า "การเปลี่ยนแปลงโหลด" แล้วการควบคุมแบบนี้ไม่ได้แก้ไขไว้ ถึงอย่างไรก็สนับสนุนระบบที่มีค่าเออเรอร์ศูนย์ หรือ c_O คงที่

3.3.2 โหมดการควบคุมแบบอินทีกรัล(Intetgal Mode)

เป็นโหมดหนึ่งที่ใช้งานบ่อยๆในระบบการควบคุม โดยมีพื้นฐานเกี่ยวเนื่องกับเหตุการณ์ในอดีตของค่าเออเรอร์ ซึ่งเกิดขึ้นในการแปรผันทางการควบคุม โหมดการควบคุมนี้สามารถเรียกอีกอย่างหนึ่งว่า "โหมดรีเซต(reset)" ถ้าการควบคุมแบบพรอปพรอชันแนลกำหนดการป้อนกลับเป็นตัวแปรเออเรอร์ของปัจจุบันโดยเฉพาะแล้ว การควบคุมแบบอินทีกรัลกำหนดการป้อนกลับบนพื้นฐานของเออเรอร์ในอดีตที่เกิดขึ้น ดังนั้นเป็นไปได้ว่า เมื่อขณะใดขณะหนึ่งถ้าค่าเออเรอร์เป็นศูนย์แล้ว การควบคุมแบบอินทีกรัลยังคงถูกป้อนกลับอยู่เพราะเกิดจากเออเรอร์ในอดีต

จากนี้เราสามารถมองเห็นภาพพจน์ของโหมดนี้ เทอมของอินทีกรัลสามารถคำนวณจากพื้นที่สุทธิ(net area) ภายใต้เส้นกราฟเออเรอร์ต่อเวลา การกระทำเอาต์พุตถูกพบว่าเป็นการคูณด้วยค่าคงที่ซึ่งเรียกว่า "อินทีกรัลเกน" ดังนั้น

$$c_I(t) = K_I A_e(t) + c_I(0)$$

ที่ซึ่ง $c_I(t)$ = คอนโทรลเอาต์พุตสำหรับโหมดอินทีกรัล(%)ที่เวลา t

K_I = ค่าเกน[% per (%-time)]

$A_e(t)$ = พื้นที่สุทธิของเออเรอร์ต่อเวลา(%-time)คำนวณจากเวลา t จาก t = 0

$c_I(0)$ = คอนโทรลเอาต์พุตที่เวลา t = 0

สมการตามปกติจะเขียนอยู่ในรูปสมการอินทีกรัลดังนี้

$$c_I(t) = K_I \int_0^t e_p(t)dt + c_I(0)$$

กระบวนการรีเซต การกระทำอินทีกรัลเตรียมเอาต์พุต "flaot" ด้วยเหตุนี้เพราะค่าเอาต์พุตที่เออเรอร์ศูนย์(zero error) จะขึ้นอยู่กับเออเรอร์ในอดีต นี้หมายความว่าในโหมดอินทีกรัลสามารถกำหนดเอาต์พุตอัตโนมัติเพื่อเผชิญการเปลี่ยนแปลงโหลดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3 โหมดการควบคุมแบบดีริเวทีฟ(Derivative Mode)

การกระทำในโหมดนี้ถูกพิจารณาว่า เป็นการปฏิบัติการของการควบคุม อยู่บนพื้นฐานของอัตราเวลาของการเปลี่ยนเออเรอร์(time rate of change of error) ในกรณีที่เออเรอร์ที่แท้จริงเป็นศูนย์ การกระทำดีริเวทีฟยังจะให้การตอบสนองได้เฉพาะอัตราเปลี่ยนเออเรอร์ ซึ่งการขึ้นอยู่กับอัตราการเปลี่ยนเออเรอร์นั้น ทำให้เรียกบ่อยๆว่า "เรตแอคชั่นโหมด(rate action)" การตอบสนองของการปฏิบัติดีริเวทีฟจะไม่ขึ้นอยู่กับค่าเออเรอร์ในขณะนั้นโดยเฉพาะ แต่จะขึ้นอยู่กับอัตราการเปลี่ยนของค่าเออเรอร์ในขณะนั้น รูปที่ 3-4 แสดงกรณีที่เออเรอร์เป็นศูนย์ขณะหนึ่ง แต่ว่าเพื่อให้กระจ่างเอาต์พุตคอนโทรลจำเป็นต้องปรับค่าบางสิ่ง โดยค่าเออเรอร์ที่กำลังเปลี่ยนแปลงแบบรวดเร็วนั้นจะประกันได้ว่าเอาต์พุตคอนโทรลจะไม่เป็นศูนย์เว้นเสียแต่ว่าจะมีเงื่อนไขอื่น



รูปที่ 3-4 *ในกรณีที่เออเรอร์เป็นศูนย์ แต่เอาต์พุตดีริเวทีฟมีการตอบสนอง*

อัตราการเปลี่ยนของเออเรอร์สามารถถูกประมาณ โดยการเอาผลต่างระหว่างค่าเออเรอร์สองค่าหารด้วยเวลาต่างกัน ดังนี้

$$c_D(t) = K_D \frac{e_p(t) - e_p(t_0)}{t - t_0}$$

ที่ซึ่ง $c_D(t)$ = เอาต์พุตการควบคุมแบบดีริเวทีฟที่เวลา t (%)

K_D = ค่าเกนดีริเวทีฟ[% per (%/time)]

$e_P(t)$ = เออเรอร์ที่เวลา t (%)

$e_P(t_0)$ = เออเรอร์ที่เวลา t_0 (%)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สังเกตได้ว่าไม่มีเทอมเอาต์พุตของการควบคุมออกมา เมื่อมีอัตราที่เปลี่ยนแปลงเออเรอร์เริ่มต้นเป็นศูนย์ ด้วยเหตุนี้การควบคุมโหมดดีริเวทีฟจะไม่ใช้งานแบบโดดเดี่ยว เพราะว่าจะไม่ตอบสนองค่าเออเรอร์คงที่(fixed error)

การแสดงสมการทางคณิศาสตร์ที่เหมาะสมสำหรับดีริเวทีฟโหมดเป็นรูปแบบแคลคูลัสคือ

$$c_D = K_D \frac{de_p}{dt}$$

3.3.4 โหมดการควบคุมแบบคอมโพสิต(Composite Mode)

การปฏิบัติการควบคุมในโหมดนี้ โดยปกติจะใช้สามโหมดข้างต้นมารวมกัน โดยนำสมการมารวมกันแบบธรรมดา ดังนี้

1. โหมด PI (proprotional-integral)

$$c_{PI} = K_P e_P + K_P K_I A_e$$

2. โหมด PD (proprotional-derivative)

$$c_{PD} = K_P e_P + K_P K_D \frac{e_P(t) - e_P(t_0)}{t - t_0}$$

3. โหมด PID (proprotional-integral-derivative)

$$c_{PID} = K_P e_P + K_P K_I A_e + K_P K_D \frac{e_P(t) - e_P(t_0)}{t - t_0}$$

สังเกตว่าพรอปพรอชันแนลเกนจะปรากฏในทุกๆเทอมของโหมดคอมโพสิต

3.4 บทนิยามอัลกอลิธึมการควบคุม(Control Action Algorithm Definition)

เมื่อได้พิจารณาปัญหาของคอมพิวเตอร์เมื่อเตรียมการใช้กับงานควบคุม โดยสำหรับข้อมูลอินพุต และเอาต์พุตการควบคุมที่แปรผัน ซึ่งได้รับอิทธิพลจากธรรมชาติของดิสครีต(discrete) บทนี้ในปัญหาทั่วๆไปของการควบคุมกระบวนการ นั้นเป็นสิ่งแปรเปลี่ยนไปอย่างต่อเนื่อง ดังนั้นเราจะได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ศึกษาต่อไป ซึ่งมองดูผิวเผินในการประยุกต์ของคอมพิวเตอร์ในงานควบคุมของกระบวนการต่อเนื่อง จะต้องแทนข้อมูลในรูปดิสครีต นี่คือปัญหาที่ต้องจัดเตรียม โดยการใช้โค๊ดแทนข้อมูลสำหรับให้คอมพิวเตอร์ประมวลผล หลักการเปลี่ยนข้อมูลของโปรเซสเป็นข้อมูลดิจิทัลไม่มีอะไรต้องวิตกเพราะเป็นความสามารถทางฮาร์ดแวร์ อย่างไรก็ดีต้องจำไว้ว่าการใช้คอมพิวเตอร์ใช้ประมวลผลในการปฏิบัติกับรูปแบบการควบคุมต่อเนื่อง โดยการใช้ดิสครีตแทนนั้นจะยุ่งยากพอควร

ต่อไปต้องทำความรู้จักกับองค์ประกอบพื้นฐานดังนี้

1. สมการควบคุม(Control Equation)
2. ค่าเออเรอร์ของตัวแปรควบคุม(Controlled Variable Error)
3. ขั้วเครื่องหมายของ DE (Polarity of DE)
4. เอาต์พุตตัวแปรการควบคุม(Controlling Variable Output)

องค์ประกอบพื้นฐานเหล่านี้สามารถอธิบายได้ดังนี้

1) สมการควบคุม(Control Equation)

วัตถุประสงค์พื้นฐานของหัวข้อนี้ เป็นการแสดงสมการการควบคุมในโหมดต่างๆ เป็นอย่างเช่น เราเคยเสนอไว้ในตอนต้น โดยที่เราต้องเพิ่มเติมบางสิ่งขึ้นมาเพื่อใช้กับคอมพิวเตอร์ นี่เองหมายความว่าจะต้องมีอัลกอลิธึม ซึ่งจะเตรียมหาไว้ในการแก้ปัญหาทางสมการควบคุม ในความเข้าใจทั่วไปสมการจะลดลงเหลือในรูปสมการ

$$DC = F(DV) \tag{3-1}$$

เมื่อ DC = เอาต์พุตดิจิทัลของค่าตัวแปรควบคุม(controlling variable)

DV = อินพุตดิจิทัลจากค่าตัวแปรควบคุม

สมการที่ 3-1 เป็นการแทนดิสครีตแทนการแปรผันต่อเนื่อง โดยที่คอมพิวเตอร์จะต้องใช้ไบต์ข้อมูลที่มีบิตจำนวนจำกัดเป็นตัวแปร ตัวอย่างเช่น ถ้าเป็น 8 บิตตามสมการที่ 3-1สามารถแทนข้อมูลด้วยความละเอียดเพียง 1 ใน 256 เท่านั้น ซึ่งต้องรู้จักข้อจำกัดนี้เป็นผลที่ตามมาภายหลังในกระบวนการควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) **ค่าเออเรอร์ของตัวแปรควบคุม**(Controlled Variable Error)

เริ่มแรกในข้อปฏิบัติ โดยเพิ่มเติมกระทำกับสมการที่ 3-1 เพื่อหาค่าเออเรอร์ของตัวแปรควบคุม DV จากตัวแปรเซตพอย์นที่กำหนดขึ้นคือ DSP อย่างไรก็ดีมีหลายวิธีที่แสดงถึงค่าเออเรอร์ซึ่งสำหรับในหัวข้อนี้ และความเข้าใจทั่วๆไปในอัลกอลิธึมในการควบคุม เราสามารถจะนิยามเพื่อหาค่าเออเรอร์ได้ เว้นเสียแต่การแทนค่าจริงของตัวแปร เช่นการเปลี่ยนข้อมูลจากระบบแอนะล็อกของกระแส 4 - 20 มิลลิแอมป์ ในกรณีการควบคุมแบบดิจิทัลช่วงกายภาพของตัวแปรในคอมพิวเตอร์ จะต้องสมมุติเป็นข้อมูลเต็มการนับในไบต์ของข้อมูล ซึ่งเป็นข้อมูล 8 บิตโดยมีช่วงระหว่าง 00 - FFh โดยเลือกค่าเออเรอร์ให้ง่ายต่อ การแสดงเหมือนเป็นเศษส่วนของช่วง(fraction of range) ด้วยการใช้การนิยามของเปอร์เซ็นต์ของช่วง โดยสมการดังนี้

$$e_P = 100\frac{V - V_{SP}}{V_{\max} - V_{\min}} \qquad (3\text{-}2)$$

ที่ซึ่ง v_{SP} คือ ตัวแปรควบคุมเซตพอย์น

สิ่งแรกจะต้องทำคือ เออเรอร์ต้องถูกกำหนดโดยค่าแตกต่างระหว่างค่าวัดได้ กับเซตพอย์นดังนี้

$$DE = DV - DSP \qquad (3\text{-}3)$$

ค่าเออเรอร์ DE จะเป็นจำนวนดิจิทัลแทนในรูป 2's คอมพลีเมนต์ ซึ่งหมายความว่าถ้า DE เป็นจำนวนลบจะอยู่ในรูป 2's คอมพลีเมนต์

ในการสอดคล้องกับการนิยามของสมการที่ 3-2 ค่าเออเรอร์สามารถแสดงอัตราส่วนของช่วงการหารของเออเรอร์ในช่วง

$$DFE = \frac{DE}{RIN} \qquad (3\text{-}4)$$

ซึ่ง RIN = ช่วงตัวแปรอินพุต = $DMAX - DMIN$

DFE = ค่าเออเรอร์โดยเป็นสัดส่วนกับช่วง(range)

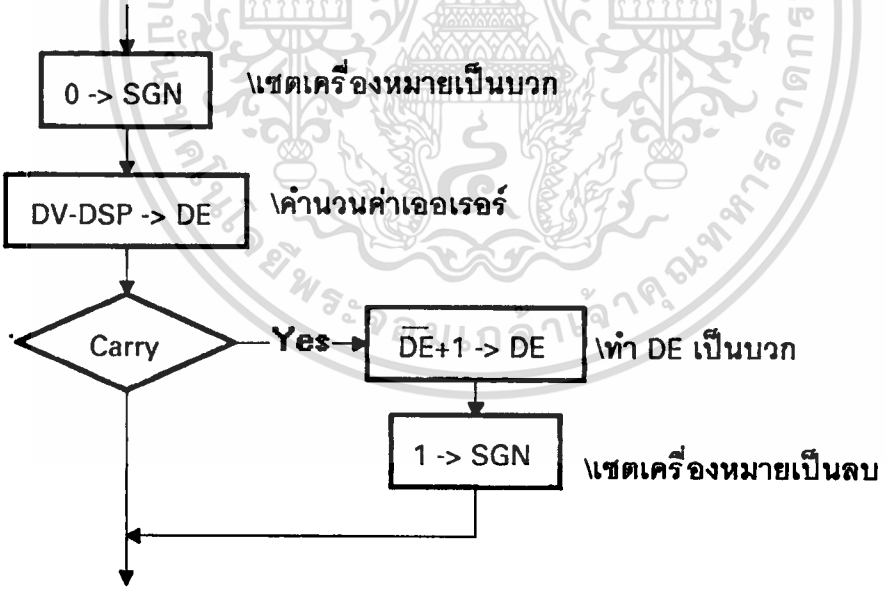
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยทั่วๆไปเราจะใช้เป็นช่วงเต็มของอินพุตคือ FFH ซึ่งเป็นการเริ่มต้นง่ายๆของสมการเพื่อให้ได้ผลลัพท์ที่ดีที่สุด ผลที่แตกต่างออกไปจะต้องพิจารณาช่วงการทำงานด้วย

3) ขั้วเครื่องหมายของ DE (Polarity of DE)

ค่าของ *DE* คำนวณจากสมการที่ 3-3 จะมีขั้วเครื่องหมายได้สองแบบ หมายความว่าถ้า *DV* มากกว่า *DSP* แล้ว *DE* จะมีเครื่องหมายเป็นลบและแทนใน 2's คอมพลีเมนต์ ซึ่งอาจจะเป็นการไม่สะดวกในการใช้แทนคำสั่งของกระบวนการคำนวนของไมโครโปรเซสเซอร์ เพื่อให้ได้เอาต์พุตออกมาเป็น 2's นั้นจะต้องกำหนดให้ *DE* ต้องใช้เป็นค่า(value)ใดๆไม่ติดขั้วเครื่องหมาย(polarity) โดยที่ขั้วเครื่องหมายจะแทนเป็นค่าแฟลก

โฟล์วชาร์ตตามรูปที่ 3-5 เป็นการแสดงถึงการทำงานโดยทั่วไปของโปรแกรม ซึ่งเป็นไปตามความจริงที่ว่าถ้าผลลัพท์เป็นลบ บิตทด(carry bit)จะถูกเซต และผลลัพท์จะเป็น 2's คอมพลีเมนต์ ในกรณีนี้แฟลก *SGN* จะถูกเซตเป็น 1 ซึ่งจะชี้ว่าค่า *DE* จะต้องเป็นค่าลบ โดยที่ตัวมันเองจะไม่แสดงเครื่องหมาย



รูปที่ 3-5 *เป็นโฟล์วชาร์ตแสดงค่า DE ทั้งขนาดและเครื่องหมาย*

4.เอาต์พุตตัวแปรการควบคุม(Controlling Variable Output)

มีรูปแบบการกำหนดตัวควบคุมและค่าเออเรอร์ โดยคงสภาพรูปแบบที่กำหนดไว้เพื่อใช้เป็นเอาต์พุตตัวแปรควบคุม ซึ่งเอาต์พุตตัวแปรควบคุมกำหนดอย่างง่ายเป็นค่าเปอร์เซ็นต์ 0% ถึง 100% เช่นในลูปควบคุมทางแอนะล็อกจะแปรค่าเป็นกระแสไฟฟ้า 4 - 20 มิลลิแอมป์ หรือถ้าเป็นสัญญาณ

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นิวเมติก คือ 3 PSI ถึง 15 PSI สำหรับกรณีทางดิจิทัลเราจะกำหนดสมการมีผลลัพท์ DC อยู่ในช่วงจาก 00H คล้องจองกับ 0% ถึง FFH คล้องจองกับ 100%

3.5 อัลกอลิธึมของโหมดควบคุม(Pure Mode Control Algorithm)

หมวดนี้จะเสนออัลกอลิธึมของคอมพิวเตอร์ ซึ่งทำงานโหมดคอนโทรลเลอร์แบบพื้นฐานขยายเพิ่มเติม คือ พรอปพรอชันแนล อินทีกรัล(รีเซต) และ ดีริเวทีฟ(เรต) ในแต่ละกรณีอัลกอลิธึมจะบรรยายโดยสมการ และ โฟล์วชาร์ตของระหัสที่ใช้จริงของไมโครโปรเซสเซอร์ ซึ่งไม่ใช่เรื่องยากที่จะรวมเอาโหมดการควบคุมเหล่านั้นมาประยุกต์ใช้งานจริง

3.5.1 โหมดพรอปพรอชันแนล(Proportional Mode)

โหมดพรอปพรอชันแนลค่าเอาต์พุตจะเป็นสัดส่วนโดยตรงกับค่าเออเรอร์ซึ่งสมการจะเป็น

$$c = K_P e_P + c_0$$

อัลกอลิธึมสำหรับใช้ของสมการในไมโครคอมพิวเตอร์คอนโทรลมีรูปแบบคล้ายคลึงกันคือ

$$DCP = (KP \bullet DFE \bullet ROUT) + DCO \qquad (3\text{-}5)$$

ซึ่ง DCP = ดิจิทัลคอนโทรลเอาต์พุต

KP = ดิจิทัลพรอปพรอชันแนลเกน

DFE = ดิจิทัลเออเรอร์กับสัดส่วนช่วง

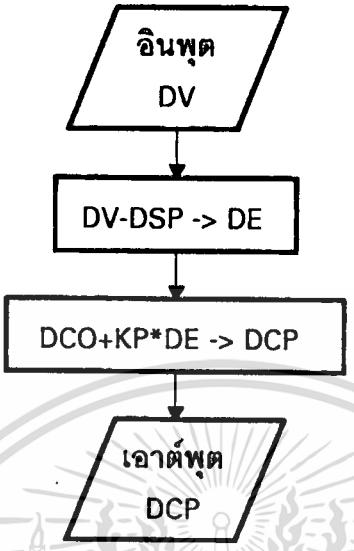
$DC0$ = เออเรอร์เริ่มต้นของเอาต์พุต

$ROUT$ = ช่วงของเอาต์พุต

ถ้าตัวแปรอินพุตคอนโทรลเท่ากับค่าเซตพอย์น ดังนั้นค่าเออเรอร์จะมีค่าเท่ากับศูนย์ของตัวเอาต์พุต $DC0$ ค่าดิจิทัลเออเรอร์จะกำหนดจากสมการที่ 3-4 และเอาต์พุตการควบคุมจะแปรค่าครอบคลุมช่วงที่ระบุโดย $ROUT$ ค่าของ KP ถูกนิยามเหมือนกับอัตราส่วนการเปลี่ยนต่อค่าเออเรอร์กับช่วงอินพุต ดังนั้นผลจาก $KP \bullet DFE$ คืออัตราส่วนของเอาต์พุตที่เปลี่ยนไประบุโดยค่าเออเรอร์ ดังนั้นจำนวนเท่าของช่วงเอาต์พุตคือ เอาต์พุตที่เปลี่ยนอย่างแท้จริง สมการนี้สามารถแสดงเป็นรูปฟอร์มที่ไม่มี DFE ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$DCP = KP \bullet DE \bullet (ROUT / RIN) + DCO \qquad (3\text{-}6)$$



รูปที่ 3-6 *โฟล์วชาร์ตอัลกอลิธึมของโหมดพรอปพรอชันแนล*

ตอนนี้สามารถมองเห็นเหตุผลความต้องการสำหรับที่จะทำช่วงอินพุต และเอาต์พุตให้เท่ากัน ถ้า $ROUT = RIN$ แล้ว สมการพรอปพรอชันแนลโหมดจะพัวพันเฉพาะค่าเกน ค่าเออเรอร์ และเทอมเออเรอร์ที่ศูนย์

1) สมการอัลกอลิธึมอย่างง่ายของโหมด P (Simplified Algorithm Equation)

สำหรับจุดประสงค์ของหัวข้อนี้ จะต้องสมมุติว่าช่วงของอินพุตและเอาต์พุตถูกเลือกให้มีค่าเท่ากัน ซึ่งตัวเลือกนี้โดยปกติจะนับได้เต็ม 1 ไบต์ข้อมูลถ้าเป็นไมโครโปรเซสเซอร์ 8 บิตจะได้ช่วงถึง FFH ซึ่งจะมีสมการที่ต้องตัดบางส่วนออกไปดังนี้

$$DCP = (KP \bullet DE) + DC0 \qquad (3\text{-}7)$$

เมื่อ

DCP = เอาต์พุตคอนโทรล

DE = DV - DSP = เออเรอร์คอนโทรล

$DC0$ = เอาต์พุตเออเรอร์ที่ศูนย์

KP = ค่าเกน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อได้เปรียบของสมการนี้ไม่มีรูทีนการหารทางคณิตศาสตร์ ซึ่งค่า KP จะเป็นจำนวนเดียวกับค่าที่ถูกกำหนดออกแบบทางแอนะล็อกคอนโทรล

2) โฟล์วชาร์ตของโหมด P

โดยปกติโฟล์วชาร์ตพรอปพรอชันแนลโหมดจะแสดงตามรูป 3-6 ซึ่งแสดงรูปแบบฟอร์แมตตามอัลกอลิธึม สังเกตว่าพรอปพรอชันแนลเกนสามารถจะเป็นได้ทั้งบวกและลบ นั่นหมายความว่าเออเรอร์อาจจะทำให้ DCP เพิ่มขึ้นหรือลดลง มีโฟล์วชาร์ตซึ่งแสดงรายละเอียดมากขึ้นคือรูปที่ 3-7 โดยได้พิจารณาส่วนอื่นเป็นความสำคัญในโครงสร้างการปฏิบัติในการควบคุมพรอปพรอชันแนล

3)การใช้งานเชิงปฏิบัติ(Practical Application)

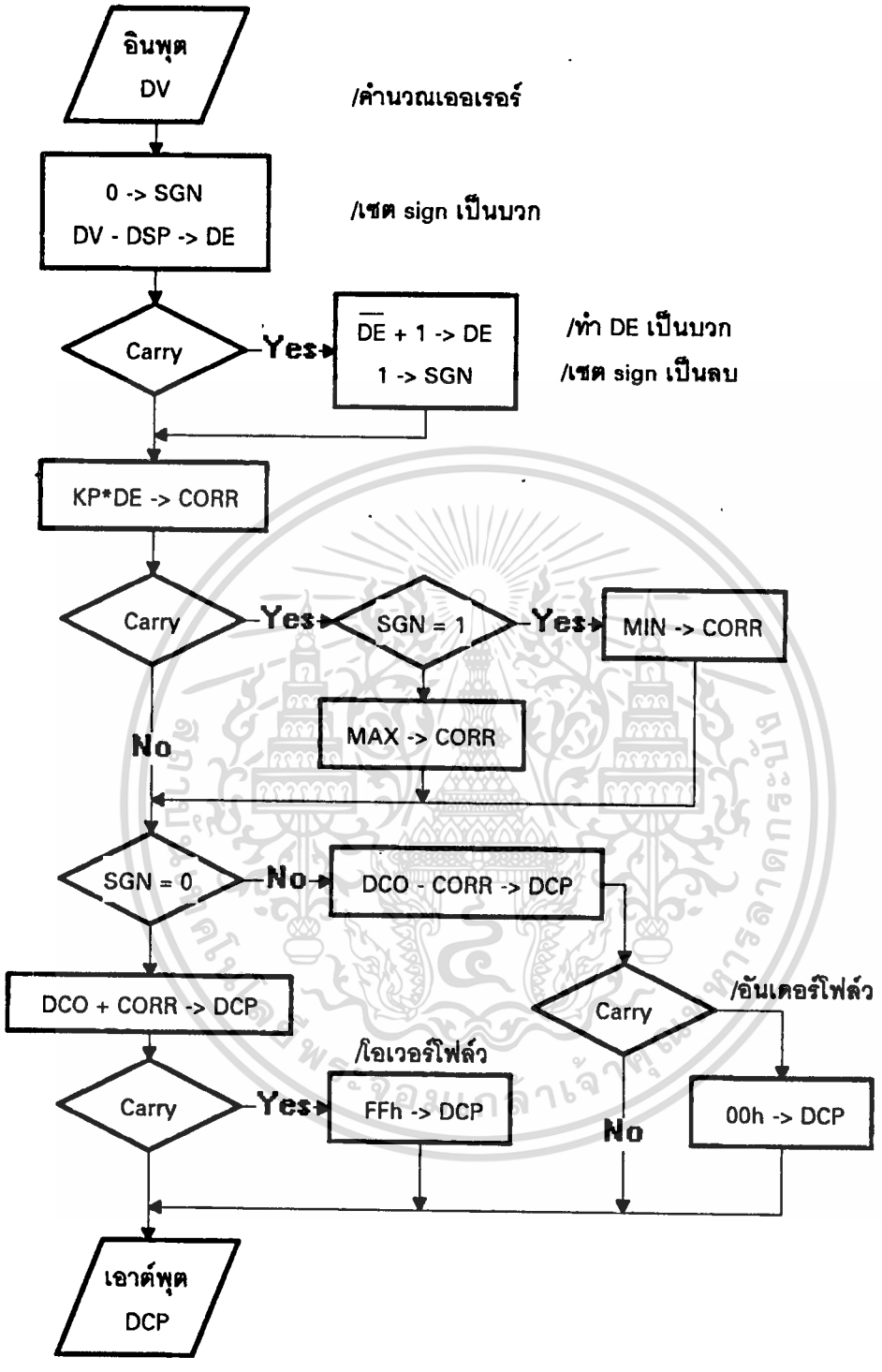
เมื่อเขียนโปรแกรมอย่างจริงจัง เพิ่มเติมจากสมการที่ 3-7 ในพรอฟพอชันนอลโหมดมีข้อพิจารณาในการปฏิบัติมากมายโฟล์วชาร์ตตามรูปที่ 3-7 แสดงเปิดเผยให้เห็นถึงรายละเอียดที่มากขึ้นของอัลกอลิธึมของพรอปพรอชันแนลโหมด ดังต่อไปนี้

1.ขั้วเครื่องหมายของ DE (Polarity of DE) เหมือนกับตอนต้น เป็นความสะดวกที่ใช้ DE แบบไม่มีเครื่องหมายกับเครื่องหมายที่ชี้โดยแฟลก(flag) ในโฟล์วชาร์ตรูป 3-7 โดยได้ร่วมหน้าตานี้เข้ากับอัลกอลิธึมของพรอปพรอชันแนลด้วย

2.ผลของโอเวอร์โฟล์ว(Product overflow) เมื่อ DE ถูกคูณด้วย KP มันเป็นไปได้ทีเดียวจะมีเงื่อนไขของโอเวอร์โฟล์วเกิดขึ้น นั่นหมายความว่าค่าของผลคูณนั้นเกินกว่าค่าไบต์ข้อมูล ถ้าเป็น 8 บิตนั้นแฟลกทด(carry flag)จะถูกเซต ดังรูปที่ 3-7 เมื่อเหตุการณ์นี้เกิดขึ้นจำเป็นต้องเซตค่า $KP \bullet DE$ เป็นแฟกเตอร์คอเร็กชั่น(correction factor)ซึ่งจะเท่ากับค่าสูงสุดและต่ำสุด

3.ผลรวมโอเวอร์โฟล์วและอันเดอร์โฟล์ว (Sum Overflow and Underflow) ในกรณีทั่วไป DCP จะเป็นจำนวนเต็มบวกในช่วงของ $ROUT$ จะแปรไปตั้งแต่ 00 ถึง FFH และแก้ไขให้ถูกต้อง(correction)โดยนำมาลบหรือรวมเข้ากับค่าเออเรอร์ที่ศูนย์($DC0$ ผลลัพท์ของ DCP แสดงในสมการที่ 3-7 โอเวอร์โฟล์วเกิดขึ้นโดยผลรวมเป็นผลลัพท์ที่เกินกว่า 8 บิต และผลลบที่ต่ำกว่า 8 บิตคือน้อยกว่า 00H ซึ่งจำเป็นที่จะให้ซอฟต์แวร์ตรวจสอบโอเวอร์โฟล์วและอันเดอร์โฟล์วเพื่อเตรียมแก้ไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-7 *รายละเอียดโฟล์วชาร์ตของโหมด P แสดงแฟลกเครื่องหมาย และการป้องกันโอเวอร์โฟล์ว อันเดอร์โฟล์ว*

โอเวอร์โฟล์วเกิดขึ้นเมื่อ *DC0* ถูกรวมเข้า และจะทำให้บิตทดถูกเซต อันเดอร์โฟล์วเกิดขึ้นเมื่อ *DC0* ถูกนำไปลบซึ่งบิตทดจะถูกเซตด้วยเช่นกันแสดงดังรูปที่ 3-7 การตอบสนองของอันเดอร์โฟล์ว และโอเวอร์โฟล์วคือ ค่าสูงสุดและต่ำสุดเป็น 00H และ FFH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.การกระทำแบบตรงและแบบผกผัน(Direct or Reverse Action) เครื่องหมายของ KP จะเป็นตัวกำหนดทิศทางของการควบคุมเป็นแบบตรงหรือผกผัน แบบตรงหมายความว่า ถ้าเออเรอร์เป็นค่าบวกผลลัพท์ของตัวแปรเอาต์พุตจะเพิ่มขึ้น และในทางตรงกันข้ามจะลดลงถ้าเป็นแบบผกผัน ซึ่งเป็นเหตุให้ต้องระวังเครื่องหมายของ KP ในอัลกอลิธึม การกระทำแบบตรงและแบบผกผันนั้นสามารถเตรียมตัวแปรที่กำหนดในบล็อก โดยเมื่อมีการบวกหรือ ลบกับ $DC0$ จะกระทำตามนั้น แต่ถ้าเป็นแบบผกผันจะการกระทำบวกหรือลบจะทำในทางตรงกันข้าม

3.5.2 โหมดอินทีกรัลหรือโหมดรีเซต(Integral Mode or Reset))

ในบทแรกโหมดอินทีกรัลถูกกำหนดเหมือนการตอบสนองการควบคุม โดยจะขึ้นอยู่กับค่าเออเรอร์ในอดีตมากกว่าเออเรอร์ของตัวมันเองในขณะนั้น ในอดีต(history)ถูกวัดได้โดยพื้นที่ในส่วนที่กำหนด หรือเรียกว่า "พื้นที่สุทธิ(Net area)" ซึ่งเออเรอร์ที่วัดได้จะอยู่ในเส้นโค้งของเวลาระหว่างเวลาเริ่มต้นกับเวลาปัจจุบัน ในทางคณิตศาสตร์จะเท่ากับการหาค่าอินทีกรัลของเออเรอร์ ซึ่งเหตุนี้จึงถูกขนานนามว่าอินทีกรัลโหมด และมันถูกเรียกว่ารีเซตเพราะว่าโหมดอินทีกรัลเป็นความจุของออฟเซตเออเรอร์ในพรอปพรอชันแนลโหมด โดยที่ "รีเซต" คือเทอมของเออเรอร์ที่ศูนย์

สมการของการคำนวณของโหมดอินทีกรัลคือ

$$C_I(t) = K_I A_e(t) + C_I(0)$$

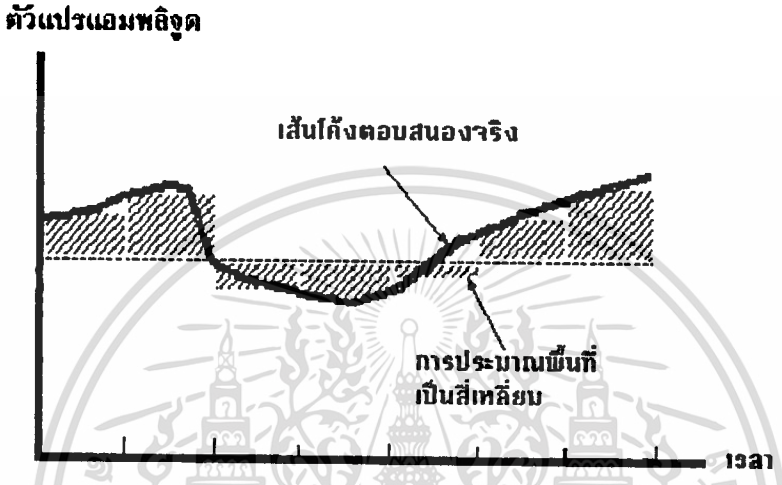
ซึ่ง K_I คืออินทีกรัลเกน(integal gain) และ $A_e(t)$ คือพื้นที่เออเรอร์(error area) ซึ่งแปรผันตามเวลา ซึ่งความสำพันธ์ของพื้นที่มีความสำคัญคือ พื้นที่จะเพิ่มขึ้นเมื่อเออเรอร์เป็นบวก(positive) และพื้นที่จะลดลงเมื่อเออเรอร์เป็นลบ

1. การประมาณพื้นที่ของเออเรอร์(Error Area Approximation)

วิธีเพิ่มเติมโหมดนี้โดยใช้คอมพิวเตอร์ ซึ่งจะต้องหาพื้นที่โดยใช้ค่าตัวแปรอินพุตแซมปลิง มีหลายวิธีที่จะทำเช่นนี้ ในเอกสารฉบับนี้จะใช้วิธีที่ง่ายคือ การอินทีกรัลแบบรูปสี่เหลี่ยม(rectangular integration) แสดงตัวอย่างรูปแบบในรูปที่ 3-8 อินพุตของคอมพิวเตอร์เป็นส่วนประกอบของแซมปลิงที่จุดเส้นโค้งของตัวแปรควบคุม(controlled varialble)กับเวลา เออเรอร์ในแต่ละกรณีจะถูกหาโดยวัดความแตกต่างของค่าและเซตพอย์น รูปที่ 3-8 เป็นการแสดงด้วยว่าแต่ละค่าของเออเรอร์ได้ใช้สร้างเป็นรูปสี่เหลี่ยมมีเซตพอย์นเป็นฐานอ้างอิง ดังนั้นจุดสูงสุดของรูปสี่เหลี่ยมคือค่าเออเรอร์ และความกว้างคือเวลาแซมปลิง เมื่อค่าตัวแปรมากเกินเซตพอย์นโดยที่ค่าเออเรอร์จะเป็นบวก และพื้น-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ใต้กราฟก็จะเป็นบวก ในช่วงความกว้างของเวลาแซมปลิง เมื่อตัวแปรมีค่าน้อยกว่าเซตพอย์น ค่าเออเรอร์จะเป็นลบและพื้นที่จะเป็นค่าลบเช่นกัน พื้นที่สุทธิ(net area)จะเป็นผลรวมระหว่างจำนวนบวกหรือจำนวนลบของพื้นที่ ซึ่งเป็นการประมาณค่าพื้นที่ของเส้นโค้งกับเวลาดังนั้นความเที่ยงตรงจะขึ้นอยู่กับการกำหนดค่าเวลาแซมปลิงให้เล็กที่สุด



รูปที่ 3-8 *กราฟแสดงตัวอย่างพื้นที่เออเรอร์ต่อเวลาโดยประมาณพื้นที่เป็นสี่เหลี่ยม*

2.การกำหนดเวลาแซมปลิง(Sample Time)

เวลาแซมปลิงที่น้อยที่สุดถูกปรารถนาที่จะใช้ในการประมาณค่าของพื้นที่ เพื่อให้ได้เที่ยงตรงมากที่สุด ความคิดที่ว่าจะให้เวลาแซมปลิงน้อยที่สุดนั้นประกอบ ด้วยความถี่ในการแซมปลิงต้องให้มากที่สุดที่จะเป็นไปได้ โดยข้อมูลที่แซมปลิงมาจะไม่เออเรอร์ ดังนั้นความถี่การแซมปลิงต้องประมาณ 10 เท่าของความถี่สูงที่สุดเพื่อหลีกเลี่ยงค่าเออเรอร์ในคาบเวลา และให้เหมาะกับกับโครงสร้างของพื้นที่ด้วย เพื่อให้ค่าเฉลี่ยของสัญญาณจะไม่เปลี่ยนแปลงมากในแต่ละคาบ ดั้งนั้นการประมาณค่าในแต่ละคาบจะเที่ยงตรงทีเดียว

3. อัลกอลิธึมของโหมดอินทีกรัล

ทั้งหมดยังคงเหลือว่าพื้นที่ซึ่งหามาเป็นสี่เหลี่ยมนั้น จะใช้ในการคำนวณในโหมดอินทีกรัลได้อย่างไร ถ้าตัวแปรทั้งหมดถูกกำหนดเป็นพื้นที่สุทธิใต้เส้นโค้ง ของเวลาแล้วสมการที่เหมาะสมในอินทีกรัลโหมดสำหรับระบบดิจิทัลคือ

$$DCI = KI \bullet AREA \bullet ROUT \qquad (3\text{-}8)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ DCI = เอาต์พุตอินทีกรัลโหมด

KI = เกนอินทีกรัลโหมด

$AREA$ = การประมาณพื้นที่ของเส้นโค้ง

$ROUT$ = ช่วงตัวแปรเอาต์พุต

สังเกตว่าไม่มีค่าคงที่รวมอยู่ในสมการนี้เป็นเพราะว่าในคอมพิวเตอร์นั้นค่า DCI จะแทนเป็นเรจิสเตอร์ หรือตำแหน่งของหน่วยความจำ โดยที่ถูกคำนวณจะถูกเปลี่ยนแปลงค่าไปในแต่ละค่าพื้นที่แซมปลิง ถ้ามีค่าเริ่มต้นต้องโหลดลงไปในหน่วยความจำก่อนที่โปรแกรมเริ่มต้นทำงาน

สมการที่ 3-8 สมมุติว่าช่วงของคอนโทรลเลอร์ถูกกำหนดโดย $ROUT$ ดังนั้น $KI \bullet AREA$ แทนสัดส่วนช่วง(range)ซึ่งเป็นผลลัพท์จากโหมดอินทีกรัล ถ้าพื้นที่สุทธิทั้งหมดเป็นศูนย์เทอมสมการนี้จะไม่มีผลลัพท์ใดๆทั้งสิ้น

การประมาณพื้นที่ถูกคำนวณ โดยการรวมพื้นที่สี่เหลี่ยมทั้งหมดจากเวลาซึ่งกระบวนการเริ่มสตาร์ตเมื่อ t = 0 ดังนี้

$$AREA = AREA + (DFE \bullet DT) \tag{3-9}$$

เมื่อ DFE คือเออเรอร์ซึ่งเป็นสัดส่วนกับช่วง(range)

DT คือเวลาระหว่างการแซมปลิง

ความคิดของสมการอย่างหลังนี้ เป็นพื้นที่ถูกปรับปรุงด้วยแต่ละแซมปลิง โดยการรวมพื้นที่ซึ่งแทนโดยพื้นที่เดิม ซึ่งถูกคำนวณก่อนหน้านี้

4.การใช้สมการของโหมดอินทีกรัลให้ง่ายขึ้น(Simplication)

นำความคิดก่อนหน้าของเราโดย การใช้ช่วงอินพุตและเอาต์พุตเหมือนกัน ซึ่งสามารถทำสมการเดิมให้ง่ายลง ซึ่งทำแบบนี้เราต้องการทำให้ สมการนี้แสดงช่วง RIN และ $ROUT$ อย่างชัดแจ้งด้วย และกำหนดตัวแปรใหม่คือ SUM แทนผลบวกของ ค่าเออเรอร์ทั้งหมด

$$SUM = SUM + DE \tag{3-10}$$

ดังนั้นพื้นที่จะแสดงอยู่ใต้รูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$AREA = SUM \bullet DT / RIN \quad (3\text{-}11)$$

สมการที่ 3-11 แทนในสมการ 3-8 เป็นผลดังนี้

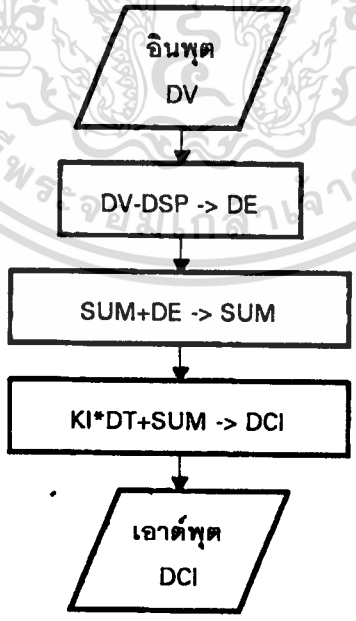
$$DCI = KI \bullet SUM \bullet DT \bullet ROUT / RIN \quad (3\text{-}12)$$

ดังนั้นถ้าทำช่วงให้เท่ากันโดยให้ $RIN = ROUT$ สมการทั้งหมดของโหมดอินทีกรัลจะได้เป็นดังนี้

$$DE = DV - DSP \quad (3\text{-}3)$$

$$SUM = SUM + DE \quad (3\text{-}10)$$

$$DCI = KI \bullet SUM \bullet DT \quad (3\text{-}13)$$



รูปที่ 3-9 *โฟล์วชาร์ตโดยทั่วไปของโหมดอินทีกรัล*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแทนสมการแบบนี้จะเห็นได้ว่า KI เหมือนกับว่าเป็นตัวลดทอนของระบบแอนะล็อก โฟล์วชาร์ตสำหรับเซตของสมการถูกกำหนดโดยรูปที่ 3-9 ซึ่งเหล่านี้มีผลในการประยุกต์ใช้ในงานควบคุมโดยตรง

5.รายละเอียดของโฟล์วชาร์ต(Detailed Flowchart)

รายละเอียดโฟล์วชาร์ตของโหมดอินทีกรัล ต้องรวมการป้องกันการเกิดโอเวอร์โฟล์ว และอันเดอร์โฟล์ว รูปที่ 3-10 แสดงโฟล์วชาร์ตซึ่งรวมเอาไว้ทั้งหมด เครื่องหมายของ DE ใช้วิธีเดียวกับที่ได้อธิบายรูป 3-5 ไปแล้วด้วยค่าแฟลกเครื่องหมาย SGN และ DE ซึ่งไม่มีเครื่องหมาย

6.โอเวอร์โฟล์วและอันเดอร์โฟล์วของ SUM

โปรแกรมโหมดอินทีกรัล โอเวอร์โฟล์ว และอันเดอร์โฟล์วคือตำแหน่งในหน่วยความจำ ซึ่งค่าตัวแปรจะเพิ่มขึ้นเมื่อเออเรอร์เป็นบวก(positive) และลดลงเมื่อเออเรอร์เป็นลบ(negative)ซึ่งกระบวนการแบบต่อเนื่องมีโอกาสข้อมูลของเออเรอร์บวก ทำให้ค่า SUM ถูกนับจนเต็มจำนวนบิตที่ใช้ในกรณีนี้ ผลบวกของ SUM จะทำให้โอเวอร์โฟล์ว ผลลัพท์ของกรณีนี้ SUM จะถูกลดทอนลงมาโดยแทนจำนวนสูงสุดเข้าไปใน SUM เหตุนี้สามารถเป็นเงื่อนไขสำหรับในระบบควบคุมได้

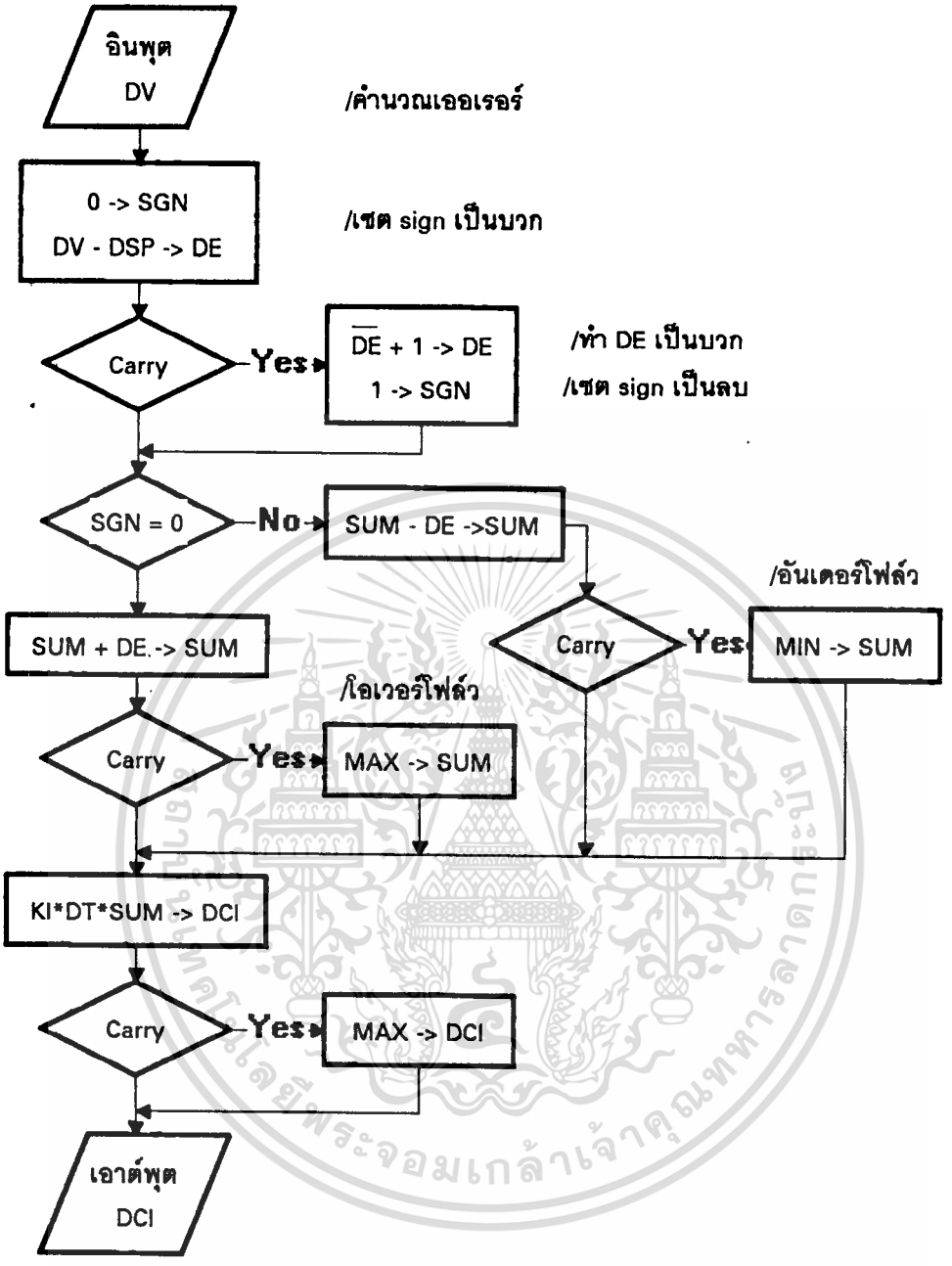
ผลลัพท์ของอันเดอร์โฟล์วนั้น เมื่อการผลของค่าเออเรอร์จาก $SUM \bullet KI$ ไม่สามารถเป็นจำนวนลบได้ ซึ่งการนับในหน่วยความจำเป็นศูนย์ ถึง จำนวนสูงสุดเท่านั้น อย่างไรก็ดีถ้าค่าเออเรอร์ยังคงเป็นลบเกินค่าน้อยที่สุดของ SUM จะเป็นอันเดอร์โฟล์วจะทำให้จำนวน SUM เพิ่มมากขึ้นจากค่าน้อยที่สุด แต่ทิศทางของการรวมจะกลายเป็นตรงกันข้ามแทน

ถ้าสังเกตรูป 3-10 อันเดอร์โฟล์วนั้นจะถูกกำหนดโดยแฟลกทด ซึ่งแปรตามการลบของ DE จาก SUM และโอเวอร์โฟล์วจะถูกกำหนดโดยแฟลกทด ซึ่งแปรตามการบวกของ DE เช่นกัน ในทั้งสองกรณี SUM ต้องถูกเซตให้จำกัดค่าไว้ ทั่วๆไปจะเป็นจำนวนจากสูงสุดถึงต่ำสุด โดยไม่มีเครื่องหมาย ถ้าเป็น 8 บิต $SUMMIN$ = 00H และ $SUMMAX$ = FFH

7.ขนาดของ SUM

ในการปฏิบัติอาจจำเป็นจะต้องเพิ่มความเที่ยงตรงสำหรับการรวบรวมของ SUM นี่เพราะว่าผลของ KI และ DT ซึ่งจะสมทบเข้ากับ SUM เป็นจำนวนทศนิยม ดังนั้นถ้า SUM เป็น 8 บิตแล้ว $KCI = KI \bullet DT \bullet SUM$ จะไม่เข้าใกล้ FFH ซึ่งหมายความว่าเอาต์พุตของตัวควบคุมจะไม่เป็นค่าสูงสุดเลยในโหมดอินทีกรัลที่เราใช้นี้ ถ้าเราเพิ่มความเที่ยงตรงของ SUM เป็นจำนวน 16 บิต ซึ่งมันเป็นไปได้ที่จะมีเอาต์พุตคอนโทรลได้สูงสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-10 *รายละเอียดโฟล์วชาร์ตของโหมดอินทีกรัล แสดงถึงแอคคิวมูเลเตอร์ SUM และ การป้องกันโอเวอร์โฟล์ว อันเดอร์โฟล์ว*

8. การกระทำแบบตรงและผกผัน(Direct and Reverse Action)

ในรูปแบบที่ได้นำเสนอมา นั้นได้สมมุติฐานได้ว่าพื้นที่ซึ่งเพิ่มขึ้นต้องการเอาต์พุตที่เพิ่มขึ้น นี่คือการกระทำแบบตรง ในหลายกรณีพื้นที่เพิ่มขึ้นต้องการเอาต์พุตลดลง นี่เป็นการกระทำแบบผกผัน ซึ่งสามารถเตรียมบล็อกไดอะแกรม สำหรับการรวมและการลบของเออเรอร์กับ *SUM* ในโฟล์วชาร์ตของรูป 3-10 ได้แสดงสำหรับแบบตรงซึ่งเออเรอร์เป็นบวก($SGN = 0$)เป็นเหตุให้ *SUM*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพิ่มขึ้นสำหรับการทำแบบผกผันได้เตรียมบล็อกไดอะแกรมไว้ เพื่อว่าเออเรอร์บวกทำให้ DE ไปลบออกจาก SUM

9.ปัญหาการในการเริ่มต้น(Start-up Problem)

หนึ่งในจำนวนปัญหาความยุ่งยากทั้งหมด กับการใช้โหมดอินทีกรัล ในระหว่างกระบวนการเริ่มต้น เช่น ระบบการควบคุมอุณหภูมิ ต้องเข้าไปพัวพันกับค่าความร้อน ซึ่งเป็นเงื่อนไขเริ่มต้นระหว่างเวลา โดยที่กำลังเพิ่มความร้อนเพิ่ม เพื่อจะเข้าสู่เซตพอย์น เทอมอินทิกรัลจะคำนวนพื้นที่ด้วยขั้วคงที่(เป็นเครื่องหมาย + หรือ - ดังเดิม ผลก็คือ จะเกิดโอเวอร์ชูต(overshoot)ขนาดใหญ่ที่เอาต์พุตอย่างไม่เหมาะสม ในกรณีเช่นนี้ต้องงดการทำโหมดอินทีกรัลชั่วคราว จนกระทั่งระบบเข้าสู่อุณหภูมิ หรือให้ค่าออฟเซตของพื้นที่ในขณะเริ่มต้นมีความแตกต่างน้อยที่สุด

3.5.3 โหมดดีริเวทีฟ(Derivative Mode or Rate)

โหมดดีริเวทีฟของระบบควบคุมแบบป้อนกลับ ซึ่งจะขึ้นอยู่กับอัตราการเปลี่ยนแปลงของค่าเออเรอร์ การกำหนดรูปแบบของโหมดนี้คือ การกำหนดการคำนวนแบบอนุพันธ์ ในการใช้กับระบบคอมพิวเตอร์ค่าที่ใช้งานมีเพียงตัวแปรคาบเวลาแซมปลิงและตัวแปรควบคุม ดังนั้นเพื่อความเหมาะสม จะกำหนดโหมดจากสมการที่กำหนดไว้เดิมต้นต้นคือ

$$c_D(t) = \frac{K_D[e_P(t) - e_P(t_0)]}{t - t_0}$$

จึงเปลี่ยนบ้างเทอมให้เหมาะสมเช่นค่าคาบเวลา

ในโหมดดีริเวทีฟจะไม่ใช้ทำงานตามลำพังโดยไม่มีการป้อนกลับ คือต้องมีการเปลี่ยนแปลงค่าเออเรอร์ตลอดเวลา ถ้าค่าเออเรอร์เป็นค่าคงที่จะไม่สามารถทำงานได้ง่ายถูกต้อง

1. อัลกอลิธึมของดีริเวทีฟโหมด

การประยุกต์ใช้โหมดดีริเวทีฟกับระบบควบคุมโดยคอมพิวเตอร์ จะมีอัลกอลิธึมหลายอย่างที่แตกต่างกัน โดยวิธีอนุพันธ์จะสามารถจัดการให้ง่ายสำหรับตัวแปร ซึ่งง่ายที่สุดคือ การใช้สมการข้างต้นโดยตรง ในกรณีทั่วไปโหมดดีริเวทีฟจำกำหนดรูปแบบสมการโดย

$$DCD = \frac{KD \bullet (DFE - DFE0) \bullet ROUT}{DT} \quad (3\text{-}14)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ซึ่ง DCD คือเอาต์พุตคอนโทรลของโหมดดีริเวทีฟ

KD คือดีริเวทีฟเกน

DFE คือเออเรอร์อินพุตในสัดส่วนของช่วง

$DFE0$ คือเออเรอร์เดิมในสัดส่วนของช่วง

$ROUT$ คือช่วงของเอาต์พุต

DT คือเวลาแซมปลิง

2.การใช้สมการในโหมดดีริเวทีฟให้ง่ายขึ้น(Simplification)

ในสองกรณีดังกล่าวเป็นไปได้ ที่ทำให้สมการง่ายลงสำหรับโหมดการควบคุม ถ้าช่วงอินพุต RIN และช่วงเอาต์พุต $ROUT$ เท่ากัน ดูจากสมการที่ 3-14 เห็นได้ว่าจะเขียนตามแฟชั่นนิยมเป็นดังนี้ อย่างแรก $DFE = DE / RIN$ และ $DFE0 = DE0 / RIN$ ที่ซึ่ง $DE0$ เป็นค่าเออเรอร์เดิม ถ้าสมการเหล่านี้ลบกันแทนในสมการที่ 3-14 ผลลัพท์จะเป็น

$$DCD = \frac{KD \bullet (DE - DE0) \bullet ROUT}{DT \bullet RIN}$$

ขณะนี้ถ้า $RIN = ROUT$ สมการนี้จะลดรูปลงเหลือ

$$DCD = \frac{KD \bullet DDE}{DT} \qquad (3\text{-}15)$$

$$DDE = DE - DE0 \qquad (3\text{-}16)$$

ผลการทำให้สมการกระทัดรัดลงดังกล่าว ถ้าเราเขียนเทอมเออเรอร์แยกจากตัวมันจะได้ $DE = DV - DSP$ และ $DE0 = DV0 - DSP$ ที่ซึ่ง $DV0$ คือตัวแปรควบคุมของแซมปลิงเดิม ดังนั้นสมการที่ 3-16 จะเป็นดังนี้

$$DDE = DV - DV0 \qquad (3\text{-}17)$$

ทั้งสองสมการสามารถใช้สนับสนุนการคำนวณ กรณีทั่วไปจะพบว่าง่ายกว่าถ้าใช้สมการที่ 3-17 เพื่อหาเทอมอนุพันธ์ นี้เป็นเพราะว่าเมื่อผลต่างระหว่าง DE และ $DE0$ มีความเป็นไปได้ถึงสี่ลักษณะ เครื่องหมายด้วยกันซึ่งจะเป็นผลให้ DDE ติดเครื่องหมายด้วย แต่ DV และ $DV0$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นปริมาณบวก(00H สู่ FFH) โดยเครื่องหมายจะกำหนดโดยผลลบเพียงเท่านั้น โฟล์วชาร์ตรูป 3-11 แสดงการประยุกต์โดยตรงของโหมดดีริเวทีฟ ซึ่งแน่นอนทีเดียวว่าโหมดนี้ไม่สามารถใช้เพียงลำพังโดยไม่มีการป้อนกลับ ซึ่งจะทำให้ค่าเออเรอร์นั้นคงที่ สังเกตได้ว่าไม่จำเป็นต้องเก็บค่า DV เดิมไว้สำหรับ ใช้ในการคำนวณในขั้นตอนต่อไป



รูปที่ 3-11 *โฟล์วชาร์ตสำหรับโหมดดีริเวทีฟแสดงถึงการกำหนดเครื่องหมาย*

3. ขั้วเครื่องหมายของอนุพันธ์(Derivative Polarity)

ผลลัพท์ของการอนุมาณการปฏิบัติโดยวิธีอนุพันธ์นั้น จะเป็นไปได้ทั้งบวกและลบ ซึ่งถ้าเป็นผลลบจะแทนด้วย 2's คอมพลีเมนต์ ในการเก็บมาเพื่อต้องการใช้จำนวนตัวเลขไม่มีเครื่องหมายจำเป็นอย่างยิ่งต้องเปลี่ยนเป็นจำนวนบวก โดยใช้เครื่องหมายเป็นหลัก นี่แสดงตามรูป 3-11 ดังนั้นขณะนี้มีแฟลกเครื่องหมายสองตัว แฟลกเครื่องหมายตัวแรกสำหรับ DE และอีกหนึ่งคือ เครื่องหมายของผลต่างจากการทำอนุพันธ์ DE - $DE0$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. การกำหนดดีริเวทีฟเกน(Derivative Gain)

ค่าดีริเวทีฟเกน แทนด้วยสัดส่วนการเปลี่ยนเอาต์พุตคอนโทรลต่ออัตราการเปลี่ยนค่าเออเรอร์ดิริเวทีฟเกนอยู่ในหน่วยของเวลา บ่อยครั้งเมื่อเราให้สมการ KD / DT เป็นเหมือนค่าคงที่ตัวหนึ่งคูณกับเทอมของผลต่าง ที่สำคัญหน่วยของ KD และ DT ต้องเข้ากันได้

5. การกระทำแบบตรงและผกผันในโหมดดีริเวทีฟ(Direct and Reverse Action)

เป็นไปได้สำหรับโหมดดีริเวทีฟทั้งกระทำแบบตรงและผกผัน แบบตรงหมายความว่าอัตราบวก(positive rate)ของการเปลี่ยนแปรค่าควบคุมเป็นผลทำให้เอาต์พุตเพิ่มขึ้น แบบผกผันหมายความว่า สโลปบวก(positive slop)ในเวลานั้นจะทำให้เอาต์พุตควบคุมลดลง นี่สามารถเตรียมได้ โดยกำหนดบล็อกไว้ทั้งบวก และลบ ให้กับเทอมของเอาต์พุตบล็อกของโฟล์วชาร์ต โดยบล็อกนี้ขึ้นอยู่กับสอง เครื่องหมายซึ่งเครื่องหมายแรกคือ DDE ถ้าอีกเครื่องหมายหนึ่งเหมือนกันก็จะกระทำเป็นแบบบวก และแบบลบ ถ้าตรงกันข้าม

3.5.4 โหมดคอมโพสิต(Composite Mode)

โดยทั่วไประบบควบคุมแบบต่อเนื่องที่ดีที่สุด เมื่อได้ร่วมการทำงานระหว่างโหมดเดี่ยว(pure mode) ดังกล่าวข้างต้นมารวมกันเป็น 3 แบบด้วยกันคือ

(1) PI เป็นการร่วมกันทำงานระหว่างพรอปพรอชันแนลและอินทีกรัล

(2) PD เป็นการร่วมกันทำงานระหว่างพรอปพรอชันแนลและดีริเวทีฟ

(3) PID เป็นการรวม 3 โหมด ด้วยกันคือพรอปพรอชันแนล อินทีกรัล และดีริเวทีฟ

มีข้อสงสัยว่า *"ทำไมเพิ่งจะใช้คอนโทรลเลอร์แบบคอมโพสิต PID ในการประยุกต์"* ในบางการประยุกต์ที่ PI ,PD และโหมดเดี่ยว เช่น P,I,D สามารถถูกใช้งานแทน PID ได้ดีกว่าเพราะว่า ประหยัดค่าใช้จ่ายทางเวลา และความพยายามที่เลือกค่าเหมาะสม เพื่อสำหรับจะให้ระบบเสถียรนั้นยากกว่า มีตัวอย่างหลายกระบวนการยังไม่ได้เตรียมการปรับปรุงระบบเป็น PID แต่จะใช้รวมกันระหว่างการทำดีริเวทีฟกับโหมด PI โหมดคอมโพสิตแบ่งได้เป็นโหมดย่อยดังนี้

1. การควบคุมแบบพรอปพรอชันแนล-อินทีกรัล(Proportional-integral control)

2. การทำ PID คอนโทรลเลอร์(PID controller action)

1. การควบคุมแบบพรอปพรอชันแนล-อินทีกรัล(Proportional-Integral Control)

ใช้การทำให้พรอปพรอชันแนลและ อินทีกรัลรวมเข้าด้วยกันนั้นธรรมดามาก เหตุผลหนึ่งคือให้มีอินทีกรัลคือ ระบบมีความสามารถปรับตัวเองในเงื่อนไขของโหลดใหม่โดยพยายามให้เออเรอร์เป็นศูนย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัลกอลิธึมของ PI

เนื้อแท้ของโหมด PI นั้น คือการรวมระหว่างพรอปพรอชันแนลและอินทีกรัล ซึ่งได้บรรยายไว้ก่อนหน้านี้ อย่างไรก็ตามเราได้พบว่าโครงสร้างที่เหมาะสมกับโหมดคอมโพสิต คือ การนำเทอมพรอปพรอชันแนลคูณกับเออเรอร์และพื้นที่เออเรอร์ของเทอมอินทีกรัล ดังนั้นในขณะใดขณะหนึ่งเอาต์พุตคอนโทรลประกอบขึ้นด้วยทั้งสองเทอม เทอมหนึ่งได้เนื่องจากการคำนวณของพื้นที่ต่อเวลาแซมปลิง และอีกเทอมหนึ่งได้เนื่องจากเออเรอร์ตามปัจจุบัน ซึ่งเป็นไปได้ที่จะเขียนเป็นสมการเดียวในโหมดนี้ดังนี้

$$DCPI = (KP \bullet DE) + (KP \bullet KI \bullet DT \bullet SUM) \qquad (3\text{-}15)$$

ซึ่ง $DCPI$ คือเอาต์พุตควบคุมของโหมด PI

KP คือพรอปพรอชันแนลเกน

DE คือเออเรอร์

KI คืออินทีกรัลเกน

DT คือเวลาแซมปลิง

SUM คือผลรวมของเออเรอร์ล่าสุด

สังเกตว่าถ้าเออเรอร์สำหรับเฉพาะบางตัวอย่างเป็นศูนย์ แล้วเอาต์พุตตามสมการ 3-15 จะเป็นผลลัพท์จากเทอมอินทีกรัล เทอมนี้คือผลลัพท์ของเออเรอร์ในอดีตของระบบสู่เทอมของ SUM จะเห็นได้ว่าเทอมอินทีกรัลนั้นจะเตรียมต่อบสนองเออเรอร์ศูนย์(zero error)ของระบบ

แล้วหลักสำคัญคือ โครงสร้างแบบอย่าง ซึ่งถูกใช้โดยสมการที่ 3-15 ในการกำหนดค่าคอนโทรลเอาต์พุต(control output) อย่างไรก็ดีที่จริงการนำไปปฏิบัติของสมการที่ 3-15 กับการปฏิบัติจริงโดยใช้ไมโครโปรเซสเซอร์สร้างนั้นยากมาก

เซตของสมการโหมด PI สมการ 3-15 กำหนดการตอบสนองของ PI แต่ไม่สามารถใส่แทนค่าโดยตรงเข้าสู่ไมโครโปรเซสเซอร์ การใส่รหัสนั้นต้องแตกสมการเป็นหลายขั้นตอน ที่ซึ่งไมโครโปรเซสเซอร์สามารถปฏิบัติได้เฉพาะเลขหนึ่งชุดที่เวลาหนึ่ง ซึ่งมีเซตของสมการดังนี้

$$DE = DV - DSP \qquad (3\text{-}3)$$

$$SUM = SUM + DE \qquad (3\text{-}10)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$DCI = KP \bullet KI \bullet DT \bullet SUM \quad (3\text{-}16)$$

$$DCP = KP \bullet DE$$

$$DCPI = DCI + DCP \quad (3\text{-}17)$$

ในแต่ละสมการสามารถแก้ปัญหาโดยคำสั่งแบบง่ายๆในคอมพิวเตอร์ ค่าคงที่ในสมการที่ 3-16 คือ $KP \bullet KI \bullet DT$ ซึ่งแน่นนอนจะถูกรวมเข้าเป็นค่าคงที่ตัวหนึ่ง

อย่างหนึ่งเป็นการพยายามรวมสมการ 3-10 และสมการ 3-16 และเขียนเทอมอินทีกรัลเหมือน $DCI = DCI + (KP \bullet KI \bullet DT \bullet DE)$ มีเหตุผลที่ดีอย่างหนึ่งว่า การเลือกสมการนี้อาจไม่ฉลาดนัก เพราะว่า DE เป็นจำนวนค่าน้อย และจะได้ผลเป็นค่าคงที่น้อยกว่า 1 ได้บ่อยๆ นี่หมายความว่าหลายเออเรอร์จะสร้างความผิดพลาดเข้าสู่ DCI แต่การปฏิบัติการคูณด้วยค่าคงที่นั้น เทอม DE จะหนีจากจำนวนค่าน้อยได้ และสามารถถูกแทนโดยข้อมูลไบต์ได้ โดยที่อย่างแรก DE ต้องบวกกับ SUM ก่อน

โอเวอร์โฟล์วและอันเดอร์โฟล์วของโหมด P I(Overflow and Underflow)

การพิจารณาที่หนักใจที่สุดคือเมื่อโหมด PI ต้องเตรียมป้องกันโอเวอร์โฟล์ว และอันเดอร์โฟล์วของตัวแปรในการคำนวณ ความคิดพื้นฐานในที่นี้คือ $DCPI$ โดยปกติจะครอบคลุมช่วงจำนวนตัวเลขของไบต์ข้อมูลตั้งแต่ 00h ถึง FFh สำหรับไมโครโปรเซสเซอร์แบบ 8 บิต ค่า DCI ยังคงถูกจำกัดอยู่ในขอบเขตด้วยซึ่งไม่มีเออเรอร์ $DCPI$ เท่ากับ DCI แต่ขีดจำกัดบน SUM จะมีผลต่อปริมาณค่าของ DCI

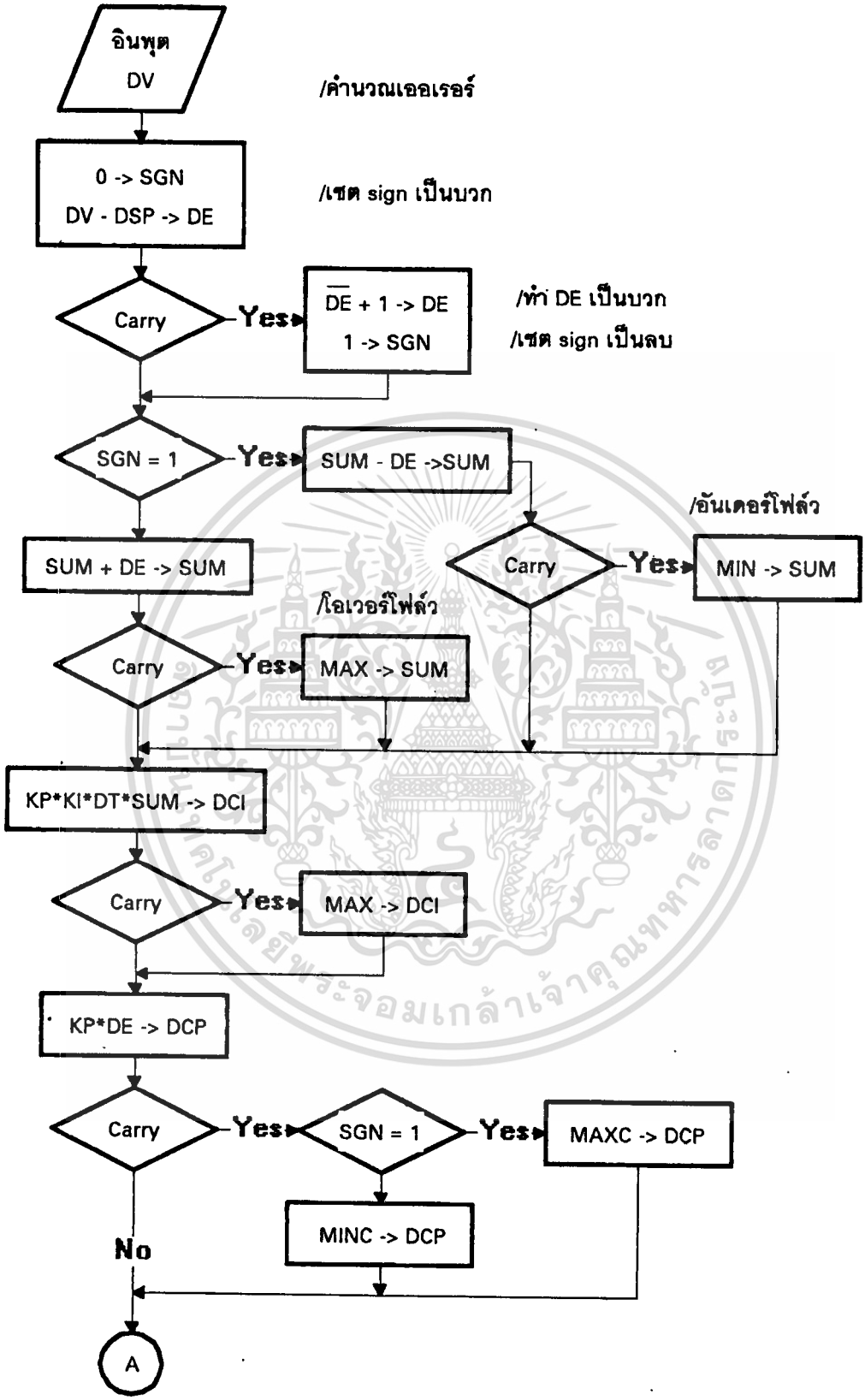
โอเวอร์โฟล์วของปริมาณเหล่านี้เกิดขึ้น เมื่อผลลัพท์จากผลรวมของการรวมในจำนวนที่เกินความเป็น ไปได้ของคอมพิวเตอร์เช่น FFH เป็นแบบของไมโครคอมพิวเตอร์ 8 บิต เมื่อเงื่อนไขเช่นนั้นถูกตรวจสอบตัวแปรนั้น ต้องเซตให้เท่ากับ FFH

อันเดอร์โฟล์วเป็นผลลัพท์จากการลบซึ่ง SUM, DCI หรือ $DCPI$ จะพ้นจากการเป็นจำนวนลบ ถ้าการลบทำให้เทอมเหล่านั้นเป็นจำนวนลบเทอมเหล่านั้นจะถูกเซตเป็นศูนย์

โฟล์วชาร์ต(Flowchart)

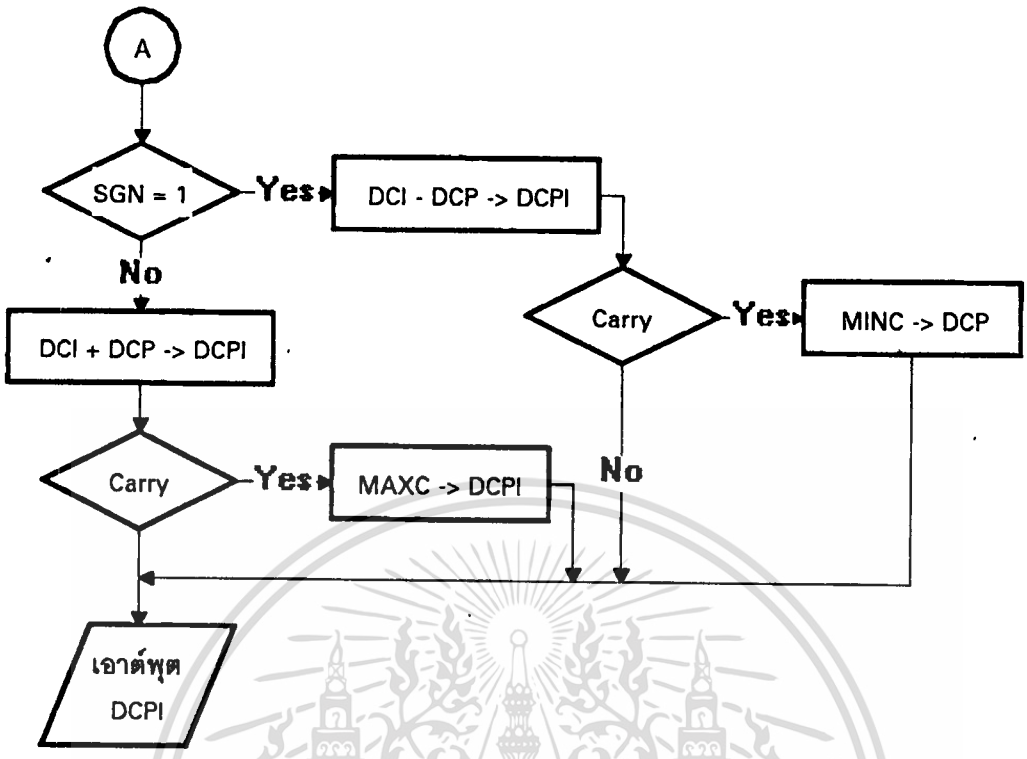
โฟล์วชาร์ตตามรูปที่ 3-12 แสดงส่วนหน้าตาที่จำเป็นของอัลกอลิธึมของ โหมด PI ซึ่งจะรวมเอาส่วนที่สำคัญเช่นการป้องกัน โอเวอร์โฟล์ว อันเดอร์โฟล์ว และการพิจารณาเครื่องหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(a)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(b)

รูปที่ 3-12 *(a),(b) โฟล์วชาร์ตของโหมด PI แสดงถึงการป้องกันโอเวอร์โฟล์ว และอันเดอร์โฟล์ว*

2 การทำ PID คอนโทรลเลอร์(PID Controller Action)

คอนโทรลเลอร์แบบ 3 โหมด ในท้ายสุดใช้ในการควบคุมกระบวนการสำหรับฟังก์ชันถ่ายโอนระบบเฉพาะ ซึ่งไม่สามารถเขียนได้ การปรับจูนมีความซับซ้อนเพราะว่ามีหลายเทอมที่พิจารณา แต่การควบคุมสามารถทำให้ประสบความสำเร็จอย่างใหญ่หลวงในหลายงานประยุกต์ ในบทนี้การเพิ่มเติมโหมด PID โดยคอมพิวเตอร์จะถูกแสดงให้ชัดเจน

อัลกอลิธึมของ PID ในตอนที่แล้วเราได้แนะนำการทำดีริเวทีฟในโหมด PI ไว้แล้ว ซึ่งคล้ายกับกรณีนั้น จึงจำเป็นที่จะต้องใช้พรอปพรอชันแนลเกนเข้าไปคูณกับดีริเวทีฟเทอม ผลลัพท์ของสมการควบคุมจะถูกแบ่งเป็นสามเทอม เทอมแรกเป็นสัดส่วนโดยตรงกับเออเรอร์ เทอมสองขึ้นอยู่กับเออเรอร์ในอดีต และเทอมสุดท้ายขึ้นอยู่กับอัตราการเปลี่ยนแปลงเออเรอร์ในปัจจุบัน สมการแสดงได้ดังนี้

$$DCPID = (KP \bullet DE) + (KP \bullet KI \bullet DT \bullet SUM) + \frac{KP \bullet KD \bullet DDE}{DT} \quad (3\text{-}18)$$

ที่ซึ่ง $DPCID$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

KP คือพรอปพรอชันแนลเกน

DE คือเออเรอร์แปรเปลี่ยน = DV - DSP

KI คืออินทีกรัลเกน

DT คือเวลาแซมปลิง

SUM คือผลบวกเออเรอร์

KD คือดีริเวทีฟเกน

$DDE = DV - DV0$ คือเออเรอร์เปลี่ยนในแต่ละแซมปลิง

สังเกตว่าถ้าเออเรอร์คือ ศูนย์ เอาต์พุตจะถูกกำหนดด้วยผลจาก SUM และการเปลี่ยนของเออเรอร์ในแซมปลิงก่อนหน้านี้

การทำโปรแกรมจริงตามสมการที่ 3-18 นั้นต้องการแตกสมการออกเป็นส่วนๆไม่ขึ้นต่อกันและดูตามแต่ละส่วนเพื่อนำไปประยุกต์

การประยุกต์สมการ การปฏิบัติในการประยุกต์ของการควบคุมแบบ PID ซึ่งต้องการสมการควบคุม 3-18 ถูกแตกเป็นหลายส่วนซึ่งไมโครโปรเซสเซอร์กระทำได้หน่วยส่วนต่อหนึ่งเวลา เซตแบบอย่างควบคุมประกอบด้วยเซตดังนี้

$$DE = DV - DSP$$

$$SUM = SUM + DE$$

$$DDE = DV - DV0$$

$$DV0 = DV$$

$$DVI = KP \bullet KI \bullet DT \bullet SUM$$

$$DCD = \frac{KP \bullet KD \bullet DDE}{DT}$$

$$DCP = KP \bullet DE$$

$$DCPID = DCP + DCI + DCD$$

(3-19)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าคงที่ต่างๆถูกรวมกับเพื่อเพิ่มเติมให้เป็นจริงในสมการ เพื่อเป็นประโยชน์การพิจารณาการปฏิบัติจำเป็นอย่างยิ่ง ในการแทนค่าในสมการสู่การประยุกต์จริง ส่วนใหญ่จะมีความคิดเห็นเหมือนก่อนหน้านี้สำหรับคอนโทรลเลอร์แบบดีริเวทีฟ และ PI เหล่านี้จะถูกระบุในการรวมกับโหมดดีริเวทีฟต่อไป

เครื่องหมายของ *DDE* (polarity of DDE) เทอมเครื่องหมายหรือขั้วของ *DDE* และดังนั้นบล็อกของการบวกหรือการลบ *DCD* จาก *DCI* + *DCP* ต้องอยู่บนเงื่อนไขของเครื่องหมายหรือขั้ว สังเกตว่าถ้ากระทำแบบตรง $KP > 0$ และเป็นอัตราบวก *DDE*. >0 มีความหมายว่า *DCD* จะเพิ่มขึ้น ขณะที่การทำผกผันสำหรับ *DDE* >0 แล้ว *DCD* จะถูกลบ

ปัญหาโอเวอร์โฟล์ว ถ้าค่าคงที่คูณกับ *DDE* มากกว่า 1 แล้วมันจะเป็นไปได้สำหรับ *DCD* โอเวอร์โฟล์ว ด้วยเหตุผลนี้การตรวจสอบโอเวอร์โฟล์วจะต้องเตรียมไว้

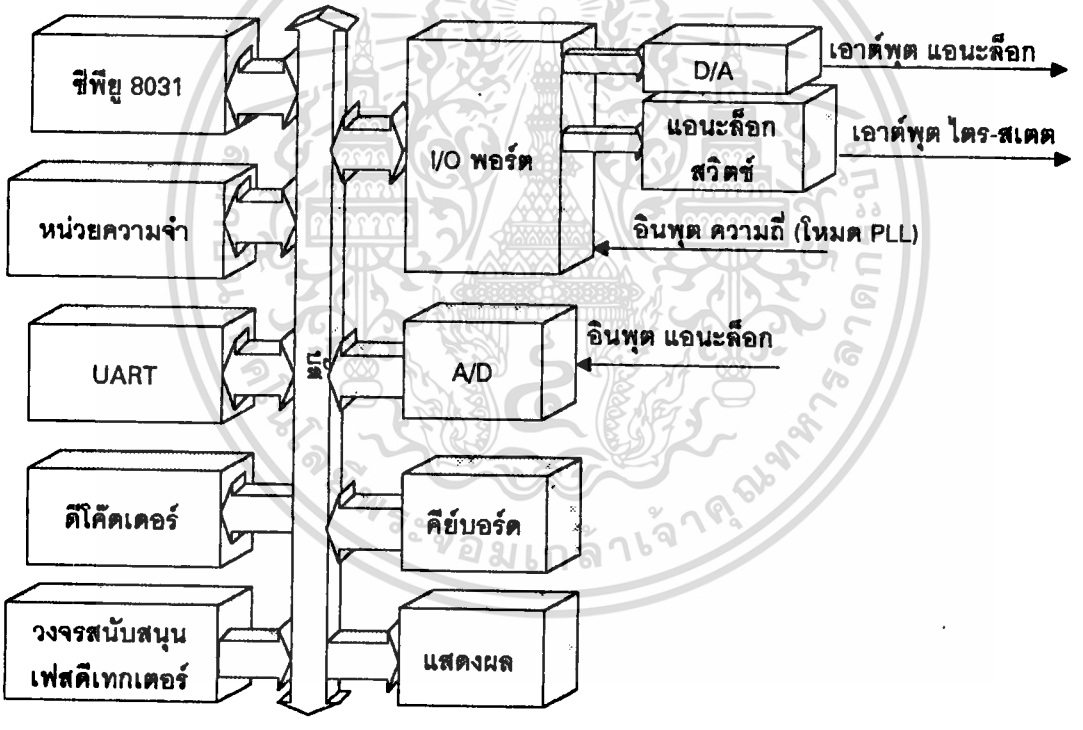
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การออกแบบวงจรตัวดิจิทัลคอนโทรลเลอร์

จุดมุ่งหมายของบทนี้เกี่ยวข้องกับฮาร์ดแวร์ เกี่ยวกับข้อมูลอินพุตสู่ไมโครโปรเซสเซอร์ ข้อมูลเอาต์พุตจากไมโครโปรเซสเซอร์ ซึ่งการแปรผันข้อมูลจะเกี่ยวกับการแปรผันทางด้านควบคุมในแบบต่างกัน

4.1 โครงสร้างของตัวดิจิทัลคอนโทรลเลอร์



รูปที่ 4-1 *แสดงบล็อกไดอะแกรมของตัวดิจิทัลคอนโทรลเลอร์*

ตามรูปที่ 4-1 เป็นบล็อกไดอะแกรมของตัวคอนโทรลเลอร์รวมทั้งหมด โดยเลือกใช้ตัวไมโครโปรเซสเซอร์ 8 บิตเบอร์ 8031 ซึ่งเรียกโดยทั่วไปว่า "ไมโครคอนโทรลเลอร์" ใช้เป็นตัวประมวลผล หรือ ซีพียู ซึ่งไมโครคอนโทรลเลอร์เบอร์นี้มีจุดมุ่งหมายสร้างขึ้น เพื่อให้ใช้ได้งานประยุกต์ต่างๆในทางดิจิทัลโดยไม่ซับซ้อนมากนัก สิ่งสำคัญคือราคาอุปกรณ์ราคาไม่แพง โดยพยายามเลือกอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ละอย่างเพื่อใช้ร่วมกันให้ดีที่สุดและให้ประหยัด ในการประยุกต์ใช้เป็นตัวคอนโทรลเลอร์นี้จำเป็นจะต้องมีอุปกรณ์อินเตอร์เฟสร่วมด้วย ตามรูปมีอุปกรณ์อินเตอร์ เฟสต่างๆที่สำคัญดังนี้

- วงจรคีย์บอร์ด(KBC)
- วงจรแสดงผล(display)
- วงจรซีเรียลอินเตอร์เฟส(UART)
- วงจรแปลงแอนะล็อกเป็นดิจิทัล(ADC)
- วงจรแปลงดิจิทัลเป็นแอนะล็อก(DAC)
- วงจรเฟส/ความถี่ดีเทกเตอร์(PFD)

4.1.1 ลักษณะทางสถาปัตยกรรมของ 8031 โดยย่อ

กรณีแรกในปัญหาคือต้องเรียนรู้เกี่ยวกับการใช้งานกับไมโครคอนโทรลเลอร์เบอร์ ซึ่งจะต้องเรียนรู้ลักษณะกลไกทางฮาร์ดแวร์ต่างๆ การกำหนดจำนวนเรจิสเตอร์(register) และการใช้งานรจิส-เตอร์ ความสามารถของคำสั่งทางซอฟต์แวร์ จนกระทั่งถึงข้อดีต่างๆในการเลือกใช้

1) ฮาร์ดแวร์ของ8051

โดยแท้จริง 8051 เป็นไมโครคอนโทรลเลอร์รวมอยู่ในตระกูลเดียวกัน มีตั้งแต่เบอร์ 8031 ถึง 8751 มีหลายแพคเกจ(package)หาเลือกใช้ได้เช่น แบบ N-Channel Metal Oxide Silicon(NMOS) ไอซีตระกูล 51 เป็นแบบ DIP 40 ขา การจัดขาต่างๆจะเป็นลักษณะเฉพาะตัว ฉะนั้นต้องดูคู่มือประกอบไมโครคอนโทรลเลอร์ 8051 ภายในประกอบด้วย เช่น

- รอมและแรมภายใน
- พอร์ตอินพุตเอาต์พุตและขาโปรแกรม
- ไทเมอร์และตัวนับ
- สื่อสารข้อมูลแบบอนุกรม

และในรูปยังมีส่วนประกอบเกี่ยวข้องกับ ซีพียู คือ โปรแกรมตัวนับ เรจิสเตอร์ใช้งาน และวงจรคล็อก

โครงสร้างทางสถาปัตยกรรมของ 8051 ประกอบด้วยดังนี้:

- 8 บิต ซีพียู กับเรจิสเตอร์ A (accumlator) และ B
- 16 บิตโปรแกรมตัวนับ(PC) และดาต้าพอย์นเตอร์(DPTR)
- 8 บิตเวอร์ดเก็บสถานะโปรแกรม(program status word) PSW
- 8 บิตสแต็กพอย์นเตอร์(SP)

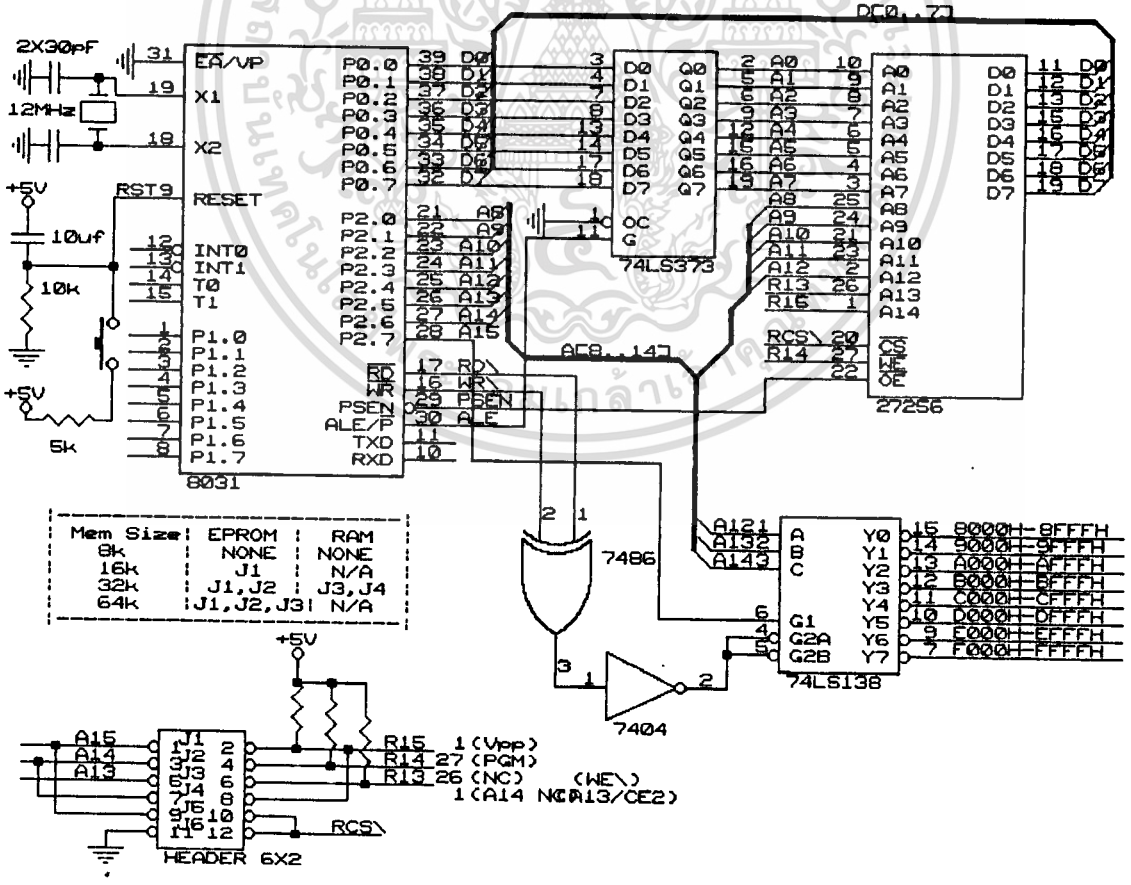
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- รอมหรืออีพรอมภายใน และไม่มีใน 8031 และ 4 กิโลไบต์ ใน 8051
- แรมภายใน 128 ไบต์

เรจิสเตอร์ 4 แบงค์ซึ่งประกอบด้วย 8 เรจิสเตอร์

- 16 ไบต์ซึ่งเป็นแอดเดรสระดับบิต
- 8 ไบต์เป็นดาต้าเมโมรี(data memory)ใช้ทั่วไป
- 32 ขาอินพุตเอาต์พุตจัดโดย 8 บิตพอร์ต P0-P3
- ฟูลดูเพล็กซ์(full duplex) ในการรับส่งสัญญาณอนุกรม
- เรจิสเตอร์ควบคุม TCON,TMOD,SCON,PCON,IP และ IE
- มีอินเตอร์รัพต์จากภายนอกได้สองแหล่ง

โมเดลการโปรแกรมของ 8051 มีเรจิสเตอร์ให้เลือกทั้ง 8 และ 16 บิต อีกทั้ง 8 บิตในหน่วยความจำ ซึ่งเรจิสเตอร์และตำแหน่งในหน่วยความจำเหล่านั้นทำงานโดย ใช้คำสั่งทางซอฟต์แวร์ซึ่งร่วมเข้าด้วยกันกับส่วนที่ออกแบบไว้



รูปที่ 4-2 วงจร 8031 บอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

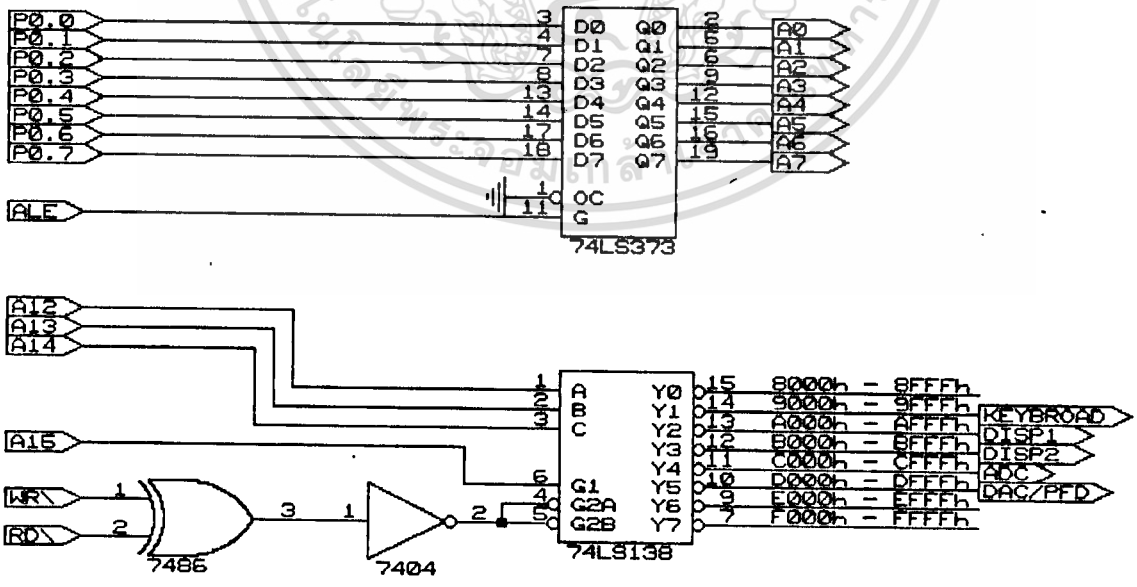
4.1.2 การออกแบบ 8031 ไมโครคอนโทรลเลอร์บอร์ด

ปกติแล้ว IC 8031(ไม่มีรอมภายใน) ไม่สามารถทำงานตามลำพังได้ ซึ่งตัวเองต้องการวงจรของหน่วยความจำภายนอก และวงจร I/O และส่วนอินเตอร์เฟสต่างๆที่นำมาใช้งานโดยเฉพาะ ในการออกแบบบอร์ด 8031 จำเป็นต้องพร้อมที่จะร่วมวงจรเหล่านั้นไปด้วย ซึ่งการออกแบบแบบนั้นต้องทำตามแนวทางวิธี ซึ่งสามารถประหยัด เกิดประโยชน์และยืดหยุ่นในการใช้ได้

1) การถอดรหัสแอดเดรสหน่วยความจำ

หน่วยความจำภายนอกถูกใช้ร่วมกับพอร์ตเบอร์ศูนย์ เป็นดาต้าบัสและแอดเดรสบัสไบต์ต่ำ และพอร์ตเบอร์สองเป็นแอดเดรสบัสไบต์สูง ดาต้าและแอดเดรสไบต์ต่ำถูกทำมัลติเพลกซ์บนพอร์ตเบอร์ศูนย์ ตัวแลทช์แอดเดรสภายนอกเบอร์ 74LS373 ถูกต่อเข้ากับพอร์ตเบอร์ศูนย์ เก็บค่าแอดเดรสไบต์ต่ำ เมื่อไรก็ตามจะเข้าถึงตำแหน่งของหน่วยความจำภายนอก โดยที่ 8031 จะส่งพัลส์ ALE ไปทริกเพื่อให้ 74LS373 แลทช์แอดเดรส และพอร์ตเบอร์ศูนย์ ยังเป็นบัสแบบสองทิศทางในระหว่างช่วงคาบไซเคิลอ่านและเขียน

แรม และรอม กำหนดการเข้าแอดเดรสในเงื่อนไขที่ต่างกัน สัญญาณ PSEN สำหรับ รอม และสัญญาณ WR หรือสัญญาณ RD สำหรับแรม ซึ่งแต่ละส่วนมีขนาดการเข้าถึง 64 กิโลไบต์ รูปที่ 4-2 คือบอร์ดวงจร ไมโครโปรเซสเซอร์ 8031 และการถอดรหัสแอดเดรสหน่วยความจำ



รูปที่ 4-3 *วงจรถอดรหัสแอดเดรสอุปกรณ์ภายนอก*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) การถอดรหัสแอดเดรสอุปกรณ์ภายนอก

วงจรของบอร์ดควบคุมต้องติดต่อกับอุปกรณ์ภายนอกที่จำเป็นในการควบคุม ซึ่งแบ่งเป็นอุปกรณ์ คีย์บอร์ด ADC DAC เฟสดีเทกเตอร์ และอุปกรณ์แสดงผลเป็นต้น รูปที่ 4-3 แสดงถึงวงจรการถอดรหัสแอดเดรสอุปกรณ์ภายนอกตามแอดเดรสค่าต่างๆ

จากวงจรสังเกตได้ว่าการจัดสรรค์แอดเดรสเพื่อติดต่อกับอุปกรณ์ภายนอกนั้น จำเป็นต้องการถอดรหัสแอดเดรสที่มีลักษณะซ้ำกับการถอดรหัสแอดเดรสแรมภายนอก โดยใช้สัญญาณ RD กับ WR เป็นสัญญาณอีนาเบิลให้กับ 74LS138 ซึ่งเป็นไอซีตัวถอดรหัส 8 ช่อง ในการถอดรหัสแอดเดรส RAM ภายนอกนั้นสามารถจัดสรรค์เนื้อที่ได้ 64 กิโลไบต์ดังกล่าว แต่อย่างไรก็ดีการถอดรหัสแอดเดรสอุปกรณ์ภายนอกนั้นจะเนื้อที่ซ้ำเฉพาะ 32 กิโลไบต์หลังเท่านั้น ทำให้เนื้อที่ของแรม ภายนอกเหลือ 32 กิโลไบต์ ซึ่งเพียงพอกับการใช้งานในหลายๆการประยุกต์ แอดเดรสการถอดรหัสอุปกรณ์ภายนอกแสดงตามตาราง 4-1 ดังนี้

แอดเดรส	อุปกรณ์ติดต่อ
8000h ถึง 8FFFh	สำรอง
9000h ถึง 9FFFh	วงจรสแกนคีย์บอร์ดแบบเมตริก 12 คีย์(KBC)
A000h ถึง AFFFh	วงจรแสดงผลส่วนที่ 1 (DISP1)
B000h ถึง BFFFh	วงจรแสดงผลส่วนที่ 1 (DISP1)
C000h ถึง CFFFh	วงจรแปลงอะนาลอกเป็นดิจิตอล(ADC)
D000h ถึง DFFFh	วงจรแปลงดิจิตอลเป็นแอนะล็อก(DAC) และวงจรเฟสดีเทกเตอร์
E000h ถึง EFFFh	สำรอง
F000h ถึง FFFFh	สำรอง

ตารางที่ 4-1 *จำแนกการถอดรหัสแอดเดรสอุปกรณ์ภายนอก*

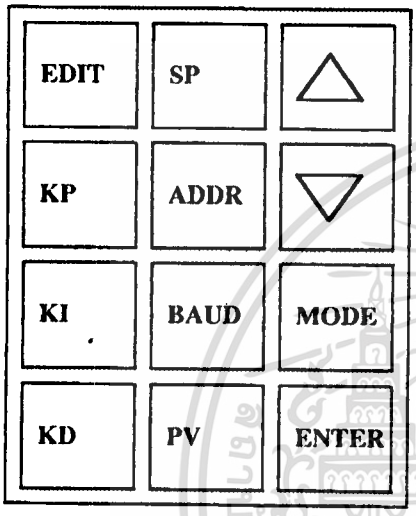
จากตารางสังเกตได้เช่น ช่วงแอดเดรสระว่าง 9000h ถึง 9FFFh นั้นสำหรับวงจรคีย์บอร์ด ซึ่งมีเนื้อที่ระหว่างแอดเดรสถึง 4 กิโลไบต์ อันที่จริงตามอุดมคติสามารถใช้เพียงไม่กี่ไบต์เท่านั้นสำหรับการถอดรหัสแอดเดรสพอร์ต แต่เพื่อความจำเป็นและความประหยัดในอุปกรณ์ถอดรหัสแอดเดรสจึงเห็นว่าการใช้ลักษณะนี้ เพียงพอสำหรับงานนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 วงจรต่างๆในตัวดิจิทัลคอนโทรลเลอร์

4.2.1 วงจรสแกนคีย์บอร์ด

อุปกรณ์ที่ติดต่อกับมนุษย์กับคอมพิวเตอร์ คือ คีย์บอร์ด ซึ่งจะประกอบด้วยแป้นคีย์บอร์ด ประกอบด้วยคีย์ต่างๆ เช่นคีย์ตัวเลข ตัวอักษร เป็นต้น ซึ่งในการประยุกต์ใช้กับตัวคอนโทรลเลอร์จะประกอบด้วยคีย์ซึ่งทำหน้าที่ต่างๆดังนี้



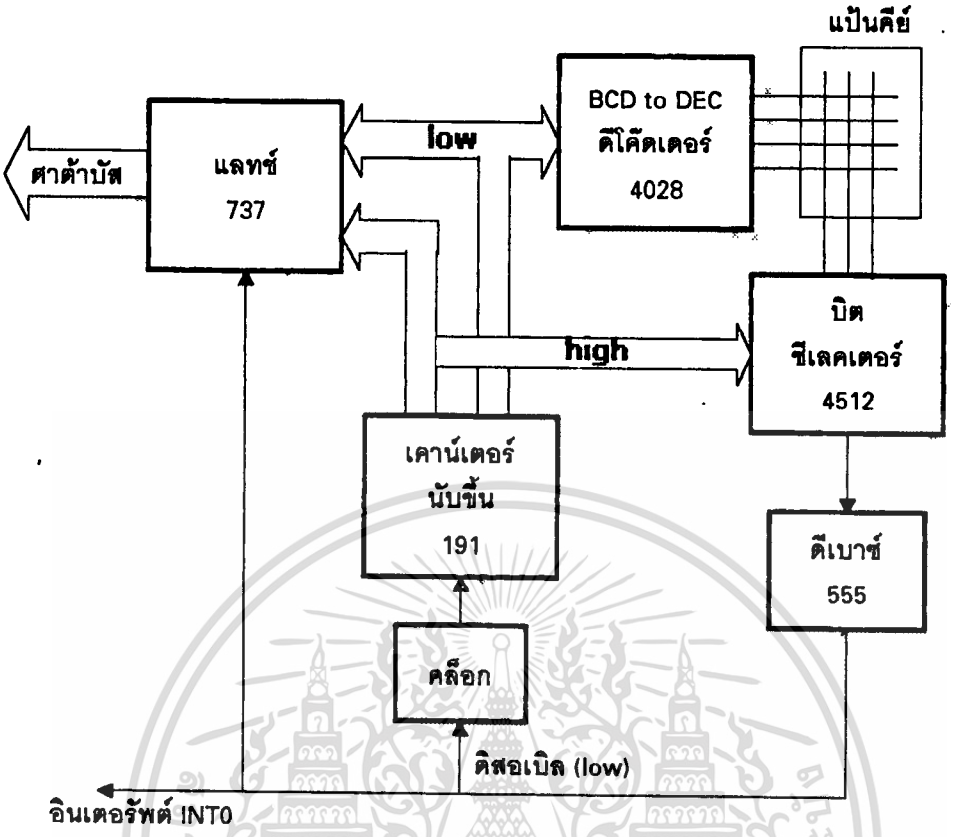
แป้นคีย์บอร์ด

EDIT : แก้ไขค่าต่างที่กำหนด
SP : แสดงค่าเซตพอย์นบนจอแสดงผล
KP : พรอปพรอชันนอลเกน
KI : อินทีกรัลเกน
KD : ดิริเวทีฟเกน
PV : แสดงค่าตัวแปร(process variable)
ADDR : แสดงหมายเลขแอดเดรสสถานี
BAUD : แสดงอัตรบอด
MODE : แสดงโหมดการทำงาน พร้อมแก้ไข
ENTER : รับข้อมูล หรือเตรียมพร้อมแก้ไข
△ : เพิ่มค่า
▽ : ลดค่า

หมายเหตุ รายละเอียดเกี่ยวกับการใช้งานดูวิธีการใช้งานในหัวข้อต่อไป

วงจรและโปรแกรมของคีย์บอร์ดต้องระแวดระวังเงื่อนไขที่เป็นไปได้เช่น การกดคีย์มากกว่าหนึ่งครั้งกดคีย์ค้างไว้ กดและปล่อยอย่างรวดเร็วเป็นต้น ซึ่งคุณสมบัติโดยทั่วไปของคีย์บอร์ดที่ถือปฏิบัติคือความสามารถในการป้องกันเปิดปิดของคีย์สวิทท์ซ้ำ หรือเรียกอีกอย่างหนึ่งว่า "ดีเบาซ์ (debounce)" การดีเบาซ์มีหลายแบบ เช่นใช้การหน่วง หรือใช้ R-S ฟลิปฟลอป การจัดองค์ประกอบของคีย์แบ่งเป็นแต่ละประเภทดังนี้ การจัดคีย์แบบ 1 คีย์ต่อ 1 บิตพอร์ต แบบนี้จะมีขาด้านหนึ่งของคีย์จะร่วมกันตลอด และอีกขาหนึ่งจะต่อกับพอร์ต ซึ่งการต่อแบบนี้จะมีการตรวจสอบคีย์เร็ว แต่ถ้ามีจำนวนคีย์มากๆจะสิ้นเปลืองจำนวนขาพอร์ตมาก การจัดคีย์อีกแบบหนึ่งคือการจัดแบบ XY ซึ่งจำนวนคีย์ทั้งหมดเท่ากับจำนวนแถว X และคูณคอลัมน์ Y โดยที่จำนวนขาพอร์ตเท่ากับ X บวก Y การจัดแบบนี้จะใช้นิยมใช้กันมาก ถ้ามีจำนวนคีย์มากกว่า 8 คีย์ขึ้นไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



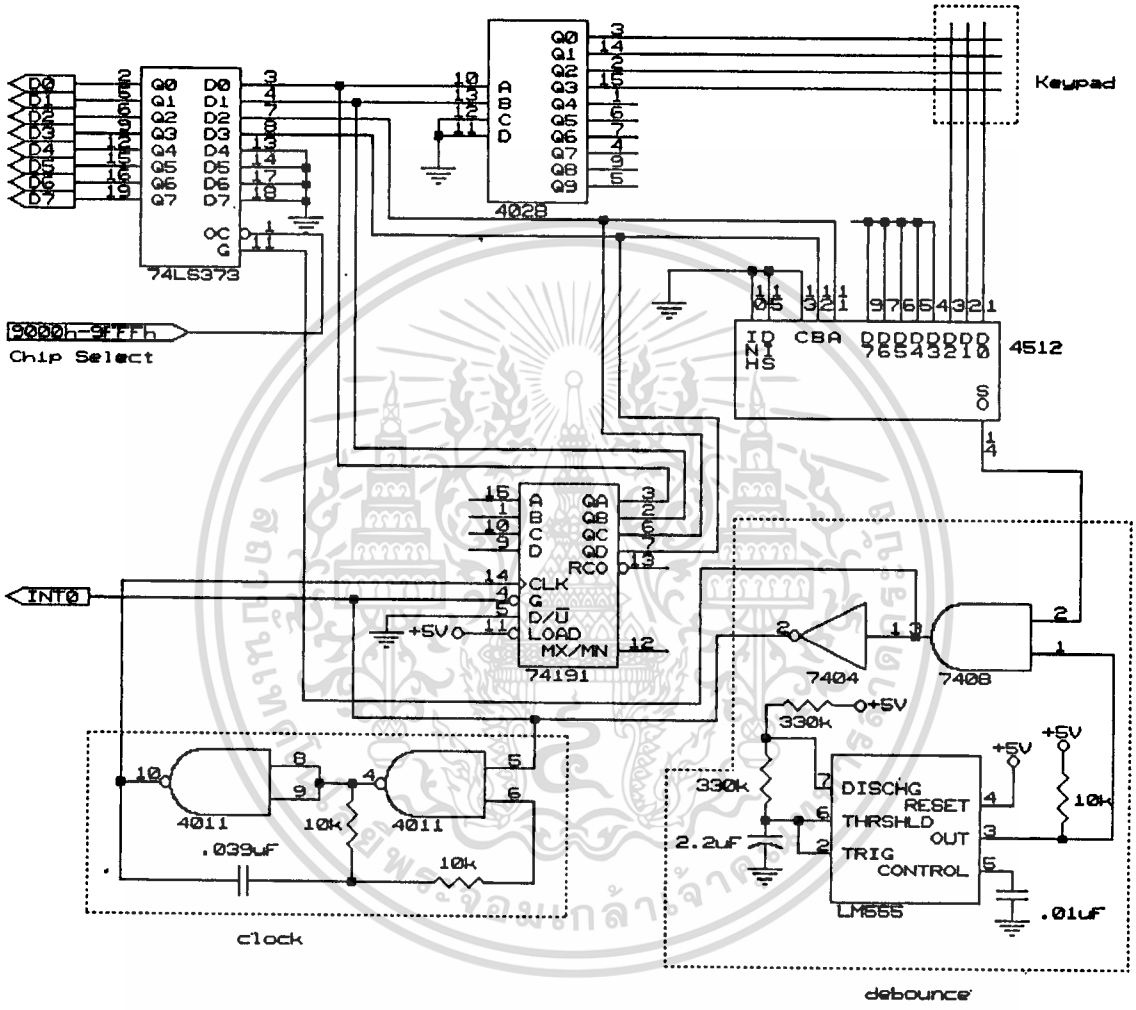
รูปที่ 4-4 *บล็อกไดอะแกรมสแกนคีย์บอร์ด*

สำหรับการสแกนคีย์สามารถใช้เป็นซอฟต์แวร์หรือเป็นวงจร ซึ่งในงานวิจัยนี้จะใช้เป็นวงจรในการสแกนคีย์ให้กับตัวคอนโทรลเลอร์ เหตุนี้เพราะว่าไม่ต้องการแบ่งเวลาของ ซีพียู มาใช้งานสแกนคีย์นั่นเอง วงจรสแกนคีย์บอร์ดนั้นมีบล็อกไดอะแกรมดังรูปที่ 4-4

ในขณะที่คีย์ยังไม่ถูกกด วงจรคล็อกจะไม่ถูกดิสอเบิล(disable) เพราะฉะนั้นวงจรคล็อกจะส่งสัญญาณคล็อกตามปกติให้กับตัวนับแบบนับขึ้น(up counter) ตัวนับจะทำการนับคล็อกเหล่านั้น ซึ่งเอาต์พุตที่ได้เป็นค่าไบนารี 4 บิตตั้งแต่ 00 ถึง 0Fh โดยที่เอาต์พุตบิตนัยสำคัญต่ำ(low-significance bit)ของตัวนับป้อนให้กับตัวถอดรหัส BCD-to-DECIMAL และเอาต์พุตบิตนัยสำคัญสูง(high-significance bit)สู่ตัวเลือกบิต(bit selector) ในการสแกนคีย์นั้นตัวถอดรหัสเมื่อรับอินพุตจากตัวนับแล้ว เอาต์พุตแต่ละขาจะสลับกันเป็นลอจิกสูงเรียงกันไปตลอดทั้ง 4 ขา เช่นเดียวกับตัวเลือกบิตจะคอยกวาดรับค่าอินพุตทั้ง 3 ขาจากคีย์บอร์ดซึ่งโดยปกติจะมีอินพุตรับคีย์เป็นลอจิกต่ำ ถ้าคีย์ไม่ถูกกดแล้วเอาต์พุตตัวเลือกจะเป็นลอจิกต่ำ สมมุติถ้ามีการกดคีย์จะมีลอจิกสูงเข้าทางอินพุตรับคีย์ของตัวเลือกบิต จะทำให้เอาต์พุตเป็นลอจิกสูง ซึ่งจะถูกดีเบาซ์จากวงจรดีเบาซ์ หลังจากนั้นวงจรสร้างสัญญาณนาฬิกาจะถูกดิสอเบิลคือหยุดทำงาน ในขณะเดียวกันตัวแลทช์ทำการเก็บไบต์ข้อมูลจากตัวนับ ซึ่งหยุดนับเมื่อไม่มีคล็อกพร้อมกับสัญญาณที่ส่งไปอินเตอร์รัพต์บนบอร์ดไมโครคอนโทรล-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลอร์ 8031 เพื่อเป็นการบอกไมโครโปรเซสเซอร์ว่ามีการกดคีย์ และข้อมูลถูกเก็บไว้ที่ตัวแลทช์ตามแอดเดรสที่กำหนด ซึ่งเมื่อไมโครโปรเซสเซอร์ถูกอินเตอร์รัพต์ก็ จะทำการ อ่าน ข้อมูลจากแอดเดรสของคีย์บอร์ดคือ 9000h-9FFFh แต่ละคีย์จะกำหนดเป็นข้อมูลไบต์ ตามตารางที่ 4-2



รูปที่ 4-5 วงจรสแกนคีย์บอร์ด

วงจรสแกนคีย์อย่างสมบูรณ์ แสดงดังรูป 4-5 สังเกตว่าเส้นประจะคล้องจองกับบล็อกไดอะแกรม ซึ่งได้แสดงดังรูป 4-4 เช่น บล็อกดีเบาซ์(debounce)ประกอบด้วย ไอซี 555 ,แอนด์เกทและน็อตเกท เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปุ่มคีย์บอร์ดถูกกด	ไบต์ข้อมูลถูกส่ง
เพิ่มค่า	00h
ลดค่า	01h
MODE	02h
ENTER	03h
SP	04h
ADDR	05h
BAUD	06h
PV	07h
EDIT	08h
KP	09h
KI	0Ah
KD	0Bh

ตารางที่ 4-2 แสดงไบต์ข้อมูลที่ถูกส่งจากวงจรสแกนคีย์เมื่อกดปุ่ม

4.2.2 วงจรแสดงผล(Display Circuit)

ถ้าคีย์บอร์ดเป็นอินพุตอินเตอร์เฟสกับมนุษย์แล้ว การแสดงผล คือเอาต์พุตอินเตอร์เฟสกับมนุษย์ บอร์ดแสดงผล แบ่งเป็น 3 อย่างดังนี้

1. แสดงเป็นจุดแสง(single light)
2. แสดงเป็นตัวอักษร(single charactor)
3. แสดงเป็นอักขระแบบฉลาด(intelligent alphanumeric)

จุดแสงคือมีความหมายเดียวกับการใช้ LED โดยใช้แสดงผลเป็นบิต หรือจะแสดงค่าเปิด-ปิดเป็นต้น การแสดงเป็นตัวอักษรแสดงเป็นตัวเลขและตัวอักขระในขอบเขตจำกัด ซึ่งอาจเป็นอย่างง่ายคือ ตัวเลข 7 ส่วน(7-segment) หรือที่ดีกว่าคือ ดอตแมตริกซ์(dot-matrix) ซึ่งการแสดงผลจะรับโค๊ดแอสกี้(ASCII) 8 บิต โดยจะคล้องจองกับแบบแผน(patterm)การแสดงผลบน ดอตแมตริกซ์ ส่วนการแสดงเป็นอักขระแบบฉลาดนั้นจะถูกจัดเตรียมไมโครคอนโทรลเลอร์อยู่ภายใน อย่างเช่นการ์ด

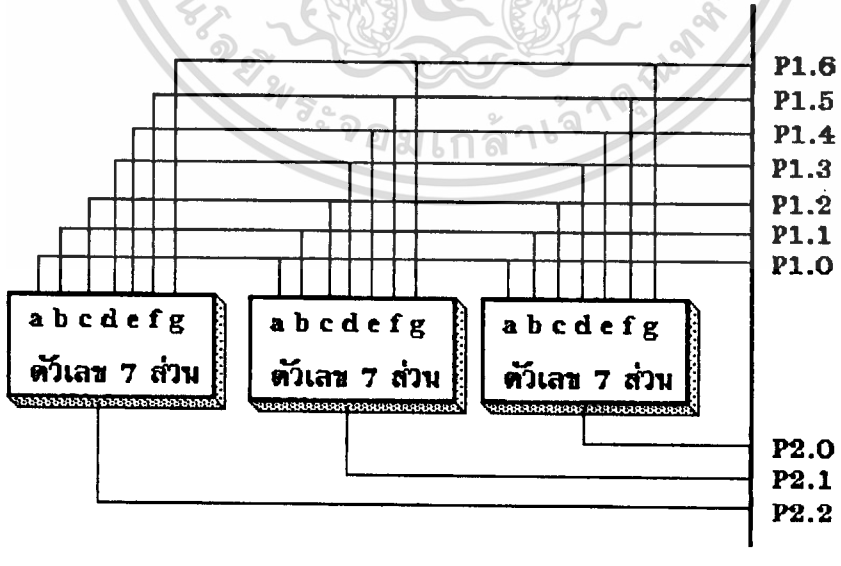
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงผล วีจีเอ(VGA) ดังที่ใช้กับคอมพิวเตอร์พีซี เป็นต้น ซึ่งในตัวคอนโทรลเลอร์ที่ใช้อยู่ในงานวิจัยจะใช้แสดงเป็นอักษระหรือตัวเลข 7 ส่วน(7-segment)

ตัวเลข 7 ส่วน(7-segment)เป็นเซกเมนต์ของ LED ประกอบกับคล้ายเลข "8." กับขาร่วม เช่นแคโทดร่วมหรือแอโนดร่วม โดยปกติจะใช้เป็นแคโทดร่วม ถ้ามากกว่าใช้ 1 หลัก แล้วต้องทำมัลติเพลกซ์เวลาเพื่อไม่ให้ตาของมนุษย์จับการกระพริบได้ ซึ่งการแสดงผลต้องต่อเนื่องไม่มากกว่า 10 มิลลิวินาที[3] ถ้าต้องการจะรู้ว่าตัวเลข 7 ส่วนแต่ละตัว ใช้ช่วงเวลาเปลี่ยนข้อมูลใหม่เท่าไรแล้วต้องนำจำนวนตัวเลข 7 ส่วนทั้งหมดเป็นตัวหารของ 10 มิลลิวินาที

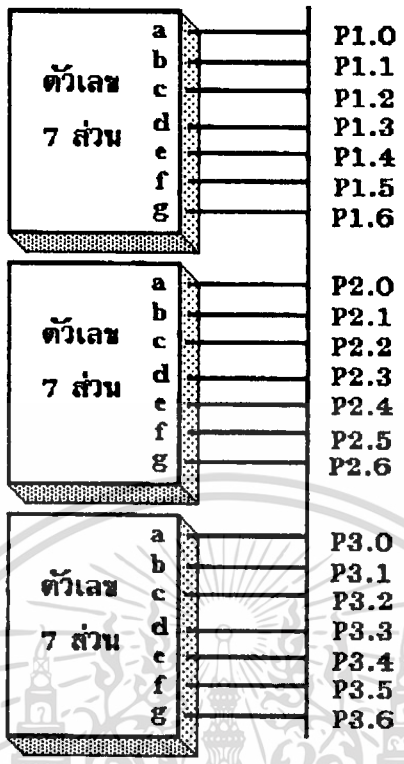
สำหรับวงจรขับตัวเลข 7 ส่วนแบบนิยมใช้ทั่วไปนั้น จะใช้วิธีให้แต่ละเซกเมนต์ (a,b,c,d,e,f,g) ของแต่ละตัวของตัวเลข 7 ส่วน ต่อเข้ากับ 1 บิตพอร์ต รวมทั้งหมดเป็น 1 ไบต์พอร์ต และ ขาร่วม (common) ของแต่ละตัวต่อเข้ากับ 1 บิตพอร์ต แสดงรูปที่ 4-6

ในเรื่องของโปรแกรมสำหรับตัวเลข 7 ส่วน(7-segment) อินเตอร์รัพต์ไทเมอร์ T0 จะคอยกำหนดช่วงเวลาโหลดตัวอักษรไปให้ตัวเลข 7 ส่วนแต่ละหลัก โดยโปรแกรมอินเตอร์รัพต์จะไปหาไบต์อักขระตามตำแหน่งของหลักในตำแหน่งของหน่วยความจำ เมื่อได้ข้อมูลไบต์จะส่งเป็นเอาต์พุตที่ P1 พอร์ต แล้วให้บิตลอจิกต่ำที่ P2 พอร์ต ที่บิตของหลักตัวเลข 7 ส่วน ก็สิ้นสุดการทำอินเตอร์รัพต์รัพต์รูทีนครั้งหนึ่ง เมื่ออินเตอร์รัพต์ไทเมอร์เกิดขึ้นต่อมาก็จะเขยิบไปทำตัวเลข 7 ส่วนในหลักต่อไป โดยที่ช่วงเวลาการเกิดอินเตอร์รัพต์ไทเมอร์เป็นช่วงเวลา ที่ไม่ทำให้ตาของมนุษย์จับการกระพริบได้ดังกล่าว



รูปที่ 4-6 แสดงเป็นวงจรขับตัวเลข 7 ส่วนอย่างง่ายแบบทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-7 การขับตัวเลข 7 ส่วนโดยวิธีตรง

อย่างไรก็ดีการกระทำแบบการแสดงผลข้างต้นไม่เหมาะสม ในสำหรับงานวิจัยเพื่อนำไปออกแบบตัวคอนโทลเลอร์นี้ เพราะว่าอินเตอร์รัพต์ T0 ถูกสำรองไว้ใช้งาน และซีพียูต้องรับภาระ จึงกระทำโดยใช้วิธีขับตัวเลข 7 ส่วนโดยตรง คือขับ 1 หลักต่อ 1 ไบต์พอร์ต ซึ่งใช้วิธีนี้การแสดงผลจะไม่กระพริบอย่างสิ้นเชิง แต่อย่างไรก็ดีถ้าเป็นจำนวนหลักของตัวเลข 7 ส่วนมากวิธีนี้จะไม่เหมาะสมเช่นกัน แสดงดังรูปที่ 4-7 ดังต่อไปนี้

4.2.3 วงจร ADC และ DAC

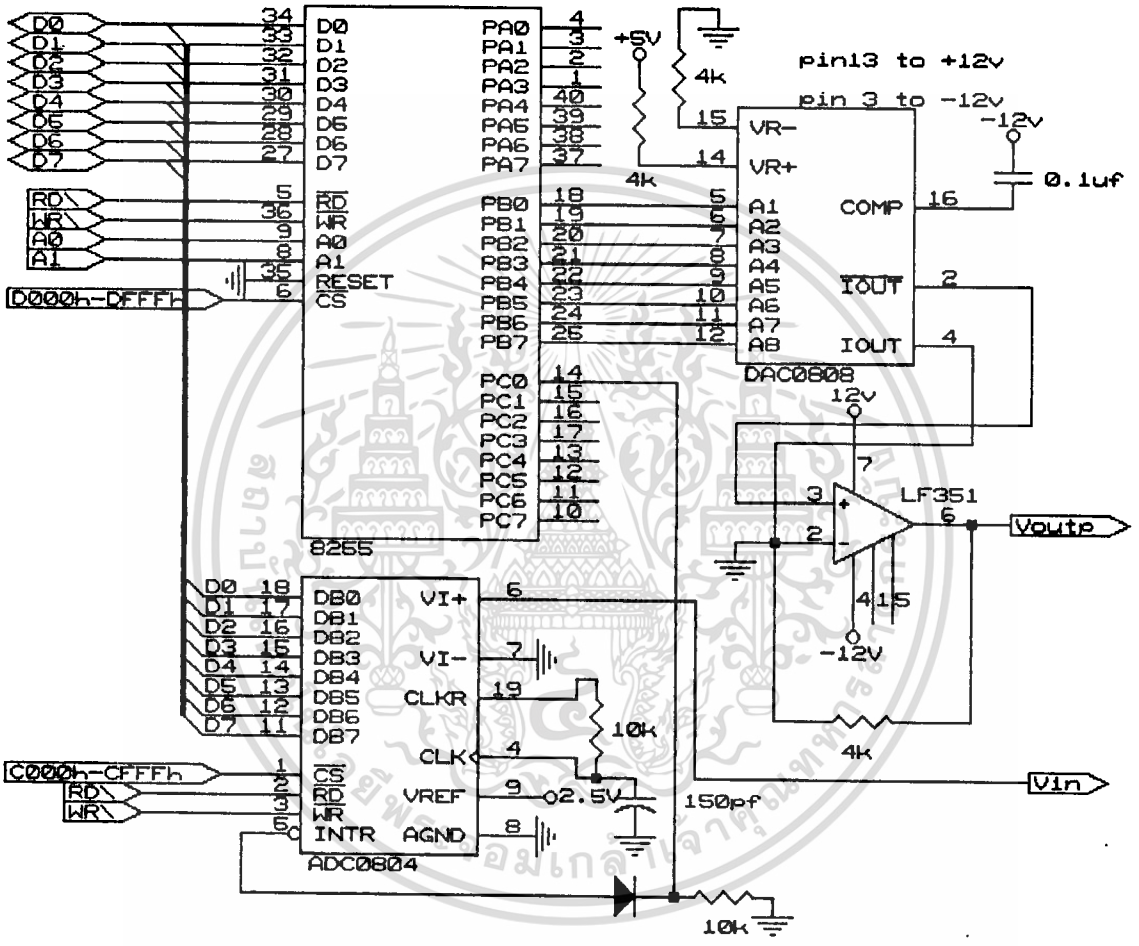
เอาต์พุตตัวทรานสดิวเซอร์โดยมากจะเป็นแอนะล็อกโวลต์เตจ ซึ่งถ้าหากข้อมูลจากทรานสดิวเซอร์แบบนี้ถูกนำเข้าไปเก็บในระบบดิจิทัลแล้ว มันต้องถูกเปลี่ยนเข้าสู่สมมูลทางดิจิทัล คือการแปลงแอนะล็อกเป็นดิจิทัลนั่นเอง ตัว ADC จะประกอบด้วยคอมพาราเตอร์และ I/O โดยทำให้ได้สมบูรณ์จะต้องได้รับการถูกสั่งการจากโปรแกรมซอฟต์แวร์อีกที อย่างไรก็ดีวิธีนี้ไม่ได้เป็นการประหยัดเวลาของซีพียู ซึ่งมันต้องแบ่งเวลาในการขบวนการเปลี่ยนแปลงแอนะล็อกเป็นดิจิทัล

ในทางกลับกันการเปลี่ยนแอนะล็อกเป็นดิจิทัลของ ADC นั้นสามารถกระทำให้บรรลุผลอย่างรวดเร็วได้โดยการใช้ ADC ในลักษณะของแฮนด์เชค(handshake) โดยยอมให้ ADC เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อิสระในขบวนการแปลงซึ่งจะไม่ขึ้นอยู่กับการทำงานของซีพียู เมื่อ ADC พร้อมที่จะส่งข้อมูลสู่ซีพียูก็จะมีสัญญาณ ready เพื่อบอกซีพียูเพื่อให้ซีพียูเตรียมพร้อม เพื่อที่ซีพียูจะส่งสัญญาณแฮนด์เชคไปที่ ADC ว่าพร้อมรับข้อมูลดิจิทัลซึ่งจะได้รับข้อมูลจะ ADC ในทันที

วงจร ADC และ DAC แสดงดังรูป 4-8



รูปที่ 4-8 *แสดงวงจร ADC และ DAC*

มีข้อมูล A/D คอนเวอร์เตอร์ตามวงจรดังนี้

- โวลต์เตจแอนะล็อกอินพุตศูนย์ โวลต์ถึง 5 โวลต์ ใช้กับแหล่งจ่ายไฟ 5 โวลต์ทางเดียว
- ไม่มีการปรับศูนย์ (zero adjust)
- ความละเอียด 8 บิต
- เออเรอร์ทั้งสิ้น + LSB ของ ADC0804
- เวลาการเปลี่ยน 100 uS ของ ADC0804

เอกสารนี้เ[illegible]การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับ DAC ใช้ชิปสำเร็จรูปเบอร์ DAC0808 ซึ่งเป็นแบบ R-2R ซึ่งนิยมใช้โดยทั่วไป ถูกต่ออยู่กับพอร์ต PB0..7 ของ 8255 โดยที่ 8255 ถูกถอดรหัสแอดเดรสหมายเลข C000h-CFFFh ที่ต้องต่ออยู่กับพอร์ตเพราะว่า DAC 0808 ไม่มีตัวแลทช์อยู่ภายในจึงจำเป็นต้องใช้ I/O พอร์ต 8255 เป็นตัวแลทช์ โดยมีข้อมูลทางวิชาการดังนี้

- โวลต์อ้างอิง +Vref เป็น + 5 โวลต์และ -Vref เป็น 0 โวลต์
- ใช้แหล่งจ่ายไฟคู่ + 12V
- ความละเอียด 8 บิต
- เวลาการแปลงข้อมูล 10 uS

4.2.4 เฟสดีเทกเตอร์(PFD)และการเลียนแบบโดยซอฟต์แวร์

ในบทที่ 2 ได้กล่าวถึงทฤษฎีต่างของระบบเฟสล็อกลูปไว้ต่างไว้มากพอ ดังที่เราเข้าใจมาแต่เดิมนั้นระบบเฟสล็อกลูปเป็นวงจรอิเลคโทรนิคใช้ในด้านต่างๆ เช่น ด้านการสื่อสาร คือ โทรทัศน์ วิทยุ เป็นต้น อย่างไรก็ดีเหล่านั้นเป็นวงจรทางด้านแอนะล็อก ต่อมาเป็นการพัฒนามาเป็นระบบดิจิ-ดิจิทัลเฟสล็อกลูป เช่น วงจรสังเคราะห์ความถี่

1) ทฤษฎีซอฟต์แวร์เลียนแบบเฟสดีเทกเตอร์

แต่การทำซอฟต์แวร์เลียนแบบเฟสดีเทกเตอร์นั้นไม่นิยมใช้กัน ซึ่งปัจจุบันมีเรื่องราวเกี่ยวกับการใช้ซอฟต์แวร์เลียนแบบมากขึ้นซึ่งใช้ไมโครโปรเซสเซอร์ หรือชิป DSP เป็นต้นอย่างเช่นในบทที่ 2.4.2 เป็นตัวเฟสดีเทกเตอร์แบบ NRPD (Nyquist-rate)โดยจะใช้ความถี่แซมปลิงมากกว่า 2 เท่า จึงสามารถสร้างสัญญาณขึ้นมาใหม่ได้ ดูรูปที่ 2-10 สัญญาณอ้างอิงและสัญญาณป้อนกลับจะถูกนำมาคูณโดยใช้รูทีนคูณ โดยที่เฟสดีเทกเตอร์แบบ NRPD จะทำงานคล้ายเฟสล็อกลูปแบบเชิงเส้น อย่างไรก็ดีขบวนการแซมปลิงสัญญาณและขบวนการคำนวนจะใช้เป็นดิจิทัลและซอฟต์แวร์ ถึงอย่างไรก็ยังเป็นระบบเฟสล็อกแบบเชิงเส้นอยู่ดี มีอ้างถึงการคำนวณดังนี้

$$u_1(t) = U_{10}\sin(\omega_1 t + \theta_1)$$

$$u_2(t) = U_{20}\cos(\omega_1 t + \theta_2)$$

เอาต์พุตเฟสดีเทกเตอร์ $u_d(t)$ ของผลคูณกันของสัญญาณทั้งสอง แสดงได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$u_d(t) = ku_1u_2 = \frac{kU_{10}U_{20}}{2}\left[\sin(\theta_1 - \theta_2) + \sin(2\omega_1 + \theta_1 + \theta_2)\right]$$

สมการข้างต้นเผยให้เห็นว่า $u_d(t)$ เป็นการวางทับกันระหว่างคอมโพเนนต์ DC และคอมโพเนนต์ AC ในส่วนคอมโพเนนต์ AC จะถูกฟิลเตอร์โดยลูปฟิลเตอร์ เพราะฉะนั้นตั้งแต่นี้พิจารณาค่าเฉลี่ยคอมโพเนนต์ u_d เป็น

$$u_d = K_d \sin\theta_e$$

ที่ซึ่ง

$$K_d = \frac{KU_{10}U_{20}}{2}$$

เป็นเฟสดีเทกเตอร์เกน และ

$$\theta_e = \theta_1 - \theta_2$$

เป็นเฟสเออเรอร์

มีอีกวิธีหนึ่งซึ่งอ้างอิงจากบทที่ 2 คือวิธีการแปลงฮิลเบอร์ต(Hilbert transform) แสดงดังรูป 2-13 เป็นความหลายหลากทางคณิตศาสตร์ ที่นำมาแก้ปัญหาทางซอฟต์แวร์ของตัวเฟสดีเทกเตอร์ตามวิธีคูณของสัญญาณอ้างอิงกับสัญญาณป้อนกลับข้างต้น วิธีการแปลงฮิลเบอร์ตนี้จะใช้สัญญาณป้อนกลับจาก VCO โดยใช้เป็น DCO แทน โดยตัว DCO จะสร้างสัญญาณป้อนกลับซึ่งมีเฟสเลื่อนไป 90 องศาอีกหนึ่งสัญญาณ ดังนั้นจึงมีสัญญาณป้อนกลับสองสัญญาณแล้วเข้าสู่ขบวนการซอฟต์แวร์เฟสดีเทกเตอร์ ซึ่งจะได้เอาต์พุตเป็นลักษณะเหมือนกับ u_d ข้างต้น อย่างไรก็ตามก็ยังไม่ใช่ดิจิทัลเฟสล็อกลูปแต่ยังเป็นระบบแบบเชิงเส้นอยู่ ถึงแม้จะเป็นขบวนการทางดิจิทัลก็ตาม

4.2.5 วงจรใช้ในโหมดเฟสล็อกลูป

ตามที่บทความดังกล่าวยังต้องมีเงื่อนไขของมอเตอร์เข้ามาเกี่ยวข้อง เพื่อหลีกเลี่ยงการส่ายของความเร็วของมอเตอร์ เช่นต้องเป็นมอเตอร์ที่มีความเฉื่อยโรเตอร์มากเป็นต้น อย่างไรก็ตามลักษณะการเลียนแบบซอฟต์แวร์ของ PD ตามบทความนั้นยังไม่เป็นไปตามทฤษฎีอย่างแท้จริง ดิจิทัลเฟสดีเทกเตอร์ที่ดีที่สุดในปัจจุบันเป็นเฟสดีเทกเตอร์แบบ 4 (อ้างจากบทที่ 2) ซึ่งเรียกได้ว่าเป็นเฟส/ความถี่ดีเทกเตอร์ หรือเฟสดีเทกเตอร์แบบไตร-สเตต(tri-state) ซึ่งทำให้ระบบเฟสล็อกลูปมีความแน่นนอนสูง ซึ่งงานวิจัยชิ้นนี้จะใช้เฟสดีเทกเตอร์แบบ 4 นี้เป็นหลักในการใช้ซอฟต์แวร์เลียนแบบดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) **เฟสดีเทกเตอร์แบบไตร-สเตต**

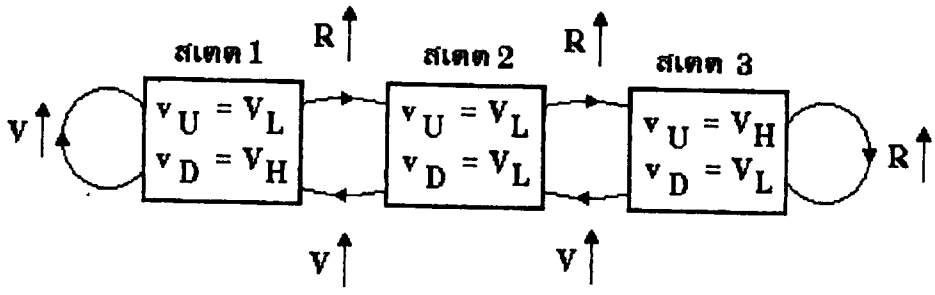
แนวความคิดของ PD แบบ n-state อาจจะขยายกว้างออกไปแล้วแต่ต้องการ แต่ PD แบบ tri-state ถูกใช้อย่างกว้างขวางเพราะว่าใช้ง่าย มีความเป็นเชิงเส้นในช่วง $+2\pi$ เรเดียล และสามารถเป็นทั้งเฟสดีเทกเตอร์และความถี่ดีเทกเตอร์ สเตตไดอะแกรม(state diagram)แสดงดังรูป 4-9 เพื่อเน้นการอธิบายได้เพิ่มเติมจากบทที่ 2 เกี่ยวกับ PD แบบไตร-สเตต หรือ PFD ดังนี้สเตตที่ 1 ซึ่ง $v_u = V_L$ และ $v_d = V_H$ สภาวะนี้จะเกิดได้ก็ต่อเมื่อขอบขาขึ้นของสัญญาณ V เกิดขึ้นติดต่อกันจนกระทั่งขอบขาขึ้นของสัญญาณ R เกิดขึ้นแล้วจะเริ่มเข้าสเตตที่ 2 ซึ่ง $v_u = V_L$ และ $v_d = V_L$ ในสเตตที่ 2 นี้สามารถเปลี่ยนเป็นสเตตที่ 1 หรือสเตตที่ 3 ได้แล้วแต่ขอบขาขึ้นสัญญาณ V หรือสัญญาณ R จะเกิดขึ้นก่อน ถ้าเป็นขอบขาขึ้นของ R จะเปลี่ยนจากสเตตที่ 2 เป็นสเตตที่ 3 ซึ่ง $v_u = V_H$ และ $v_d = V_L$ หรือถ้าเป็นขอบขาขึ้นของ V จะเปลี่ยนจากสเตตที่ 2 เป็นสเตตที่ 1 อย่างไรก็ดีจะไม่มีการเปลี่ยนข้ามสเตตจาก 1 ไปสเตต 3 หรือสเตต 3 เป็นสเตต 1

รูปที่ 4-9d เป็นการตอบสนองลักษณะเอาต์พุตของ PD สังเกตว่าจะเป็นเชิงเส้นจะอยู่ในช่วง 4π หลังจากนั้นจะเป็นบวก(positive) ซึ่งจะกระทำซ้ำไซเคิลทุก 2π และถ้า θ_e ลดลง ก็เป็นช่วงกราฟการตอบสนองลดลงเมื่อลดลงเกิน 4π จากนั้นเอาต์พุต PD เป็นลบ(negative) และจะเป็นซ้ำไซเคิลตลอดทุก 2π

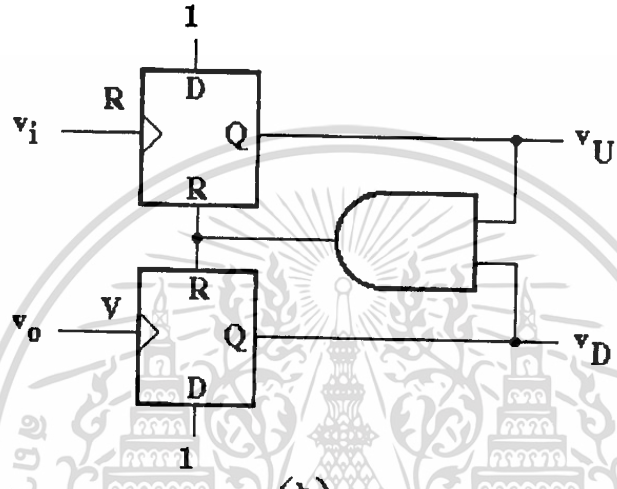
ในลักษณะการกระทำ PD แบบความถี่ดีเทกเตอร์ คือสำหรับกรณี $\omega_i > \omega_O$ นั้น θ_e จะเพิ่มมากขึ้นตลอดเวลา จะทำให้เอาต์พุต PD ยังเป็นบวก(positive) สำหรับกรณี $\omega_i < \omega_O$ นั้น θ_e จะน้อยลงตลอดเวลา ทำให้เอาต์พุต PD เป็นลบ(negative) นี้เป็นส่วนช่วยอย่างมากในการเข้าสู่การล็อกเมื่อสองความถี่ต่างกัน

ลักษณะเอาต์พุตเฉลี่ยของ PD เป็นลักษณะค่าซ้อน(multiple-value) แสดงดังรูปที่ 4-9d นี้ทำให้มันไม่เหมาะสมในการใช้ประยุกต์บางอย่าง ซึ่งพัลส์ v_i อาจจะหายไป ทำให้ค่า v_d กระโดดสู่ V_{dm} เพราะฉะนั้น PD แบบไตร-สเตตไม่สามารถใช้ในสถานะสัญญาณรบกวนมากๆได้ เช่นใช้สำหรับการนำคืนสัญญาณนาฬิกาหรือคล็อกรีคัฟวอรี่(clock recovery) จากข้อมูล

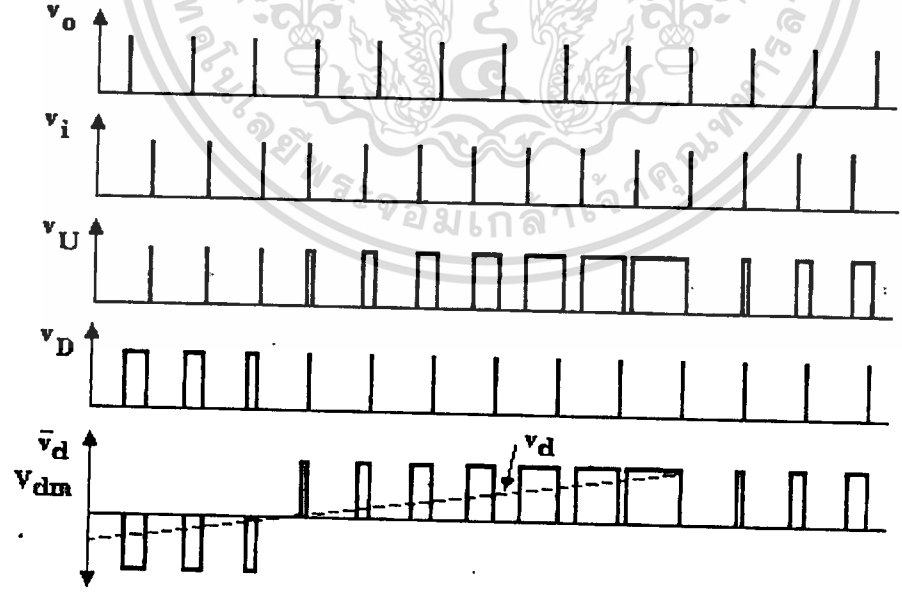
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(a)

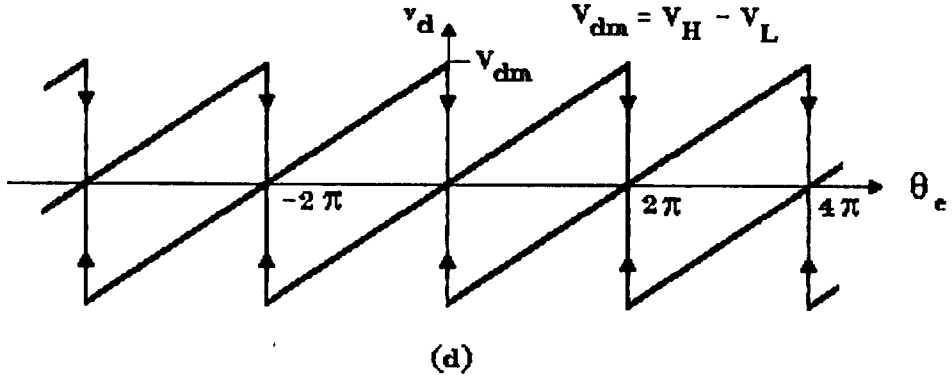


(b)



(c)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

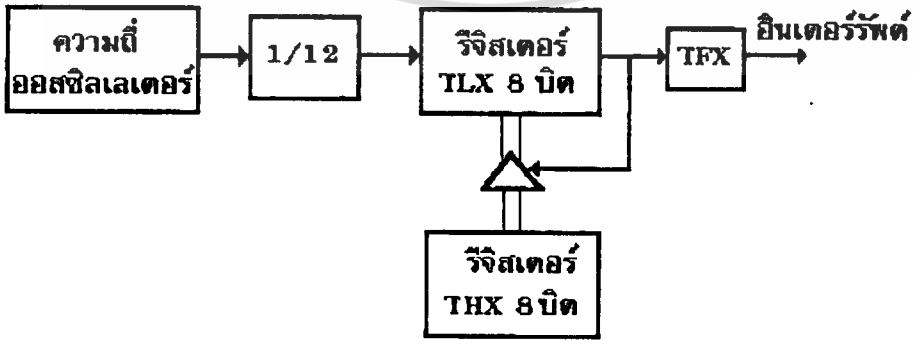


รูปที่ 4-9 *ไดอะแกรมการทำงานแบบเฟสดีเทกเตอร์ไตร-สเตต*

2) หลักการสร้างวงจร

ขอเกริ่นเพื่อเตือนความจำสักเล็กน้อย บทความของ Howard A. Raphael [6] ซึ่งอ้างอิงถึงรูปที่ 4-12 แสดงบล็อกไดอะแกรมของวงจร โดยที่การกำหนดเซตพอยน์นั้น จะกำหนดจากตัวนับแบบโปรแกรมได้(programable counter)เป็นค่า n ซึ่งมีความกว้างของช่วงเป็น 8 บิตค่า n สูงที่สุดจะเป็น 256 และต่ำสุดเป็น 0 ดังนั้นเซตพอยน์กำหนดตั้งแต่ 0 ถึง 256 เทียบเป็นเปอร์เซ็นต์เท่ากับ 0 เปอร์เซ็นต์ ถึง 100 เปอร์เซ็นต์

ในงานวิจัยไมโครโปรเซสเซอร์คือ 8031 ใช้ทำงานด้วยความถี่ของคล็อก 12 เมกะเฮิรตซ์ ซึ่งมีไทเมอร์ และตัวนับภายในสองช่อง แต่ไม่สามารถใช้ไทเมอร์ภายในได้เพราะว่า ค่านับถูกโหลดผ่านเรจิสเตอร์ THX ได้ 8 บิตสู่เรจิสเตอร์ TLX ทุกๆเวลาที่ TLX โอเวอร์โฟล์ว และถูกรีโหลดจาก TLX เมื่อไม่มีการเปลี่ยนแปลงบนเรจิสเตอร์ THX แสดงตามรูปที่ 4-10



รูปที่ 4-10 กระบวนการ *Auto-reload ของ TLX จาก THX*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปความถี่ออสซิลเลเตอร์เป็น 12 เมกะเฮิรตซ์ จะถูกหารด้วย 12 จะเหลือความถี่ 1 เมกะเฮิรตซ์ สมมุติถ้าค่าเซตพอย์นเท่ากับ 50 เปอร์เซ็นต์ แปลงเป็นเลขฐาน 16 เท่ากับ 80H หรือ 128D ถูกโหลดเข้าสู่ THX ดังนั้นคิดเป็นไทเมอร์ดีเลย์เท่ากับ $128/10^6$ เท่ากับ 0.128 มิลลิวินาที

ไมโครโปรเซสเซอร์ 8031 มีแมชีนไซเคิล(machine cycle)เท่ากับ 1 ไมโครวินาที ซึ่งในหนึ่งคำสั่งของไมโครโปรเซสเซอร์ต้องใช้เวลาเฉลี่ย 2 แมชีนไซเคิล ถ้าดีเลย์จากไทเมอร์เป็น 0.128 มิลลิวินาทีตามข้างต้น แล้วไมโครโปรเซสเซอร์จะสามารถทำได้คำสั่งได้เพียง 80 คำสั่ง หรือถ้าเซตพอย์นเป็น 0 เปอร์เซ็นต์ แล้วจะมีไทเมอร์ดีเลย์เพียง 1 ไมโครวินาที เท่านั้น ซึ่งไมโครโปรเซสเซอร์จะไม่สามารถทำคำสั่งได้แม้เพียง 1 คำสั่ง

จากนี้ทำให้ความคิดที่จะใช้ไทเมอร์ภายในนั้นไม่สำเร็จ จึงจะต้องออกแบบวงจรไทเมอร์แบบโปรแกรมได้ เพื่อใช้เป็นความถี่อ้างอิงให้กับระบบดิจิทัลเฟสล็อกลูป ดังรูปที่ 4-11a แสดงเป็นบล็อกไดอะแกรมของวงจร

บล็อกไดอะแกรม 4-11a มีหลักการทำงานดังต่อไปนี้ ข้อมูลเซตพอย์น 8 บิตจะถูกส่งผ่าน I/O พอร์ตถูกถอดรหัสแอดเดรสโดยไมโครโปรเซสเซอร์หมายเลขแอดเดรส D000H (PORT A) ถูกแลทช์เป็นข้อมูลอินพุตของตัวนับ ซึ่งตัวนับได้รับสัญญาณนาฬิกาคงที่ความถี่ประมาณ 300 ตลอดเวลา เมื่อตัวนับนับจนโอเวอร์โฟล์วจะสร้างริปเปิลแคร์รี่เอาต์พุต(ripple carry output) ถูกดีเลย์ไปสองไซเคิลจากบล็อกดีเลย์เพื่อเตรียมส่งให้โมโนสเตเบิลตัวที่ 1 และตัวที่ 2 เพื่อจัดไทมิ่ง(timing)ให้เหมาะสมกับตัวนับ เพื่อทำโหลดใหม่(reload)ให้กับตัวนับ และส่งสัญญาณอินเตอร์รัพต์เข้าสู่ไมโครโปรเซสเซอร์ 8031 ที่ขา INT1 เพื่อใช้เป็นความถี่อ้างอิงให้กับโปรแกรมรูทีนของเฟสล็อกลูปต่อไป ตัวอย่างเช่นเมื่อโหลดค่าเซตพอย์นค่าต่ำสุดเป็น 00H จะไมโครโปรเซสเซอร์จะถูกอินเตอร์รัพต์ทุกๆ 256 ไซเคิลของสัญญาณนาฬิกา หรือคิดเป็นความถี่อ้างอิงของระบบเป็น 1.17 กิโลเฮิรตซ์ และถ้าเซตพอย์นเป็น FFH จะถูกอินเตอร์รัพต์ทุกๆ 1 ไซเคิลนาฬิกา คิดเป็นความถี่อ้างอิงได้เท่ากับ 300 กิโลเฮิรตซ์ ซึ่งสามารถหาความสำพันธ์ระหว่างเซตพอย์น กับความถี่อ้างอิงได้ตามสมการดังนี้

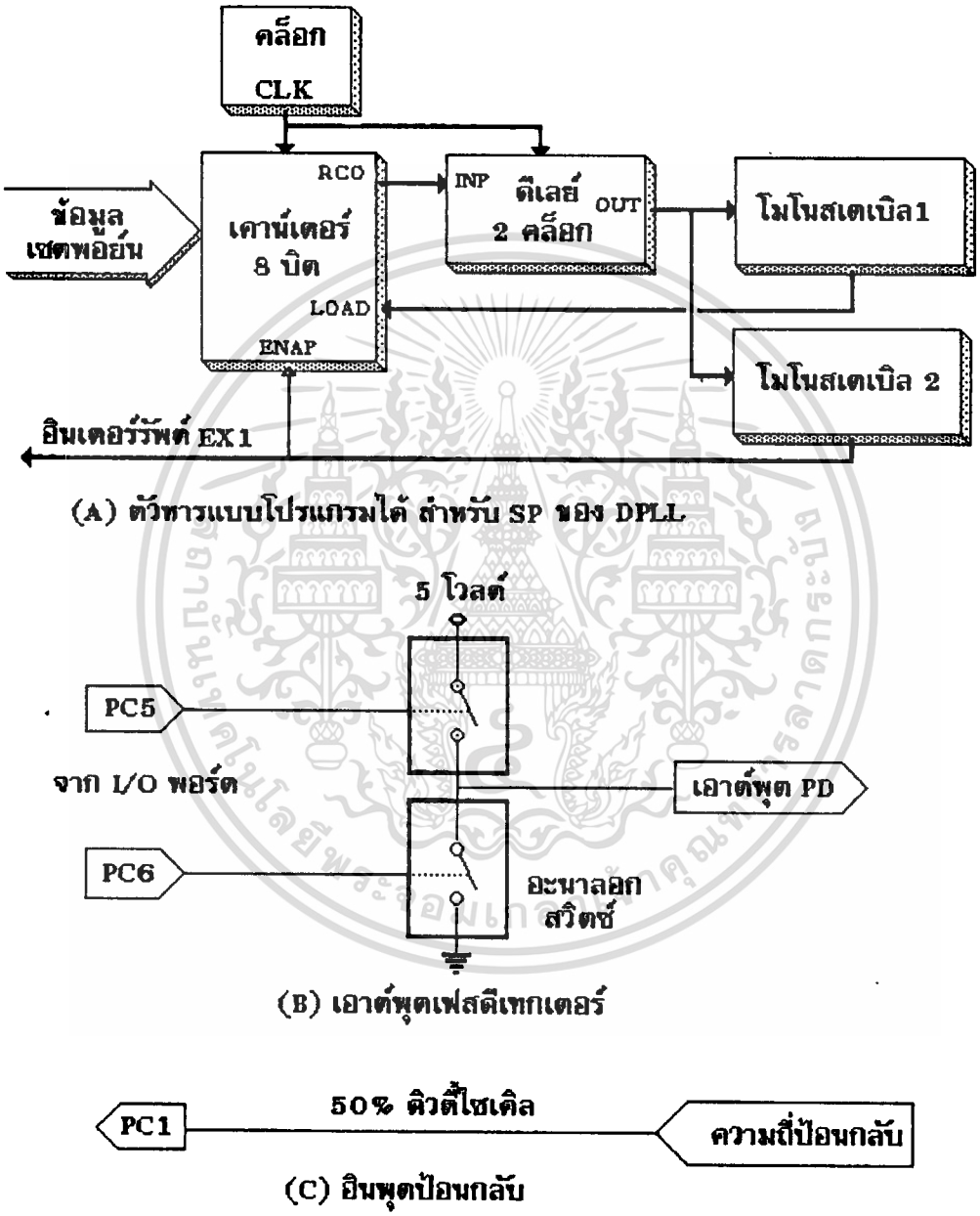
FRQ(Hz) = Clock(Hz)/Complement of Setpoint(HEX)

ดังนั้นถ้าใช้สัญญาณนาฬิกาเป็น 300 กิโลเฮิรตซ์ จะได้ช่วงกว้างความถี่อ้างอิง ตั้งแต่ 1.17 กิโลเฮิรตซ์ ถึง 300 กิโลเฮิรตซ์ หรือช่วงเซตพอย์นตั้งแต่ 0 เปอร์เซ็นต์ ถึง 100 เปอร์เซ็นต์เป็นต้น

รูปที่ 4-11b แสดงเอาต์พุตของเฟสดีเทกเตอร์(เลียนแบบโดยซอฟต์แวร์ ซึ่งทำให้เป็นลักษณะไตร-สเตต โดยใช้แอนะล็อกสวิตช์ โดยที่ไมโครโปรเซสเซอร์จะให้เอาต์พุตทาง I/O ขา PC5,PC6 แอดเดรสที่ D002H ก่อนที่จะเข้าสู่แอนะล็อกสวิตช์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4-11c แสดงถึงขาอินพุตของความถี่ป้อนกลับจากกระบวนการ(process) เข้าที่อินพุตทาง I/O แอด-เดรสที่ D002H ซึ่งความถี่ป้อนกลับต้องมีดิวตี้ไซเคิล(duty cycle)เป็น 50 เปอร์เซ็นต์เพื่อการแสดงผลได้ถูกต้อง ซึ่งจะไม่มีผลต่ออัลกอลิธึมโปรแกรมของเฟสล็อก

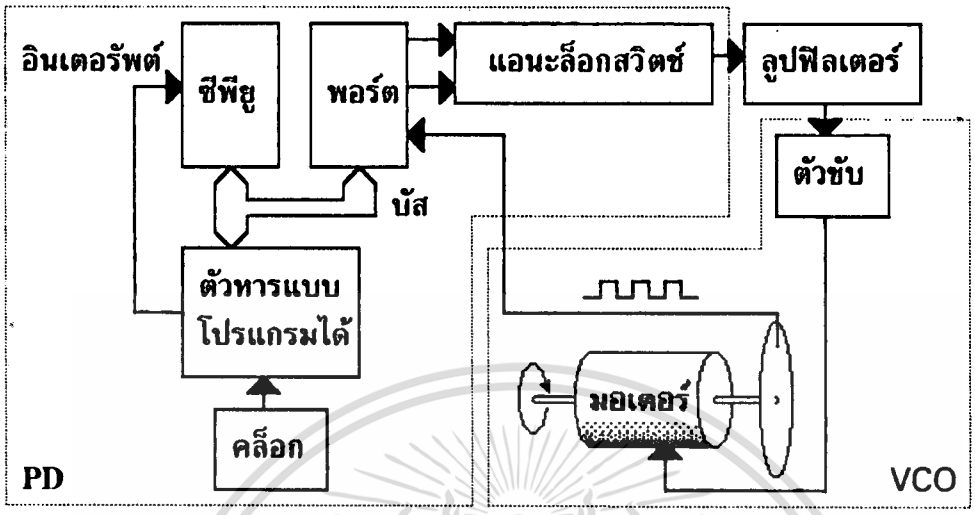


รูปที่ 4-11 แสดงบล็อกไดอะแกรมของวงจรสนับสนุนโหมด PLL

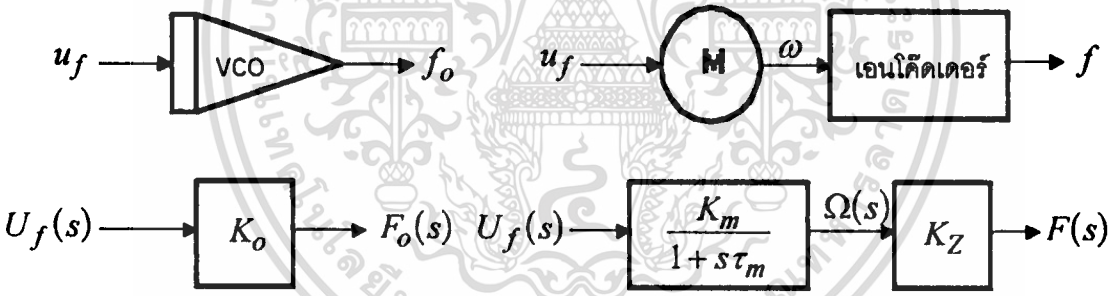
วงจรตัวอย่างซึ่งใช้ระบบเฟสล็อกควบคุมความเร็วมอเตอร์แสดงดังรูปที่ 4-12 และ รายละเอียดวงจรแสดงดังรูปที่ 4-14 อย่างไรก็ตามต้องทำงานร่วมกับซอฟต์แวร์เสมอ ซึ่งวิธีการจะได้กล่าวในบทต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4-13 แสดงให้เห็นว่ามอเตอร์ใช้แทนตัว VCO ได้อย่างไร



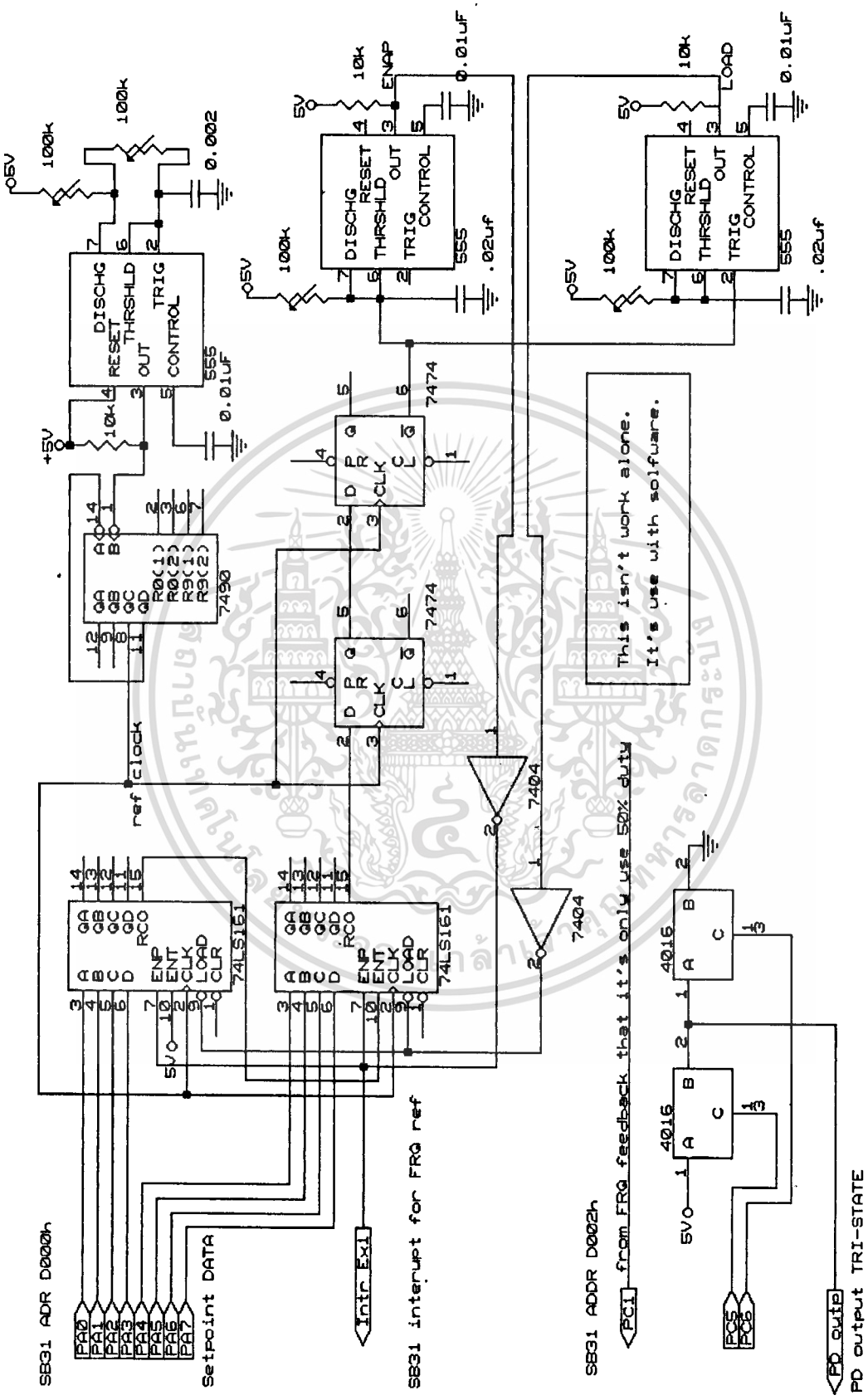
รูปที่ 4-12 *บล็อกไดอะแกรมของ DPLL ควบคุมความเร็วมอเตอร์*



$$F_o(s) = K_o U_f(s) \qquad F(s) = \frac{K_m K_z}{1+s\tau_m} U_f(s)$$

รูปที่ 4-13 *ฟังก์ชันถ่ายโอน ของ VC0 เปรียบเทียบกับมอเตอร์*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-14 วงจรดิจิทัลเฟสล็อกลูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

หลักการออกแบบซอฟต์แวร์และการใช้งาน

5.1 จุดมุ่งหมายของการออกแบบซอฟต์แวร์

บทนี้ได้ชี้ให้เห็นถึงความแตกต่างของการใช้งานระหว่าง ไมโครคอนโทรลเลอร์ และไมโครคอมพิวเตอร์ เพื่อว่าจะได้เข้าใจหน้าที่ และการใช้งานในการควบคุมกระบวนการในระบบพล้านต์ ซึ่งจะประกอบด้วยหน้าที่หลักดังนี้

1.ไมโครคอนโทรลเลอร์ใช้งานเป็นตัวดิจิทัลคอนโทรลเลอร์ ทำงานเพียงลำพัง

2.จำแนกการทำงานระบบระหว่างไมโครโปรเซสเซอร์ และไมโครคอมพิวเตอร์

3.การใช้ไมโครคอนโทรลเลอร์สำหรับฟังก์ชันควบคุมมี 4 โหมดฟังก์ชัน และ การใช้ไมโครคอมพิวเตอร์แสดงผล

เทคโนโลยีการพัฒนาไมโครคอมพิวเตอร์มุ่งเข้าสู่มินิคอมพิวเตอร์ในอดีต ซึ่งความสามารถของไมโครคอมพิวเตอร์ปัจจุบันไม่ยิ่งหย่อนกว่ามินิคอมพิวเตอร์ในอดีต ดังนั้นจึงพร้อมที่จะนำมาใช้กับ ระบบกระบวนการกระจาย(distributed processing system)ได้เป็นอย่างดี แทนที่มินิคอมพิวเตอร์ได้ แต่อย่างไรก็ดีมินิคอมพิวเตอร์ปัจจุบันได้ขยับไปใช้งานที่สูงกว่านี้ เช่น ระบบสารสนเทศ เป็นต้น

1. *ระบบการแสดงผลร่วม* หรือระบบมอนิเตอริง คือ แสดงผลโดยสังเขปของวัดค่าพารามิเตอร์ต่างๆจากพล้านต์ และการสั่งการควบคุมประกอบอยู่บนแผงควบคุม แผงควบคุมโดยมากจะประกอบด้วยหน่วยแสดงผล VDU(vidio display unit) ซึ่งโดยปกติจะประกอบด้วยเป็นจอภาพสี และอุปกรณ์ขับภาพ ส่วนเครื่องพิมพ์และส่วนเก็บข้อมูล หรือจานบันทึก(disk) ซึ่งโดยปกติจะรวมอยู่ด้วย การใช้ VDUs นี้และแผงควบคุมนี้ใช้แทนห้องคอนโทรลซึ่งเป็นพื้นที่กว้างแบบเดิม โดยที่ห้องควบคุมแบบเดิมจะประกอบด้วยหลอดไฟแสดงผลจำนวนมาก ชาร์ตเรคอร์เดอร์(chart recorder) สวิตช์ และ ปุ่มหมุน เป็นต้น

รูปที่ 5-1 แสดงถึงห้องควบคุมแบบเดิม ซึ่งที่เห็นเป็นห้องควบคุมสำหรับหลอมโลหะจะประกอบด้วยพื้นที่ของแผงควบคุมมากกว่า 90 ตารางเมตร ซึ่งมีหน่วยแสดงผล และ เรคอร์เดอร์ เป็นต้น

รูปที่ 5-2 แสดงถึงอุปกรณ์การควบคุมแบบเดียวกับรูปที่ 5-1 ลดรูปลงมาเหลือแค่แผงควบคุมแบบขนาดโต๊ะตัวหนึ่งมีจอภาพแสดงผล และคีย์บอร์ด การดูแลพล้านต์(plant supervision)แบบนี้โดยปกติจะประกอบด้วย การบันทึกแนวโน้มข้อมูล(trend recording) ระบบเตือน การปรับเซตพอย์นโดยทางคีย์บอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยปกติจะประกอบด้วย การบันทึกแนวโน้มข้อมูล(trend recording) ระบบเตือน การปรับเซตพอย์นโดยทางคีย์บอร์ด

ในประยุกต์ในระบบพล้านต์ขนาดเล็ก มีช่องอินพุต เอาต์พุตควบคุม ไม่มาก ไปจนกระทั่งระบบซับซ้อน เช่น พล้านต์ทางเคมี ซึ่งมีช่องสัญญาณอินพุต เอาต์พุตมากมาย และลูปการควบคุมจำนวนมาก สังเกตได้ว่าเป็นระบบซึ่งเตรียมการไว้อย่างครบถ้วน แม้จะมีปัญหาข้อขัดข้องก็ตาม คอนโทรลเลอร์แบบดิสครีทซึ่งโดยปกติจะใช้ไมโครโปรเซสเซอร์ในการเตรียมควบคุมแต่ละลูป ซึ่งสามารถเซตเป็นแบบแมนนวล (manual) ได้ถ้าต้องการ

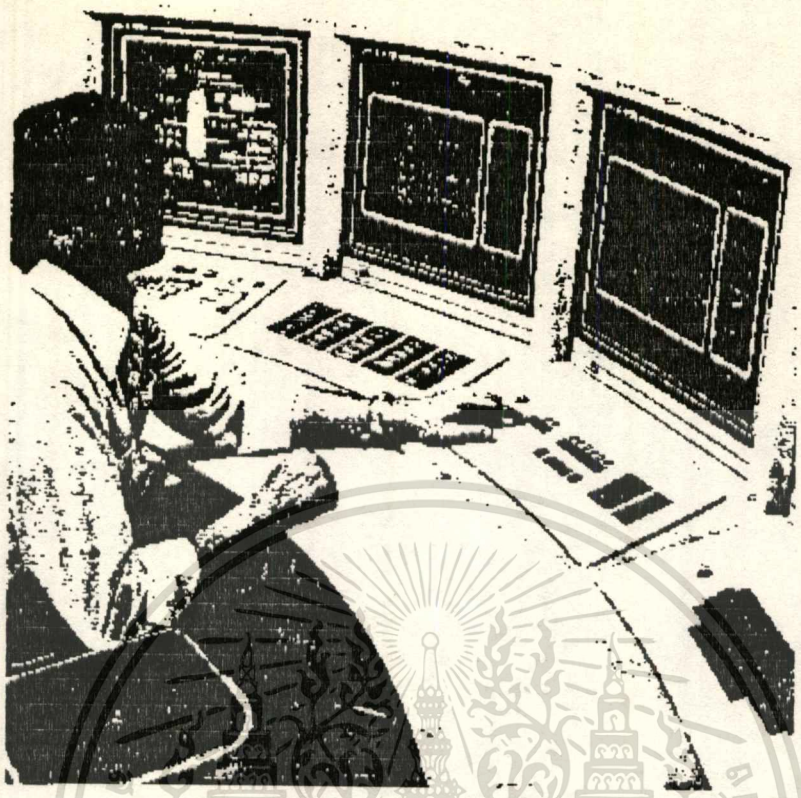


รูปที่ 5.1 *แผงควบคุมซึ่งใช้แทนห้องควบคุมแบบดั้งเดิม*

การจำแนกฟังก์ชันสำหรับการประยุกต์ในการควบคุมโดยเฉพาะ ประกอบด้วย คอนโทรล-สัปรูทีน ซึ่งบางทีใช้สัปรูทีนเป็นโมเดลทางคณิตศาสตร์ การควบคุมแบบลำดับ(sequential control) หรือการควบคุมแบบลอจิก การเก็บข้อมูลในเวลายาวนาน รวมทั้งดาต้าล็อกกิ้ง(data logging) และการพล็อต โดยปกติการทำงานจะแยกจากกันแต่ละส่วน แต่ก็จะเชื่อมโยงกับมินิคอมพิวเตอร์หรือไมโครคอมพิวเตอร์

อย่างไรก็ดีการโฆษณาเกี่ยวกับระบบพล้านต์มอนิเตอริงนั้น ส่วนใหญ่จะได้รับการต้อนรับจากระบบอุตสาหกรรมขนาดใหญ่เป็นส่วนมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.2 *ห้องควบคุม และเครื่องมือต่างๆสำหรับการควบคุมแบบใหม่*

5.2 วิธีการออกแบบซอฟต์แวร์โปรแกรมพล้านต์มอนิเตอริง

5.2.1 แนวคิดการออกแบบโปรแกรมพล้านต์มอนิเตอริง

ระบบแสดงผลของพล้านต์(plant display)ของโปรแกรมพล้านต์มอนิเตอริง เป็นแนวความคิดที่จะวัดค่าต่างๆของตัวแปรที่อยู่ในพล้านต์ เช่น ความดัน ความสูง น้ำหนักเป็นต้น ซึ่งสามารถแสดงผลที่ได้บนบนแผงควบคุม(control panel)ซึ่งประกอบด้วย ซึ่งประกอบด้วยจอภาพ คีย์บอร์ด เครื่องพิมพ์ และส่วนที่จัดเก็บข้อมูล คือ จานบันทึก(floppy disk) และจานบันทึกแบบแข็ง(hard disk)

โปรแกรมพล้านต์มอนิเตอริง เป็นโปรแกรมสำหรับคอยแสดงผลของการตอบสนองของพารามิเตอร์ต่างๆของตัวดิจิทัลคอนโทรลเลอร์ที่อยู่ในพื้นที่พล้านต์ ซึ่งมีการแสดงผลเป็นแบบกราฟบันทึกแนวโน้มข้อมูล(trend recorder) กราฟบาร์(bar graph) การเซตป้าย(tag setting) พร้อมทั้งมีระบบยูสเซอร์อินเตอร์เฟส(user interface)ที่สมบูรณ์ ดังนั้นในการออกแบบโปรแกรมจะต้องจัดการทำงานของซอฟแวร์โดยแบ่งการทำงาน และการจัดการหัวข้อหลักดังนี้

-ใช้ยูสเซอร์อินเตอร์เฟส เป็นแบบป๊อป-อัปพูลดาวน์เมนู(pop-up pull down menu) อีกทั้งมีเมนูย่อย(sub menu) และเมนูแบบบาร์ เพื่อให้การใช้งานโปรแกรมนั้นเข้าใจได้ง่ายขึ้น

-มีข้อความช่วยเหลือเบื้องต้นในส่วนต่างๆตลอดโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-มีฟังก์ชันการทำงานเกี่ยวกับกราฟบันทึกแนวโน้มข้อมูล ซึ่งสามารถกำหนดจำนวนกราฟบนจอภาพได้ตั้งแต่ หนึ่งกราฟ สองกราฟ และสี่กราฟ การเลือกตัวคอนโทรลเลอร์มาแสดงผลโดยกำหนดเป็นชื่อหมายเลข การกำหนดฐานเวลาในแต่ละกราฟ การเลือกเวลาแซมปลิง เป็นต้น

-มีการเซตป้าย(tag setting)ของแต่ละตัวคอนโทรลเลอร์ โดยผู้ใช้สามารถเปิดไฟล์ใหม่(ไฟล์นามสกุล TAG)เพื่อใช้เฉพาะในพล้านต์ใดๆ โดยที่ผู้ใช้ไม่ต้องแก้ไขข้อมูลป้ายบ่อยๆ

-การทำงานเกี่ยวกับบาร์กราฟ(bar graph) แสดงผลเป็นกราฟแนวตั้งหรือกราฟแท่ง โดยมี 12 กราฟบนหนึ่งจอภาพ ซึ่งครบจำนวนของคอนโทรลเลอร์ทั้งหมดที่ คอมพิวเตอร์สามารถเชื่อมโยงได้

-มีฟังก์ชันสัปรูทีนจัดการเกี่ยวกับการเชื่อมต่ออนุกรม และจัดการเกี่ยวกับไทเมอร์

-มีฟังก์ชันสัปรูทีนจัดการเกี่ยวกับโครงแบบ (configulation)

5.2.2 ทำไมต้องเลือกใช้ภาษา C++

เราถือว่า C++ เป็นภาษาทางคอมพิวเตอร์แห่งปี 1990 ซึ่งดัดแปลงมาจากภาษาซี ซึ่งจุดเด่นของภาษาซีคือ การมีโครงสร้างที่ดี ทั้งในแง่โครงสร้างของภาษาและโครงสร้างชนิดของข้อมูล

ภาษา C++ เอาหลักการโปรแกรม 2 แบบไว้ด้วยกัน คือการเขียนโปรแกรมเป็นโพรซีเดอร์ต่างๆ หรือมีความหมายตรงกับฟังก์ชัน หรือสัปรูทีน(subroutine)ในภาษาซี กับหลักการเขียนโปรแกรมเชิงวัตถุ โดยอ้างอิงแนวคิดแบบออปเจ็กต์โอเรียนเต็ด(object oriented) โดยจะมองโปรแกรมเป็นระบบที่น่าสนใจระบบหนึ่ง และมีออปเจ็กต์เป็นสมาชิกในโปรแกรม ในออปเจ็กต์หนึ่งจะประกอบไปด้วย เซกเมนต์รหัส(code segment) คือ สัปรูทีนจัดการต่างๆ และ เซกเมนต์ข้อมูล(data segment) อันได้แก่ ตัวแปรในโปรแกรม

หลักการเขียนโปรแกรมแบบโครงสร้างช่วยทำให้โปรแกรมดูดีขึ้น และง่ายต่อการบำรุงรักษา โดยเน้นที่อัลกอลิธึม ส่วนโปรแกรมเชิงวัตถุจะเน้นที่ดาต้า(data) แทนที่พยายามจะใส่ปัญหาต่างๆไว้ในโพรซีเดอร์ โดยที่ออปเจ็กต์โอเรียนเต็ดโปรแกรม หรือ OOP พยายามจะใส่ปัญหาแนวคิด คือการออกแบบรูปแบบของข้อมูลให้สอดคล้องกับลักษณะที่จำเป็นของปัญหา

อย่างไรก็ตามซอร์สโค๊ดของโปรแกรมพล้านมอนิเตอริงนี้ต้องคอมไพล์ด้วย ตัวคอมไพเลอร์ของ TURBO C++ ตั้งแต่รุ่นที่หนึ่ง ขึ้นไป

สำหรับการเขียนซอร์สโค๊ด(sorce code)จะเขียนเอาไว้หลายไฟล์ด้วยกัน ซึ่งแต่ละไฟล์จะประกอบด้วยสัปรูทีนซึ่งมีหน้าที่ต่างกัน ในตัวซอฟต์แวร์โปรแกรมแบ่งเป็นไฟล์ย่อย และการจำแนกหน้าที่ ดังต่อไปนี้

1.BARGRAPH.CPP เป็นไฟล์สำหรับใช้งานในการเขียนบาร์กราฟ(bar graph)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.DOPULL.CPP เป็นไฟล์มีหน้าที่ในการจัดการทำงานในเมนูย่อยต่างๆ

3.MOUSEOBJ.CPP เป็นไฟล์เพื่อให้โปรแกรมพล้านต์พล้านต์มอนิเตอริง ใช้งานกับเมาส์ได้

4.PULLDOWN.CPP เป็นไฟล์สำหรับสร้างเมนูแบบพลูดาวน์

5.SUBMENU.CPP เป็นไฟล์สำหรับสร้างหัวข้อของเมนูย่อย

6.T&SINTR.CPP เป็นไฟล์สำหรับอินเตอร์รัพต์ของ ไทเมอร์ และอินเตอร์รัพต์อนุกรม

7.TAGSET.CPP เป็นไฟล์สำหรับการสร้างแทกไฟล์(tag files) และเซตพารามิเตอร์ของกราฟ

8.PNTMNT.CPP เป็นไฟล์หลัก(main file) ของโปรแกรม

9.TIMEINTR.CPP เป็นไฟล์สำหรับอินเตอร์รัพต์ไทเมอร์

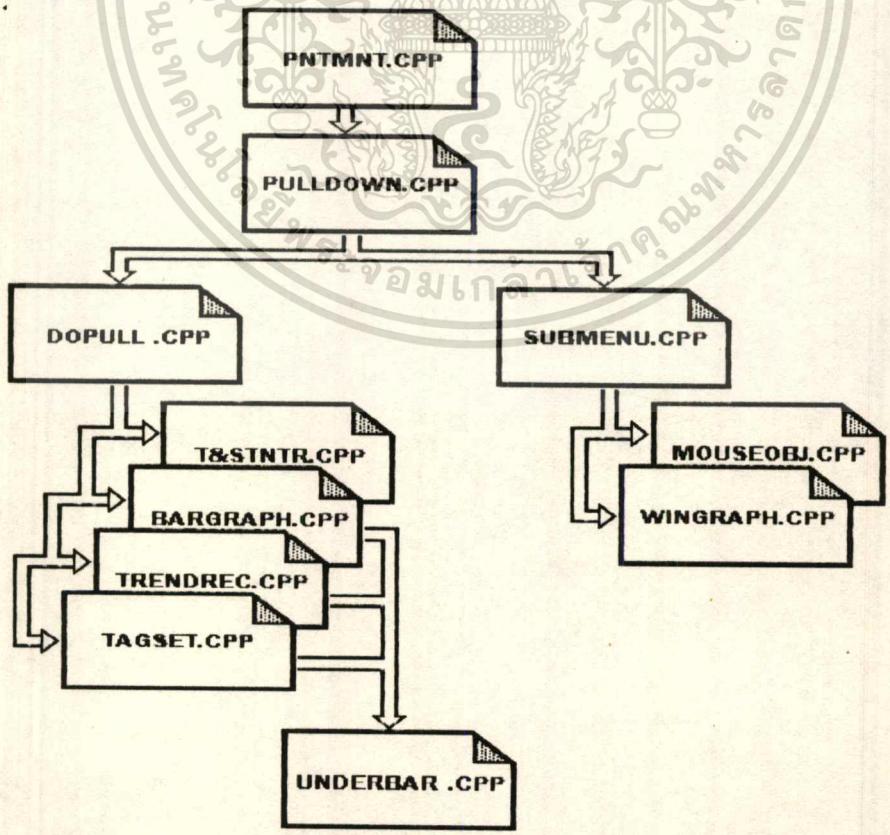
10.UNDERBAR.CPP เป็นไฟล์สำหรับสร้างเมนูแบบบาร์ซึ่งจะปรากฏอยู่ด้านล่างจอภาพ

11.WINGRAPH.CPP เป็นไฟล์สำหรับสร้างวินโดว์(window)

12.?.TAG เป็นไฟล์พิเศษที่โปรแกรมเขียนขึ้นเองอัตโนมัติเพื่อเก็บค่าพารามิเตอร์

13. PLANT.CFG เป็นไฟล์สร้างขึ้นมาเพื่อเก็บ การจัดรูปแบบ(configulation) ของโปรแกรม

5.2.3 โครงสร้างของซอฟต์แวร์โปรแกรม



รูปที่ 5-3 *โครงสร้างไฟล์มอดูล*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการจำแนกซอร์สโค๊ดแบ่งเป็นแต่ละไฟล์ข้างต้นนั้น จะต้องมีความเกี่ยวพันของไฟล์ต่างๆซึ่งประกอบกันเป็นโครงสร้างของตัวโปรแกรมนั้น สามารถเขียนเป็นบล็อกของมอดูล ตามรูปที่ 5-3

จากรูปสังเกตได้ว่าบล็อกไฟล์มอดูลต่างๆ ซึ่งไฟล์เหล่านั้นจะประกอบกันเป็นโปรแกรม บล็อกมอดูลข้างบนสุดชื่อ PNTMNT.CPP จะเป็นไฟล์หลัก ซึ่งจะเรียกใช้งานตามลำดับบล็อกของไฟล์ต่างๆซึ่งอยู่ด้านฐาน โดยที่จะเรียกใช้งานตามหน้าที่ของโปรแกรมพื่อให้ได้ตามต้องการ

5.2.3 หน้าที่ของสับรูทีนในไฟล์ต่าง ๆ

ในแต่ละไฟล์ตามรายการข้างต้น จะประกอบด้วยหลายสับรูทีนด้วยกัน สับรูทีนเหล่านั้นจะเป็นโพรซีเดอร์(procedure) หรือฟังก์ชัน(function)ก็ได้ ซึ่งในแต่ละสับรูทีนถูกเรียกใช้งานตามโปรแกรมในหน้าที่ต่างๆกัน โดยสามารถจำแนกสับรูทีนในไฟล์แต่ละไฟล์ดังนี้ ในที่นี้จะระบุหน้าที่ของสับรูทีนเพียงคร่าวๆ ส่วนรายละเอียดในส่วนลึกดูในตัวโปรแกรมประกอบ

1.สับรูทีนที่สำคัญในไฟล์ PNTMNT.CPP ประกอบด้วยสับรูทีนต่างๆดังนี้

- GRAPHINIT เป็นสับรูทีนติดตั้งกราฟิกไดรเวอร์ให้เหมาะสมกับจอภาพ เช่น VGA
- MAIN เป็นเมนสับรูทีน ซึ่งเป็นสับรูทีนหลักของโปรแกรม

2.สับรูทีนที่สำคัญในไฟล์ PULLDOWN.CPP เป็นไฟล์สำหรับสร้างเมนูแบบพูลดาว์น ซึ่งใช้เป็นแบบมาตราฐานของหลักการยูสเซอร์ ในไฟล์ตัวตัวแปร hdtxt เป็นตัวแปรแบบเรคอร์ด โดยใช้กำหนดลักษณะของเมนูพูลดาว์น เช่น กำหนดสี กำหนดตำแหน่ง กรอบของเมนูเป็นต้น ไฟล์นี้ยังประกอบด้วยสับรูทีนต่างๆดังนี้

-BARHEAD เป็นสับรูทีนใช้สร้าง high light บนเมนูหลัก(main menu)ของโปรแกรม

-BAROFF เป็นสับรูทีนที่ใช้ตรงข้ามกับสับรูทีน BARHEAD คือจะลบไฮไลต์(high light) ออกจากเมนูหลัก

-PULLSHOW เป็นสับรูทีนทำหน้าที่นำเมนูแบบพลูดาวน์ มาแสดงบนจอซีเกต(CRT) ตามตัวเลือกของเมนูหลัก โดยมีตัวแปร statushead เป็นตัวชี้

-CHOICELF เป็นสับรูทีนสำหรับเลื่อนไฮไลต์(high light) ของเมนูหลักไปทางซ้าย ซึ่งควบคุมได้จากคีย์บอร์ดหรือว่าเมาส์

-CHOICERT คล้ายกับสับรูทีน CHOICELF แต่จะเลื่อนไปทางซ้าย

-HEADSELECT เป็นสับรูทีนที่สำคัญที่สุดในไฟล์นี้ คือมีหน้าที่ในการจัดการรับคีย์บอร์ดและเมาส์โดยจะจำแนกออกมาว่าต้องการทำอะไร อย่างเช่นถ้ากดปุ่ม ENTER คือทำให้

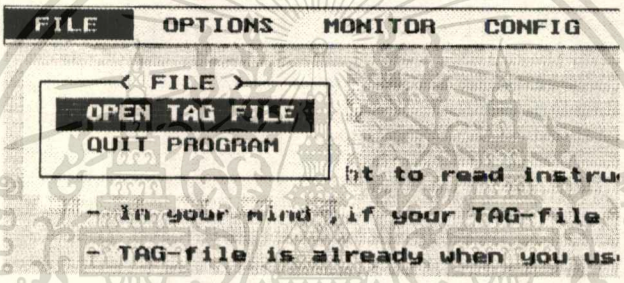
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชันตามเมนูย่อยที่ไฮไลต์(high light) จะปรากฏ โดยมีสับรูทีนฟังชัน DOPULLDOWN รับช่วงไปทำ

-SHOWMAIN เป็นสับรูทีนสำหรับสร้างไตเติล และแสดงเมนูหลักบนจอ

-DOPULLDOWN เป็นสับรูทีนรับช่วงการทำงาน HEADSELECT

3. สับรูทีนของไฟล์ DOPULL.CPP มีรูทีนย่อยตั้งแต่สับรูทีน DO00 ถึง DO03 จะเป็นการตอบสนองในเมนูหลัก FILE สับรูทีน DO10 ถึง DO13 ตอบสนองเมนูหลัก OPTION สับรูทีน DO20 ถึง DO23 ทำงานตอบสนองเมนูหลัก MONITOR และ สับรูทีน DO30 ถึง DO33 ตอบสนองการทำงานของเมนูหลัก CONFIG ซึ่งสามารถจำแนกการทำงานของแต่ละสับรูทีนได้ดังนี้



รูปที่ 5-4 *แสดงเมนูหลัก FILE และเมนูย่อยต่างๆซึ่งตอบสนองการทำงานของสับรูทีน DO00 ถึง DO03*

-DO00 เป็นสับรูทีนซึ่งทำงานตอบสนอง เมนูย่อย OPEN TAG FILE โดยทำหน้าที่จะเปิดไฟล์ตามที่ผู้ใช้กำหนด

-DO01 เป็นสับรูทีนซึ่งตอบสนองการทำงาน เมนูย่อย QUIT โดยทำหน้าที่ออกจากโปรแกรมถาวร

-DO02 เป็นสับรูทีนสำรอง เพื่อขยายโปรแกรมในอนาคต

-DO03 เป็นสับรูทีนสำรอง เพื่อขยายโปรแกรมในอนาคต

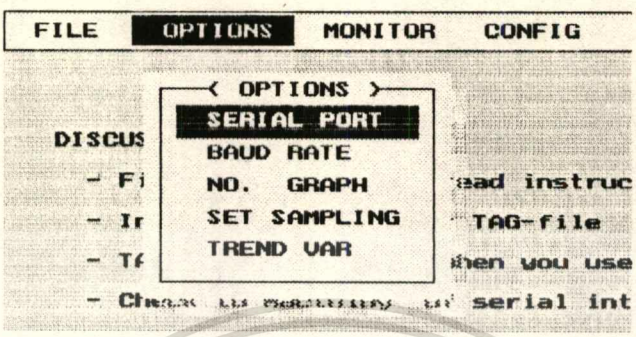
-DO10 เป็นสับรูทีนตอบสนองการทำงาน ของเมนูย่อย SERIAL PORT โดยทำหน้าที่เลือกหมายเลขพอร์ตอนุกรม

-DO11 เป็นสับรูทีนตอบสนองการทำงาน ของเมนูย่อย BAUD RATE โดยให้ตัวเลือกอัตราบอดค่าต่างเพื่อให้ผู้ใช้เลือก

-DO12 เป็นสับรูทีนตอบสนองการทำงาน ของเมนูย่อย NO GRAPH โดยจำนวนกราฟที่ปรากฏบนหนึ่งจอภาพสามารถกำหนดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-DO13 เป็นสับรูทีนตอบสนองการทำงาน ของเมนูย่อย SET SAMPLING โดยผู้ใช้จะต้องใช้จำนวนค่าตัวเลขพร้อมทศนิยมลงไปในกรอบที่กำหนด



รูปที่ 5-5 *แสดงเมนูหลัก OPTION และเมนูย่อยต่างๆซึ่งตอบสนองการทำงานของสับรูทีน DO10 ถึง DO13*



รูปที่ 5-6 *แสดงเมนูหลัก monitor และเมนูย่อยต่างๆซึ่งตอบสนองการทำงานของสับรูทีน DO20 ถึง DO23*

-DO20 เป็นสับรูทีนตอบสนองการทำงานของเมนูย่อย TAG EDITING เกี่ยวกับการแก้ไขไฟล์ *.tag

-DO21 เป็นสับรูทีนตอบสนองการทำงานของเมนูย่อย MONITOR

-DO22 เป็นสับรูทีนตอบสนองการทำงานของเมนูย่อย TREND REC ในรูทีนี้ทำงานเกี่ยวกับแสดงผลของกราฟบันทึกแนวโน้มข้อมูล

-DO23 เป็นสับรูทีนตอบสนองการทำงานของเมนูย่อย BAR GRAPH ทำงานเกี่ยว กับการสร้างกราฟบาร์

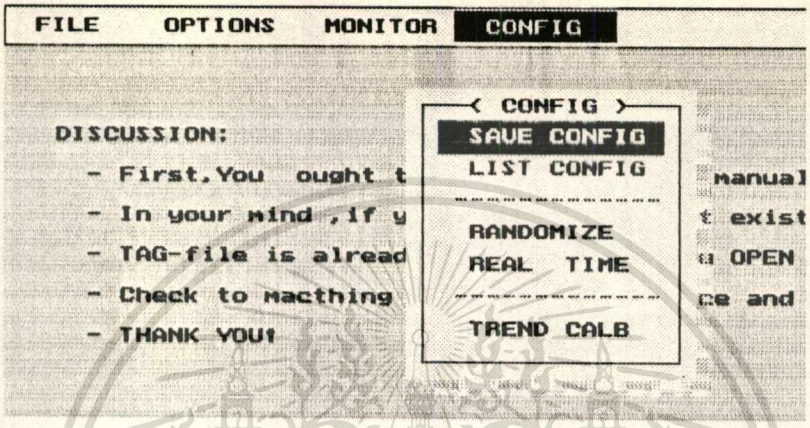
-DO30 เป็นสับรูทีนตอบสนองการทำงานของเมนูย่อย SAVE CONFIG โดยจะเก็บข้อมูลลงในไฟล์ plant.cfg

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-DO31 เป็นสับรูทีนตอบสนองการทำงานของเมนูย่อย LIST CONFIG โดยจะแสดงรายค่าการจัดลักษณะ(configulation) ของโปรแกรม

-DO32 สำรองไว้ในอนาคต

-DO33 สำรองไว้ในอนาคต



รูปที่ 5-7 *แสดงเมนูหลัก CONFIG และเมนูย่อยต่างๆซึ่งตอบสนองการทำงานของสับรูทีน DO30 ถึง DO33*

4.สับรูทีนของไฟล์ TAGSET.CPP เป็นไฟล์สำหรับใช้เกี่ยวกับการเซตป้าย(TAG) คือให้ป้ายชื่อกับกราฟการตอบสนองของคอนโทรลเลอร์ โดยจะถามผู้ใช้เกี่ยวกับพารามิเตอร์ที่สำคัญไฟล์นี้มีตัวแปรเรคอร์ดคือ taglist โดยตัวแปรนี้ประกอบด้วยตัวแปรย่อยที่ใช้ในเมนูย่อย TAG EDITING ในมีสับรูทีน การใช้งานดังนี้ต่อไปนี้

-SHOWTAGTBL เป็นสับรูทีนแสดงตาราง TAG table

-TAGMENU เป็นสับรูทีนสำหรับสร้างช่องใส่ค่าชื่อป้าย (tag name) ,คำบรรยาย(description)

-GETUNIT เป็นสับรูทีนทำหน้าที่รับคีย์บอร์ดเพื่อใส่หมายเลขยูนิตของคอนโทรลเลอร์

-GETTAG เป็นสับรูทีนทำหน้าที่รับคีย์บอร์ดเพื่อกำหนดป้ายชื่อหรือ TAG name

-GETDISCP เป็นสับรูทีนทำหน้าที่รับคีย์บอร์ด เพื่อใส่คำบรรยายของคอนโทรลเลอร์

-GETUNIT เป็นสับรูทีนทำหน้าที่รับคีย์บอร์ด เพื่อใส่หน่วยทางวิศวกรรมของขนาดตัวแปรที่ใช้ ในคอนโทรลเลอร์

-GETUPPER เป็นสับรูทีนทำหน้าที่รับคีย์บอร์ด เพื่อใส่ค่าขนาดสูงสุดเทียบเป็น 100 เปอร์-เซ็นต์

-GETLOWER เหมือนกับสับรูทีน GETUPPER แต่เป็นค่าขนาดต่ำสุดเทียบเป็น 0 เปอร์เซ็นต์

-GETSAMP เป็นสับรูทีนสำหรับรับค่าเวลาแซมปลิงจากคีย์บอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อไปนี้เป็นสัปรูทีนสำหรับแสดงตารางของ TAG

-INITTAG ใส่ค่าเริ่มต้น(initial ให้กับตัวแปร taglist

-PUTTAGTBL วาดตาราง TAG table บนจอภาพ

-EDITTAG เป็นสัปรูทีนสำหรับจัดการเกี่ยวกับการใส่ค่าพารามิเตอร์ TAG โดยใช้สัปรูทีน GET... จัดการ

-GETFNAME เป็นสัปรูทีนสำหรับใส่ชื่อไฟล์ *.TAG จากคีย์บอร์ด

-NEWTAGF เป็นสัปรูทีนคอยตรวจสอบว่าชื่อไฟล์ซึ่งได้จากสัปรูทีน GETFNEME เป็นไฟล์ใหม่หรือ ไฟล์ที่มีอยู่ก่อน ถ้าเป็นชื่อใหม่ก็จะสร้างไฟล์ใหม่โดยมีนามสกุล *.TAG เสมอ

-LOAGTAGF เป็นการโหลดไฟล์ซึ่งมีอยู่ก่อนมาใช้

-TAGWORK เป็นสัปรูทีนจัดการเกี่ยวกับการเปิดไฟล์ TAG โดยใช้ไฟล์ ...TAGF เป็นสัปรูทีนช่วยเหลือ ซึ่งสั่งการมาจากคีย์บอร์ด

5.สัปรูทีนของไฟล์ BARGRAPH.CPP เป็นไฟล์ที่มีสัปรูทีนต่างๆซึ่งช่วยเหลือเกี่ยวกับการสร้างบาร์กราฟ(bar graph) มีตัวแปรเรคอร์ดที่สำคัญ คือ bar-trend มีหน้าที่สำหรับเก็บพารามิเตอร์ที่สำคัญในการสร้างบาร์กราฟ และมีสัปรูทีนใช้งานดังนี้

- BARCREAT เป็นสัปรูทีนสร้างวาดกรอบภาพและสเกลต่างๆให้เหมาะสม
- BARTREND เป็นสัปรูทีนพอร์ตเส้นแท่งบาร์ PV , CV และ SP ลงไปในกรอบซึ่ง
- CLEARBAR เป็นสัปรูทีนลบแท่งบาร์ต่างๆออกจากกรอบและพร้อมที่จะวาดใหม่

6.สัปรูทีนของไฟล์ T&SINTR.CPP มีหน้าที่เกี่ยวกับการทำงานในโหมดการสื่อสารแบบอนุกรม และ ไทเมอร์

- INITSERIAL เป็นสัปรูทีนเซตค่าเริ่มต้นของการทำงานการติดต่อสื่อสารแบบอนุกรม
- DATAREADY เป็นสัปรูทีนเช็คว่าข้อมูลที่จะถูกรับเข้ามาหรือไม่
- THOLD เป็นสัปรูทีนเช็คว่าข้อมูลถูกส่งเข้าไปในสายส่งอนุกรมหรือยัง
- SERIALRECV เป็นสัปรูทีนทำให้ PC พร้อมที่เป็นตัวรับ
- SERIALSEND เป็นสัปรูทีนทำให้ PC พร้อมที่จะเป็นตัวส่ง
- TIMERINTR เป็นอินเตอร์รัพต์เซอร์วิตรูทีนของไทเมอร์
- TIMERSTART เป็นรูทีนที่อนุญาติให้เกิดหรืออีนาเบิลอินเตอร์รัพต์ไทเมอร์
- TIMERSTOP เป็นรูทีนดิสอเบิลอินเตอร์รัพต์ไทเมอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.สัปรูทีนของไฟล์ TRENDREC.CPP เป็นไฟล์สำหรับสร้างกราฟบันทึกแนวโน้มข้อมูล(trend recorder) โดยมีตัวแปรเรคอร์ดที่สำคัญคือ vdata คือเป็นตัวแปรสำหรับเก็บค่า PV ,CV และ SP ตัวแปรเรคอร์ด trend มีหน้าที่เก็บพารามิเตอร์ต่างๆซึ่งใช้พอร์ตกราฟ ไฟล์นี้มีสัปรูทีนซึ่งใช้งานที่สำคัญดังต่อไปนี้

-ACKNUMOFTREND เป็นสัปรูทีนซึ่งจะสร้างเมนูบน CRT เพื่อจะถามถึงจำนวนกราฟบันทึกแนวโน้มข้อมูล ที่ต้องการจะวาดบนจอ

-CLEARGRAPH สำหรับลบเส้นกราฟต่างๆซึ่งอยู่ในกรอบ

-TRENDCREAT สำหรับสร้างกรอบและสเกลของกราฟ TREND

-GRAPHPLOT เป็นสัปรูทีนสำหรับวาดเส้นกราฟ PV,CV และ SP ลงไปในกรอบ ซึ่งถูกสร้างไว้แล้วโดยสัปรูทีน TRENDCREAT

-PLOTSCALE เป็นสัปรูทีนสำหรับสร้างแกน Y ของกราฟ และกำหนดสเกล แล้วก็สร้าง แกน X ของกราฟและกำหนดสเกลเช่นกัน

-CUTFLOAT เป็นสัปรูทีนช่วยเหลือ สำหรับการแปลงจำนวนทศนิยมเป็นจำนวนเต็ม เพื่อให้สะดวก ในการพอร์ตกราฟ

8.สัปรูทีนของไฟล์ UNDERBAR.CPP เป็นสัปรูทีนสร้างเมนูบาร์คล้ายเมนูหลักแต่จะอยู่ด้านด้านล่างของ CRT แทน เมนูบาร์นี้จะใช้ในเมนูย่อยของ TRENDREC และ BARGRAPH ในไฟล์นี้มีตัวแปรเรคอร์ด underbar ใช้สำหรับเก็บพารามิเตอร์เกี่ยวกับการสร้างเมนูบาร์ ไฟล์นี้มีสัปรูทีนการใช้งานดังนี้

-BARON เป็นสัปรูทีนสร้างไฮไลต์เป็นตัวเลือกบนเมนู

-BAROFF ทำหน้าที่ลบไฮไลต์ออกจากตัวเลือก

-LF ทำหน้าที่เลื่อนไฮไลต์ของตัวเลือกออกไปทางซ้าย

-RT ทำหน้าที่เลื่อนไฮไลต์ของตัวเลือกออกไปทางขวา

-SELECT เป็นสัปรูทีนสำหรับรับคีย์บอร์ด และเมาส์เพื่อใช้สำหรับเลื่อนตัวชี้ตำแหน่ง(cursor) โดยใช้งาน สัปรูทีน LF และ RT อีกทอดหนึ่ง

-SHOW เป็นสัปรูทีนสำหรับสร้างเมนูบาร์บนจอ CRT ด้านล่างของจอภาพ

-SETACT เป็นสัปรูทีนสำหรับกำหนดว่าตัวเลือก(choice) ซึ่งอยู่บนเมนูบาร์ตัวไหนใช้งานบ้าง โดยกำหนดเป็น TRUE ทำงาน FALSE ไม่ทำงาน

หมายเหตุ ใน รายการ ของโปรแกรมชื่อสัปรูทีนของตามหลังชื่อ class เสมอ เพราะว่าเป็นวิธี การเขียนโปรแกรมแบบโปรแกรมเชิงวัตถุ(object oriented program)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.สับรูทีนของไฟล์ SUBMENU.CPP มีหน้าที่เป็นตัวสร้างเมนูย่อยในโปรแกรม ตัวแปรเรคอร์ด SUBM เป็นตัวแปรกำหนดคุณสมบัติของเมนูย่อย และมีคลาส CLASS ที่ชื่อว่า SUB โดยมีเมทอด(method) ของคลาสดังนี้ เมทอดเป็นเช่นเดียวกับสับรูทีน โดยเมทอดจะใช้เฉพาะในคลาสของตัวมันเองเท่านั้น

-SUB::SUBACT เป็นเมทอดให้กำหนดตัวเลือกในเมนูย่อยแอกทีฟ หรือไม่แอกทีฟ ที่ซึ่งถ้าไม่แอกทีฟแล้ว แถบไฮไลต์(hight light)จะข้ามตัวเลือกที่ไม่แอกทีฟ

-SUB::COUNTDOWN ทำหน้าที่เลื่อนไฮไลต์ลงด้านล่างหนึ่งตัวเลือก โดยตอบสนองจากคีย์บอร์ด

-SUB::COUNTUP ทำหน้าที่เลื่อนไฮไลต์ขึ้นด้านบนหนึ่งตัวเลือกโดยตอบสนองจากคีย์บอร์ด

-SUB::BARSHOW ทำหน้าที่แสดงไฮไลต์ให้ปรากฏ

-SUB::SHOWSUB แสดงเมนูย่อยให้ปรากฏบนจอภาพ

-SUB::SUBCLOSE ปิดเมนูย่อยลง ซึ่งจะไม่มีการปรากฏเมนูย่อยบนจอภาพ

-SUB::MESSAGE แสดงข่าวสารที่ต้องการบนจอ

10.สับรูทีนของไฟล์ WINGRAPH.CPP เป็นโปรแกรมช่วยสำหรับสร้างหน้าต่าง(window)บน CRT โดยสามารถเปิดหน้าต่างซ้อนทับกันได้หลายชั้น ซึ่งมีประโยชน์ในการช่วยสร้างระบบยูสเซอร์-อินเตอร์เฟส(user interface) อย่างเช่นเมนูแบบพูลดาว์นเป็นต้น ในไฟล์นี้มีสับรูทีนที่สำคัญดังนี้

-WINDOW_CREAT เป็นสับรูทีนสำหรับสร้างกรอบหน้าต่าง และเงาของหน้าต่างในแบบต่างๆ

-WINDOW_OPEN เป็นสับรูทีนสำหรับใช้เปิดหน้าต่างใหม่ซ้อนทับขึ้นมาบนจอ CRT โดยที่จะเก็บส่วนที่ถูกซ้อนทับไว้ในหน่วยความจำ

- WINDOW_CLOSE เป็นสับรูทีนสำหรับใช้ปิดหน้าต่างที่เปิดไว้สุดท้าย โดยที่ปิดแล้วส่วนพื้นที่ถูกซ้อนทับจะกลับมาเหมือนเดิม

11.สับรูทีนของไฟล์ MOUSEOBJ.CPP ในไฟล์นี้เป็นสับรูทีนสนับสนุนการทำงานของเมาส์ ซึ่งจะไม่ขี้แจงรายละเอียดของสับรูทีนภายในไฟล์

ในสับรูทีนต่างๆของแต่ละไฟล์จะมีหน้าที่ในการทำงานในโปรแกรมต่างกันไป ซึ่งไม่จำเป็นต้องเขียนแสดงเป็น โฟล์วชาร์ต(flow chart) ประกอบสับรูทีนขึ้นมาอีก เพราะว่าในรายการของรายละเอียดของโปรแกรมที่ให้มานั้นเป็นภาษาซี ซึ่งเป็นภาษาโครงสร้าง(structer language)อยู่แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 แนวทางการออกแบบซอฟต์แวร์ของตัวดิจิทัลคอนโทรลเลอร์

5.3.1 ไมโครคอนโทรลเลอร์และซอฟต์แวร์

เป็นที่รู้กันอยู่แล้วว่าไมโครคอนโทรลเลอร์พัฒนามาจากไมโครโปรเซสเซอร์ โดยมีเทคนิคการเขียนโปรแกรมที่เหมือนกัน แต่ต่างกันที่ไมโครคอนโทรลเลอร์ใช้งานได้กว้างขวางกว่า(general perpose) ซึ่งในบทความนี้จะเน้นการใช้ไมโครโปรเซสเซอร์ 8031 ซึ่งเป็นไมโครโปรเซสเซอร์ 8 บิต เหมือนกับไมโครโปรเซสเซอร์ Z80 ซึ่งเราทราบกันดีโดยมีคุณสมบัติ และข้อต่างกันดังนี้

	Z80	8031
การกำหนดขา		
จำนวนขาทั้งหมด	40	40
ขาแอดเดรส	16(fixed)	16
ขาดาต้า	8(fixed)	8
ขาอินเตอร์รัพต์	2(fixed)	2
ขา I/O	0	32
สถาปัตยกรรม		
เรจิสเตอร์ 8 บิต	20	34
เรจิสเตอร์ 16 บิต	4	2
ขนาดสแต็ก(stack)	64K	128
รอม(ROM)ภายใน	0	0
แรม(RAM)ภายใน	0	128
หน่วยความจำภายนอก	64K	128K
แฟลก(flags)	6	4
ไทเมอร์(timer)	0	2
พอร์ตขนาน(parallel port)	0	4
พอร์ตอนุกรม	0	1
เซตคำสั่ง		
คำสั่ง mov ภายนอก	4/14	2/6
คำสั่ง mov บล็อก	2/4	0
การจัดการบิต	4/4	12/12
คำสั่ง jump บนบิต	0	3/3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสตก(stack)	3/15	2/2
คำสั่ง single-byte	203	49
คำสั่ง multi-byte	490	62

28 ไบต์ 7Fh UPPER (8032 ONLY) ACCESSIBLE BY INDIRECT	SFR BIT ACCESSIBLE 80h - FFh ACCESSIBLE BY DIRECT
ACCESSIBLE BY 5Fh DIRECT AND INDIRECT	
BIT ACCESSIBLE SPACE (00 - 7F)	
REGISTER BANK	

ตารางที่ 5-1 *แสดงการจัดหน่วยความจำของไมโครคอนโทรลเลอร์ 8031*

จุดนี้ไม่ได้ชี้ว่าอย่างไหนดีกว่ากัน ซึ่งทั้งสองออกแบบการใช้งานในจุดประสงค์ต่างกัน มีบางซีพียู เช่น Z80 นั้นมีคำสั่งมากกว่าคำสั่งแบบหลายไบต์(multi-byte) ถึง 71 เปอร์เซ็นต์ ซึ่งการใช้คำสั่งแบบหลายไบต์ นี้ต้องเสียเวลาเฟทช์คำสั่ง(fetch)จากหน่วยความจำของโปรแกรม ดังนั้นในลักษณะงานที่คล้ายกัน 8031 จะทำงานได้เร็วกว่า

การใช้งานพื้นที่แรมภายใน(internal data RAM) ของไมโครคอนโทรลเลอร์ตระกูล MCS51 จัดแบ่งหน่วยความจำภายในของไมโครคอนโทรลเลอร์แสดงเป็นตาราง 5-1 ได้ดังนี้

หน่วยความจำแรมภายใน(internal RAM) ดังกล่าวนำใช้ประโยชน์โดยมาใช้เป็นพื้นที่กำหนดตัวแปรต่างๆ ซึ่งใช้ในมอนิเตอร์โปรแกรมของวงจรดิจิทัลคอนโทรลเลอร์ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.พื้นที่ระหว่างแอดเดรสของหน่วยความจำที่ 30H - 7FH

ADDRESS RAM แอดเดรสแรม	VARIABLE ตัวแปร	คำบรรยาย
30H	KPCount	ตัวนับในโหมด KP
31H	KDCount	ตัวนับในโหมด KD
32H	KICount	ตัวนับในโหมด KI
33H	MDCount	ตัวนับสำหรับรูทีนเลือกโหมด
34H	MDWork	ค่าโหมดสำหรับทำงานปัจจุบัน
35H	ADBuff	บัฟเฟอร์เก็บค่าจาก A/D คอนเวอร์เตอร์
36H	DABuff	บัฟเฟอร์สำหรับ D/A คอนเวอร์เตอร์
37H	KeyBuff	บัฟเฟอร์สำหรับวงจรรับข้อมูลคีย์บอร์ด
38H	TogBuff	บัฟเฟอร์สำหรับ toggle ของการแสดงผล
39H	SetPoint	สำหรับเก็บค่าเซตพอย์นก่อนใช้(preload)
3AH	Status	เก็บสถานะ(status)ของคีย์บอร์ด
3BH	PLLCoun	ตัวนับสำหรับโหมด PLL
3CH	T0Count	ตัวนับสำหรับ Timer
3DH		
3EH	DCPID	Data control ของ PID-control
3FH	DSP	Data ของ เซตพอย์นซึ่งใช้ในโปรแกรม
40H	DCP	Data control ของโหมด P-control
41H	DC0	Data control เมื่อแซมปลิงก่อนหน้า
42H	KP0	เป็นไบต์แรกของค่า KP
43H	KP1	เป็นไบต์ที่สองของค่า KP
44H	DCI	Data control ของโหมด I-control
45H	KI0	เป็นไบต์แรกของค่า KI
46H	KI1	เป็นไบต์สองของค่า KI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.พื้นที่ระหว่างแอดเดรสของหน่วยความจำที่ 30H - 7FH

ADDRESS RAM แอดเดรสแรม	VARIABLE ตัวแปร	คำบรรยาย
30H	KPCount	ตัวนับในโหมด KP
31H	KDCount	ตัวนับในโหมด KD
32H	KICount	ตัวนับในโหมด KI
33H	MDCount	ตัวนับสำหรับรูทีนเลือกโหมด
34H	MDWork	ค่าโหมดสำหรับทำงานปัจจุบัน
35H	ADBuff	บัฟเฟอร์เก็บค่าจาก A/D คอนเวอร์เตอร์
36H	DABuff	บัฟเฟอร์สำหรับ D/A คอนเวอร์เตอร์
37H	KeyBuff	บัฟเฟอร์สำหรับวงจรรับข้อมูลคีย์บอร์ด
38H	TogBuff	บัฟเฟอร์สำหรับ toggle ของการแสดงผล
39H	SetPoint	สำหรับเก็บค่าเซตพอย์นก่อนใช้(preload)
3AH	Status	เก็บสถานะ(status)ของคีย์บอร์ด
3BH	PLLCoun	ตัวนับสำหรับโหมด PLL
3CH	T0Count	ตัวนับสำหรับ Timer
3DH		
3EH	DCPID	Data control ของ PID-control
3FH	DSP	Data ของ เซตพอย์นซึ่งใช้ในโปรแกรม
40H	DCP	Data control ของโหมด P-control
41H	DC0	Data control เมื่อแซมปลิงก่อนหน้า
42H	KP0	เป็นไบต์แรกของค่า KP
43H	KP1	เป็นไบต์ที่สองของค่า KP
44H	DCI	Data control ของโหมด I-control
45H	KI0	เป็นไบต์แรกของค่า KI
46H	KI1	เป็นไบต์สองของค่า KI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ADDRESS RAM แอดเดรสแรม	VARIABLE ตัวแปร	คำบรรยาย
47H	DT	เป็นค่าเวลาแซมปลิง
48H	SUM0	เป็นตัวแปร SUM ไบต์แรกใช้ในโหมด integal
49H	SUM1	เป็นตัวแปร SUM ไบต์สองใช้ในโหมด integal
4AH	DCD	Data control ของโหมด I-control
4BH	DV0	เป็นข้อมูลจาก A/D แซมปลิงก่อนหน้า
4CH	DE	เออเรอร์เอาต์พุต DSP - DV0
4DH	KD0	เป็นไบต์แรกของค่า KD
4EH	KD1	เป็นไบต์สองของค่า KD
4FH	DCPI	Data control ของโหมด PI-control
50H - 7FH	stack area	เป็นพื้นที่สำหรับ stack

2. พื้นที่หน่วยความจำระหว่างแอดเดรส 20H-2FH โดยพื้นที่นี้จะใช้ตัวแปรไบต์ร่วมกับบิตแฟลกดังนี้

ADDRESS RAM แอดเดรสแรม	VARIABLE ตัวแปร	คำบรรยาย
20H	Flag area	พื้นที่ที่ใช้เป็นแฟลก
21H		
22H		
23H		
24H		
25H	reserve	พื้นที่สำรองสำหรับใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ADDRESS RAM แอดเดรสแรม	VARIABLE ตัวแปร	คำบรรยาย
26H		
27H		
28H		
29H		
2AH		
2BH		
2CH		
2DH		
2EH		
2FH		

3.การกำหนดบิตแฟลก ใช้หน่วยความจำระหว่างแอดเดรสที่ 20H - 24H มีการใช้งานดังนี้

แอดเดรส	หมายเลขบิต	ตัวแปร	คำบรรยาย
20H	0	SGNFg	คือ sign flag แฟลกเครื่องหมาย
	1	SGNDFg	คือแฟลกเครื่องหมายของโหมด differential
	2	Multc	คือ multiply carry flag
	3	SGNIFg	คือแฟลกเครื่องหมายของโหมด integal
	4	CurBlinkFg	คือแฟลกสำหรับกำหนดให้ตัวชี้กระพริบ
	5	ToggleFg	คือแฟลกกำหนด toggle ของการแสดงผล
	6	EdFg	คือแฟลกกำหนด edit โหมด
	7	Num1Fg	คือแฟลกกำหนด bank ข้างหน้าตัวเลขแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอดเดรส	หมายเลขบิต	ตัวแปร	คำบรรยาย
21H	0(08)	PVFg	คือแฟลกแสดงผลของ PV (process VRB)
	1(09)		
	2(0A)		
	3(0B)		
	4(0C		
	5(0D)		
	6(0E)	Intr2ActFg	คือแฟลกแสดงว่า serial INTR ทำงาน
	7(0F)	PVTogFg	คือแฟลกกำหนด toggle ของ PV
22H	0(10)		
	1(11)		
	2(12)		
	3(13)		
	4(14)		
	5(15)		
	6(16)		
	7(17)		
23H		reserve	พื้นที่สำรองสำหรับบิตแฟลก
24H		reserve	พื้นที่สำรองสำหรับบิตแฟลก

5.3.2 โครงสร้างของโปรแกรมมอนิเตอร์ของตัวดิจิทัลคอนโทรลเลอร์

หลักการออกแบบซอฟต์แวร์ เพื่อเป็นโปรแกรมมอนิเตอร์ให้กับตัวคอนโทรลเลอร์ โดยใช้ไมโครโปรเซสเซอร์ตระกูล MPS-51 ของบริษัทอินเทล โดยใช้สัญญาณนาฬิกา 12 เมกะเฮิรตซ์ จึงมีความเร็วพอเพียงที่จะเขียนซอฟต์แวร์โปรแกรม ซึ่งโปรแกรมนี้จะใช้งานร่วมกับกระบวนการ ทำให้โปรแกรมนี้ต้องทำงานในลักษณะเป็นเวลาจริง(real time) โดยที่ควบคุมลูบคอนโทรลได้ 1 ลูบซึ่งสามารถต่อแบบ เครือข่าย สามารถควบคุมลูปคอนโทรลในพล้านต์(plant)ได้ถึง 12 ลูปโดยใช้ โฮส(host) ช่วยเหลือ โปรแกรมสำหรับตัวคอนโทรลเลอร์นี้แบ่งออกเป็น 4 โหมดด้วยกัน ดังนี้

1. โหมดการควบคุมพรอปพรอชันแนล หรือ P

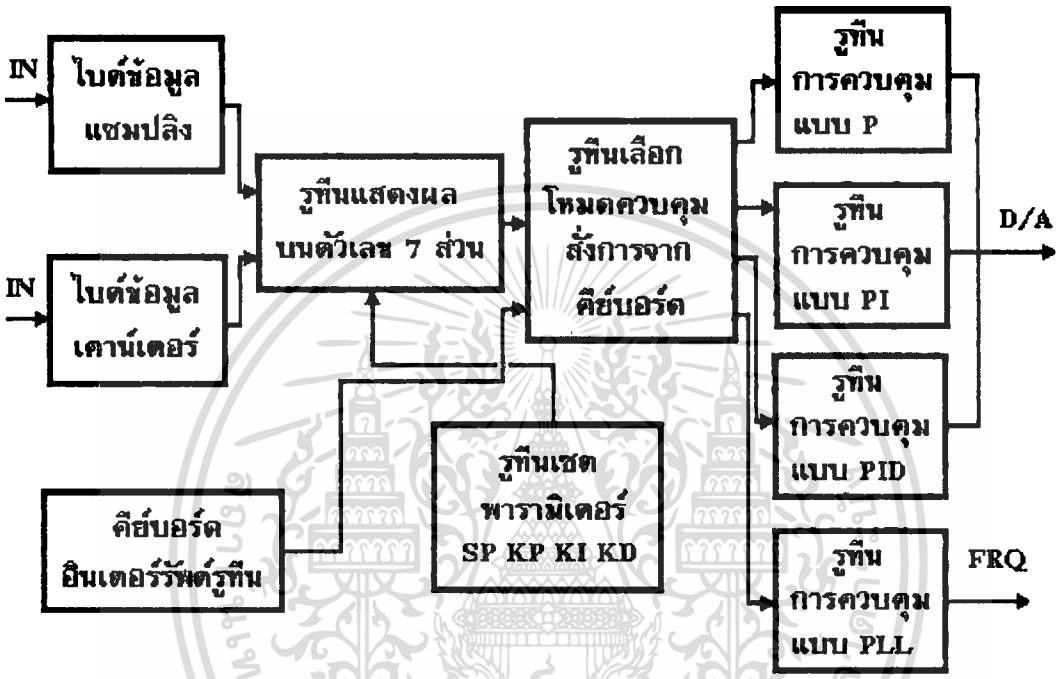
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. โหมดการควบคุมพรอปพรอชันแนล-อินทีกรัล หรือ PI

3. โหมดการควบคุมพรอปพรอชันแนล-อินทีกรัล-ดีริเวทีฟ หรือ PID

4. โหมดการควบคุมเฟสล็อก หรือ PLL

มีบล็อกไดอะแกรมการทำงานของโปรแกรมของซอฟต์แวร์รูทีนต่างๆ ที่จัดการกับตัวดิจิทัลคอนโทรลเลอร์ เป็นดังนี้



รูปที่ 5-8 *บล็อกไดอะแกรมของซอฟต์แวร์โปรแกรมมอนิเตอร์ของคอนโทรลเลอร์*

จากรูปที่ 5-8 อธิบายได้ว่า อย่างแรก อินพุตของตัวดิจิทัลคอนโทรลเลอร์นั้นแบ่งอินพุต 2 ประเภทด้วยกันคือ

1.แรงดันอินพุตซึ่งป้อนกลับมาจากกระบวนการมีช่วงระหว่าง 0 โวลต์ ถึง 5 โวลต์ โดยที่แรงดัน 0 โวลต์ เทียบเป็น 0 เปอร์เซ็นต์ และ 5 โวลต์เทียบเป็น 100 เปอร์เซ็นต์ของการการควบคุม ซึ่งอินพุตนี้ใช้กับโหมดควบคุมแบบ P, PI และ PID

2.ความถี่อินพุตซึ่งป้อนกลับมาจาก เอนโคเดอร์(encoder) หรือแหล่งความถี่ใดๆ โดยมีช่วงความถี่ระหว่าง 60 เฮิรตซ์ ถึง 1.5 กิโลเฮิรตซ์ โดยจะเทียบเป็น 0 เปอร์เซ็นต์ และ 100 เปอร์เซ็นต์ ตามลำดับ โดยอินพุตแบบนี้ใช้เฉพาะโหมดควบคุมแบบ PLL เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.3 การจัดการทำงานของโปรแกรม

จากรูปเนื่องจากเป็นระบบควบคุมแบบเวลาจริง(real time) ทำให้กระบวนการในประมวลผลของไมโครโปรเซส เซอร์ตามโปรแกรมนั้นต้องเร็วที่สุด ที่ซึ่งการทำงานของไมโครคอนโทรลเลอร์เบอร์ 8031 ใช้สัญญาณนาฬิกา 12 เมกะเฮิรตซ์ จะได้เปรียบกว่าไมโครโปรเซสเซอร์เบอร์ Z80 หรือ 8085 ซึ่งต่างก็เป็น ไมโครโปรเซสเซอร์แบบ 8 บิต ในกระบวนการตั้งแต่อินพุตผ่านรูทีนของโปรแกรมต่างจนกระทั่งได้ เป็นค่า CV (control variable)ไปควบคุมกระบวนการจะใช้เวลาของแต่ละโหมดต่างกันดังนี้

1.โหมด P จะใช้เวลาการคำนวณรวมทั้งหมดประมาณ 17 มิลลิวินาทีโดย โหมดนี้จะผ่านซอฟต์แวร์ โปรแกรมจำนวนคำสั่งน้อยที่สุด

2.โหมด PI ใช้เวลาประมาณ 20 มิลลิวินาที

3.โหมด PID ใช้เวลาประมาณ 21 มิลลิวินาที

4.โหมด PLL ใช้เวลาประมาณ 0.6 มิลลิวินาที เหตุที่น้อยที่สุดเพราะว่าทั้ง 3 โหมดแรกใช้ซอฟต์แวร์การคำนวณทางคณิตศาสตร์ซึ่ง IC 8031 ทำได้ไม่เร็วนัก ด้วยเหตุนี้โหมด PLL ไม่ใช้การคำนวณทางคณิต-ศาสตร์และรูทีนมีคำสั่งน้อย จึงทำให้เวลาน้อยที่สุด

5.3.4 หน้าที่ของรูทีนคีย์บอร์ด

หน้าที่ของซอฟต์แวร์จัดการคีย์บอร์ด หรือเรียกว่าไบออส(bios) สำหรับการจัดการคีย์บอร์ดเพื่อให้ผู้ใช้สามารถติดต่อกับตัวดิจิทัลคอนโทรลเลอร์ได้ โดยใช้เซตการทำหน้าที่ต่างดังนี้

1.เลือกโหมดได้ 4 โหมดดังกล่าวมาแล้ว

2.เลือก setpoint (SP) ได้ตั้งแต่ 0 - 100 เปอร์เซ็นต์

3.เลือกค่า KP (proportional gain) ได้ตั้งแต่ 0 - 200 เท่า

4.เลือกค่า KI (integal time) ได้ตั้งแต่ 0.01 - 99 วินาที

5.เลือกค่า KD (dirivative time) ได้ตั้งแต่ 0.01 - 99 วินาที

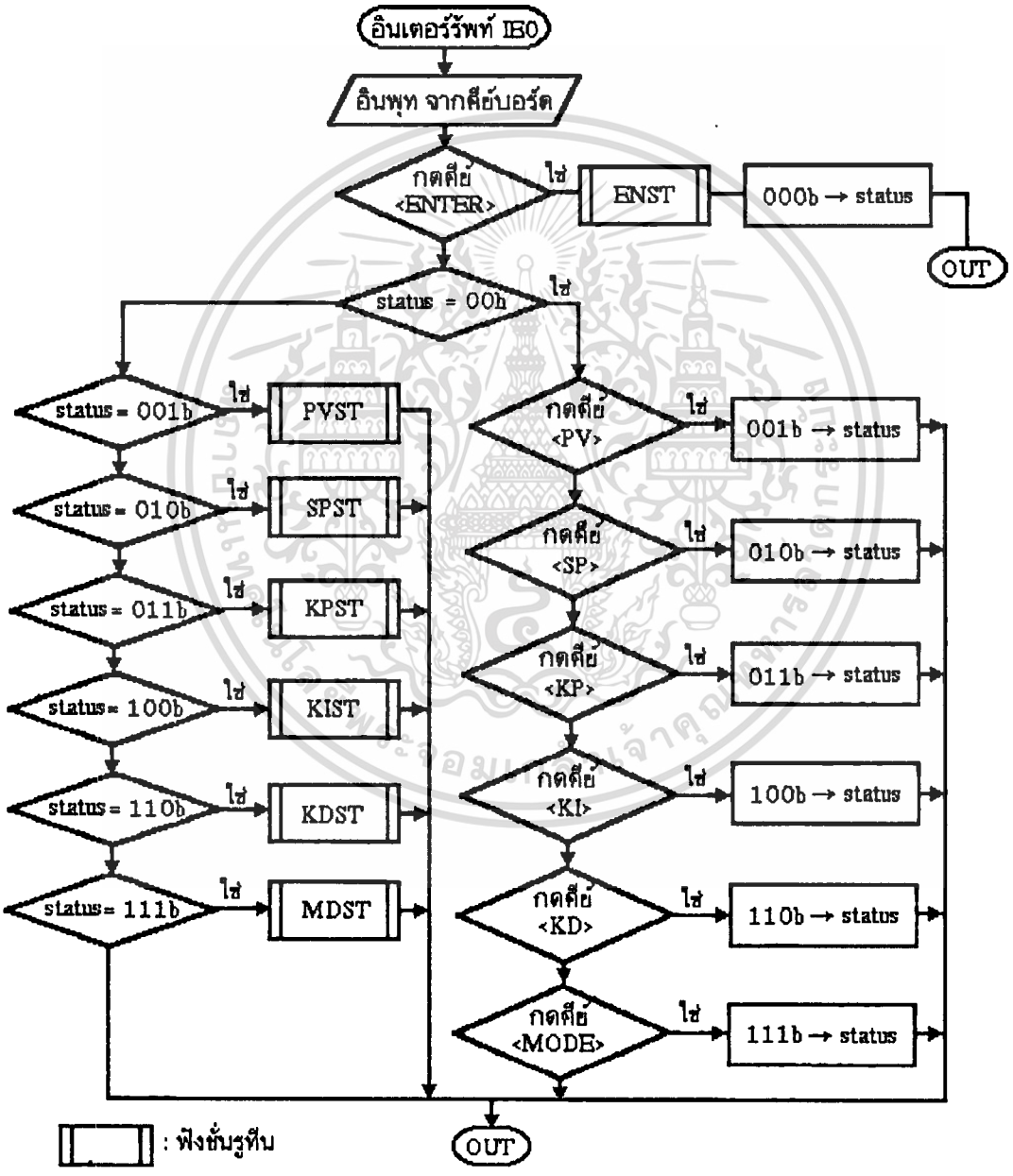
6.แสดงค่า PV (process variable)

เนื่องจากการจัดการทั้ง 6 หัวข้อของคีย์บอร์ดดังกล่าว ต้องไม่ทำให้การทำงานปกติในการควบคุมกระบวนการในโหมดไดก็แล้วแต่ ต้องไม่หยุดการทำงาน เพราะว่าจะเกิดการเสียหายแก่กระบวนการ ดังนั้นซอฟต์แวร์รูทีนสำหรับคีย์บอร์ด จึงต้องใช้วิธีอินเตอร์รัพต์ โดยใช้ IE0 คืออินเตอร์-รัพต์ภายนอกเบอร์ 0 ของขา 12 ของ IC 8031 ซึ่งจะชี้ไปทำอินเตอร์รัพต์รูทีนที่แอดเดรส 0003H อย่างไรก็ตามเพื่อไม่ให้เสียเวลาในรูทีนอินเตอร์รัพต์นี้ เพราะการต้องรอคีย์บอร์ดต่อเนื่อง ดังนั้นทำให้ต้องเก็บสถานะเดิม ของคีย์บอร์ดไว้เพื่อไม่ต้องรอ ซึ่งจะใช้เวลาในอินเตอร์รัพต์รูทีนนี้น้อยมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คีย์บอร์ดหรือแป้นคีย์(keypad)ประกอบด้วยปุ่ม(botton)ใช้งานทั้งหมด 12 คีย์คือ <ENTER>, <MODE>, <UP>, <DOWN>, <SP>, <ADDS> <PV>, <EDIT>, <KP>, <KI>, <KD> และคีย์สำรอง ส่วนวิธีใช้จะนำเสนอต่อไป

โฟล์วชาร์ตการทำงานของคีย์บอร์ดรูทีน มีรูปแบบการทำงานตามโฟล์วชาร์ตไดอะแกรม ตามรูป 5-9



รูปที่ 5-9 บล็อกไดอะแกรมของซอฟต์แวร์รูทีนคีย์บอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีลักษณะอธิบายได้ดังนี้ สังเกตได้ว่าบล็อก IF (เป็นรูปเพชร) แบ่งเรียงเป็นสองแถวทางด้านซ้ายและด้านขวา ซึ่งการจัดบล็อก IF ลงมาในลักษณะจะทำหน้าที่เหมือนคำสั่ง CASE ในภาษาระดับสูง รูทีนหรือซับรูทีนนี้จะทำงาน โดยตัวไมโครคอนโทรลเลอร์ถูกอินเตอร์รัพต์จากวงจรรับคีย์-บอร์ดภายนอก ตัวซอฟต์แวร์จะทำการตรวจสอบว่าเป็น ENTER คีย์หรือแทนเป็น <ENTER> หรือไม่ ถ้า yes จะทำซับรูทีน ENST ต่อไป โดยที่ no โปรแกรมจะตรวจสอบสถานะจากตัวแปร status ซึ่งตัวแปรนี้จะเก็บสถานะคีย์เดิมไว้ เมื่อตรวจสองสถานะแล้วปรากฏว่าเท่ากับ 00h แสดงว่าสถานะคีย์เดิมนั้นไม่มีดังนี้จะทำงานตามบล็อก IF ทางด้านขวาซึ่งมีหน้าที่เปลี่ยนค่าในตัวแปร status ตามสถานะ คีย์ปัจจุบันเช่น <PV>, <SP> เป็นต้น ตรงกัน-ข้าม ถ้า status ≠ 00h จะทำตามซับรูทีนทางด้านขวามือ ซึ่งเป็นรูทีนไปเซตพารามิเตอร์ต่างของการควบคุมของตัวคอนโทรลเลอร์ดังนี้

1.*สับรูทีน PVST* เป็นรูทีนเพื่อให้ดิสเพลย์ ของคอนโทรลเลอร์แสดงค่า PV (process variable)

2.*สับรูทีน SPST* เป็นรูทีนสำหรับเลือกค่า SP หรือเซตพอย์น โดยที่จะแสดงผลเซตพอย์นปัจจุบัน (ถ้ารีเซตค่าเซตพอย์นจะเป็นที่ 50 เปอร์เซ็นต์)

3.*สับรูทีน KPST* เป็นรูทีนสำหรับเลือกค่า KP คือพรอปพรอชันแนลเกน

4.*สับรูทีน KIST* เป็นรูทีนสำหรับเลือกค่า KI คืออินทีกรัลเกนมีหน่วยเป็นวินาที

5.*สับรูทีน KDST* เป็นรูทีนสำหรับเลือกค่า KD คือดีริเวทีฟเกนมีหน่วยเป็นวินาที

6.*สับรูทีน MDST* เป็นรูทีนสำหรับเลือกโหมดซึ่งมีสามโหมด มีรูทีน ดังนี้ P-control PI-control PID-control และ PLL-control

5.3.5 หน้าที่ของรูทีนแสดงผลบนตัวเลข 7 ส่วน

จอแสดงผลของตัวดิจิทัลคอนโทรลเลอร์จะใช้ตัวเลข 7 ส่วนจำนวน 6 หลักด้วยกัน รูทีนต่างๆ ซึ่งเกี่ยวข้องกับการแสดงผลบนตัวเลข 7 ส่วน ซึ่งใช้จำนวนตัวเลข 7 ส่วน 6 หลักนี้คือ สองหลักแรกทางซ้ายมือแสดงตัวแปรต่างๆ เช่น PV SP เป็นต้น หลักที่สามแสดงเคอร์เซอร์ ส่วนสามหลักสุดท้ายแสดงค่าตัวแปรเป็น เปอร์เซ็นต์ หมายเลข วินาที ซึ่งแล้วแต่หน้าที่ของคีย์บอร์ดในขณะนั้น

รูทีนที่เกี่ยวข้องกับตัวเลข 7 ส่วนนั้นจะรวมอยู่ในไฟล์ 7-SEG.ASM มีรูทีนต่างๆดังนี้

1.*สับรูทีน SHOWDISP1* ซึ่งรูทีนนี้จะนำค่าที่อยู่ภายในเรจิสเตอร์ R0 และ R1 แสดงบนตัวเลข 7 ส่วนหลักที่ 1 หลักที่ 2 และ หลักที่ 3 นับจากขวามือ ซึ่งผลที่แสดงบนจอแสดงผลจะเป็นเลขฐาน 16 ไม่เกินค่า FFFh สมมุติว่า จะแสดงผลบนตัวดิจิทัลคอนโทรล-เลอร์เป็นจำนวนตัวเลขฐาน 16 เป็น 57C เราต้องให้ค่าตัวเลขกับเรจิสเตอร์ R0 เป็น 7Ch และ R1 เป็น 05h เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. *สับรูทีน SHOWDISP2* หน้าที่เป็นเช่นเดียวกับ รูทีน SHOWDISP1 แต่จะแสดงผลบนตัวเลข 7 ส่วนหลักที่ 4 หลักที่ 5 และหลักที่ 6 จากขวามือ
3. *สับรูทีน SHOWDISP3* หน้าที่เป็นเช่นเดียวกับ รูทีน SHOWDISP2 แต่ต่างกันตรงที่การแสดงผลนั้น จะเป็นช่องว่าง(blank) ถ้าหลักแรกของตัวเลขเป็นศูนย์ ตัวอย่างเช่นถ้าค่าเป็น 0C5h ถ้าเป็นรูทีน SHOWDISP1 หรือ SHOWDISP2 จะแสดงผลบนจอแสดงผลเป็น 0C5h แล้วซึ่งหน้าที่ของรูทีน SHOWDISP3 จะแสดงค่าเป็น 5C สังเกตว่าข้างหน้าตัวเลข 5 จะไม่มีเลข 0 โดยให้เป็นช่องว่างไป
4. *สับรูทีน HEXtoPER* หน้าที่ของรูทีนคือแปลงตัวเลขฐาน 16 เป็นค่าเปอร์เซ็นต์ โดยเทียบจาก 00h เป็น 0 เปอร์เซ็นต์ และ FFh เป็น 100 เปอร์เซ็นต์ ซึ่งวิธีการของรูทีนนี้จะใช้การเปิดตาราง โดยตารางที่ใช้คือ TABLEHD0, TABLEHD1, TABLEHD2, TABLEHD0, TABLEHP1, TABLEHP2 มีตารางจำนวนทั้งสิ้น 6 ตารางด้วยกัน ตัวแปรอินพุตของรูทีนนี้เป็น เรจิสเตอร์ R0 และ R1 โดยที่ R0 แทนบิตนัยสำคัญต่ำและ R1 แทนบิตนัยสำคัญสูง เมื่อเข้าสู่กระบวนการในรูทีน HEXtoPER แล้วจะให้เอาต์พุตเป็นค่าของเปอร์เซ็นต์ที่ R0 และ R1 ถึงแม้จะสิ้นเปลืองเนื้อที่มอนิเตอร์โปรแกรม ถ้าเทียบกับความเร็วในการแสดงผลการแสดงผลถือว่าคุ้มค่า
5. *สับรูทีน INITDISP* มีหน้าที่เซตค่าเริ่มต้นของฮาร์ดแวร์ ที่ซึ่งสนับสนุนการแสดงผล



รูปที่ 5-10 *แสดงโฟล์วชาร์ตของการแสดงผลเป็นค่าเปอร์เซ็นต์*

ทั้งหมดทั้ง 5 รูทีนนั้นไม่มีความสำคัญอะไร ที่ต้องแสดงรายละเอียดของโฟล์วชาร์ตของแต่ละรูทีน เพราะว่าไม่เกี่ยวข้องกับคุณสมบัติความสามารถของตัวดิจิทัลคอนโทรลเลอร์แต่อย่างใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่เป็นเพียงรูทีนสนับสนุนเท่านั้น อย่างไรก็ดีถ้าต้องการรายละเอียดแต่ละรูทีนสามารถดูได้จากไฟล์ "7-SEG.ASM" ได้ ซึ่งโฟล์วชาร์ตการแสดงผลบนจอแสดงผลของดิจิทัลคอนโทรลเลอร์ แสดงดังรูปที่ 5-10

5.3.6 หน้าที่ของรูทีนเซตพารามิเตอร์ของตัวดิจิทัลคอนโทรลเลอร์

พารามิเตอร์ต่างของตัวดิจิทัลคอนโทรลเลอร์เช่น เซตพอย์น(SP) ,พรอปพรอชันแนลเกน(KP), อินทีกรัลเกน(KI), ดีริเวทีฟเกน(KD) เป็นต้น จำเป็นต้องเปลี่ยนแปลงตลอดเวลาเพื่อให้เหมาะสมกับกระบวนการในขณะนั้น การเซตนั้นจะทำโดยคีย์บอร์ด ที่ซึ่งการทำงานของคีย์บอร์ดนั้นได้อธิบายไว้ข้างต้นอย่างชัดเจน ซึ่งในหัวข้อนี้จะอธิบายเจาะจงเกี่ยวกับการเปลี่ยนค่าพารามิเตอร์เหล่านั้น รูทีนที่เกี่ยวข้องกับหัวข้อนี้จะอยู่ในซอร์สโค้ดแอสเซมบลีในไฟล์ชื่อว่า " ALLKTabl.ASM " และไฟล์ "INTKEY2.ASM" โดยที่ไฟล์แรกเกี่ยวข้องกับการแสดงผล และไฟล์สองเกี่ยวกับการนำสถานะคีย์บอร์ดมากระทำ

ในไฟล์ ALLKTable.ASM ดังกล่าวประกอบด้วยสับรูทีนต่างๆที่สำคัญดังนี้

1. *สับรูทีน LookATKtable* มีหน้าที่รับค่าจากตัวแปรเกนเช่น KP0,KP1 ซึ่งเป็นพรอปพรอชันแนลเกนใช้จำนวนบิต 16 บิตโดยบิตนัยสำคัญต่ำจะอยู่ใน KP0 และบิตนัยสำคัญสูงจะอยู่ใน KP1 ที่ซึ่งจำเป็นต้องใช้ถึง 16 บิตเพราะว่าต้องใช้เป็นทศนิยม 4 บิต เช่นเดียวกับ KI0, KI1 ของอินทีกรัลเกน และ KD0,KD1 ของดีริเวทีฟเกน

สับรูทีนนี้มีอินพุตตัวแปรเป็นเรจิสเตอร์ R0,R1 โดยข้อมูลของเรจิสเตอร์แทนจาก KP0,KP1 หรือ KI0,KI1 หรือ KD0,KD1 ซึ่งรูทีนนี้จะโดยเอาข้อมูลจากเรจิสเตอร์ไปเปิดตาราง โดยข้อมูลที่ได้จากการเปิดตารางจะเป็นข้อมูลใหม่ ซึ่งพร้อมจะนำไปแสดงผลบนจอแสดงผลของตัวคอนโทรลเลอร์ โดยข้อมูลใหม่จะเข้าสู่รูทีน ShowKTab ก่อนจะแสดงผล

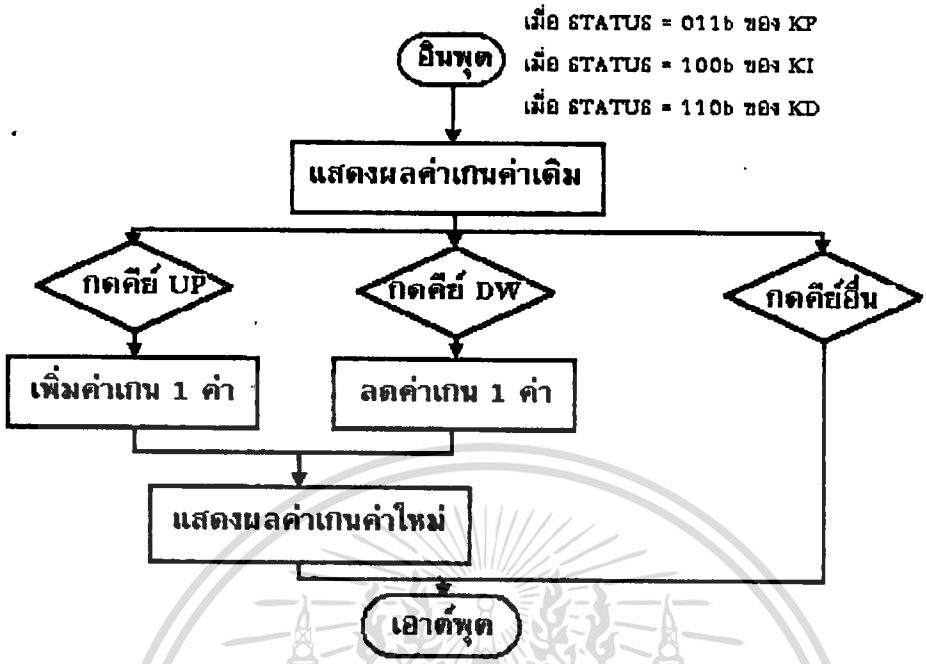
2. *สับรูทีน ShowKTab* มีหน้าที่เข้าข้อมูลใหม่ดังกล่าวอยู่ในเรจิสเตอร์ R3 ,R4 นำไปแสดงผลบนจอแสดงผลของตัวคอนโทรลเลอร์เป็น 3 หลักด้วยกัน ซึ่งค่าที่ได้จากการเปิดตารางเมื่อนำไปแสดงผลจะได้ค่าต่ำสุดเป็น 0.001 และสูงสุดเป็น 999 โดยสามารถครอบคลุมได้ทั่วถึงค่าปกติซึ่งใช้อยู่ในทั่วไปของคอนโทรลเลอร์ และมีตาราง LOOK-UP คือ tableK0 ,tableK1 ,tablePK0 ,tablePK1

ในไฟล์ INTKEY2.ASM ยกมาอธิบายเฉพาะที่เกี่ยวกับการเซตค่าเกนดังนี้

1. *สับรูทีน KPst* เกี่ยวข้องกับการเปลี่ยนค่าพรอปพรอชันแนลเกน
2. *สับรูทีน KIst* เกี่ยวข้องกับการเปลี่ยนค่าอินทีกรัลเกน
3. *สับรูทีน KDst* เกี่ยวข้องกับการเปลี่ยนค่าดีริเวทีฟเกน

โดยทั้ง 3 รูทีนมีลักษณะโฟล์วชาร์ตคล้ายคลึงกัน ดังนี้ ตามรูปที่ 5-11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-11 แสดงโฟล์วชาร์ตของการเปลี่ยนค่าเกน

5.3.7 หน้าที่ของรูทีนเลือกโหมดการสั่งการจากคีย์บอร์ด

คล้ายคลึงกันกับวิธีการเซตค่าเกนของโหมดควบคุม ที่ซึ่งรูทีนเกี่ยวกับการเลือกโหมดนั่นค่า รูทีน MDst อยู่ในไฟล์ชื่อว่า " INTKEY2.ASM " ดูในซอร์สโค๊ดแอสเซมบลีของคอนโทรลเลอร์ ซึ่งรูทีน MDst จะทำงานก็ต่อเมื่อตัวแปร STATUS มีค่าเป็น 111b เมื่อรูทีน MDst ทำงานจะแสดงค่าโหมดการควบคุมที่ใช้อยู่ปัจจุบันโดยอ่านค่าจากตัวแปร MDcount ซึ่งถ้า MDcount เท่ากับ 0 จะเป็นโหมด P ถ้าเท่ากับ 1 จะเป็นโหมด PI ถ้าเป็น 2 จะเป็นโหมด PID ถ้าเป็น 3 จะเป็นโหมด PLL โดยที่การเปลี่ยนโหมดกดคีย์ UP หรือคีย์ DW

5.3.8 หน้าที่ของรูทีนโหมดการควบคุมแบบ P

โหมดการควบคุมต่างมีรูทีนอยู่ในไฟล์ชื่อว่า "USEPLL.ASM" โดยการควบคุมแบบพรอป-พรอชันแนลจะมีรูทีนใช้เฉพาะคือ P_control ดังมีรายละเอียดดังนี้

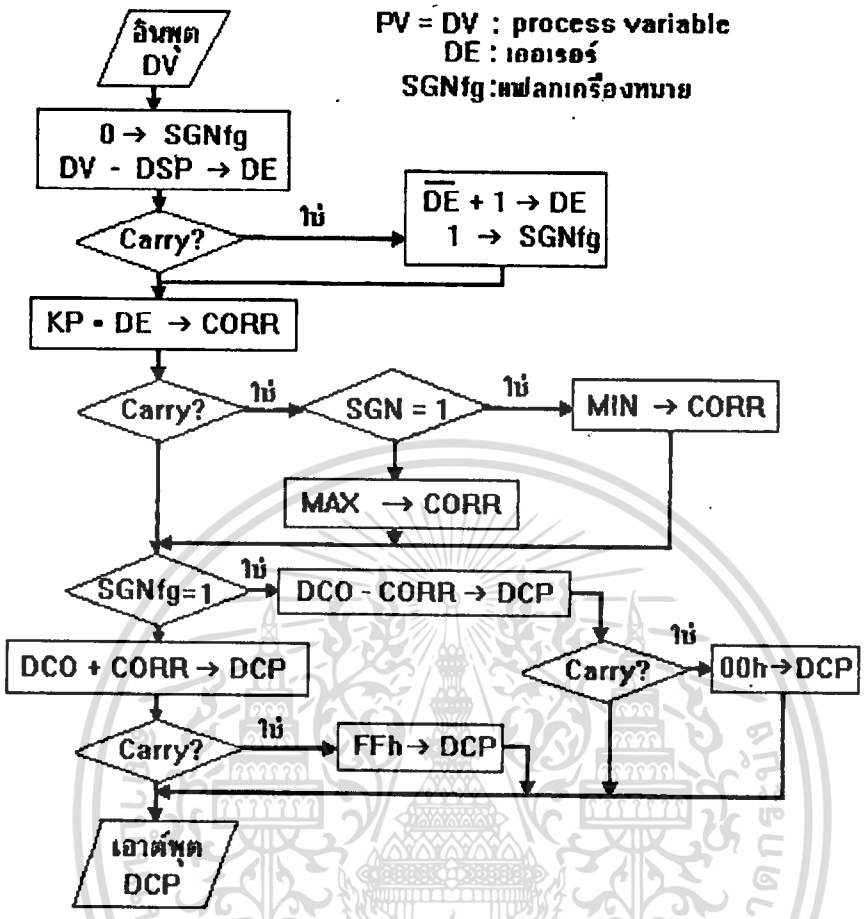
```
P_contl:                          ;ตัวอย่างสับรูทีนของ การควบคุมแบบพรอปพรอชันแนล
;-------------------
;DE:control variable error        ;กำหนดตัวแปรเออเรอร์เป็น DE
;CORR  : R3                       ;ใช้เรจิสเตอร์ R3 แทน ตัวแปร CORR ซึ่งเป็นตัวแปรชั่วคราว
;-------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CLR     SGNFg           ;เคลียร์แฟลกเครื่องหมาย ซึ่งแสดงโพลาริตี้ของ DE
        CLR     C               ;เคลียร์แฟลกทด
        MOV     A,ADBUFF        ;อ่านค่า PV จาก A/D
        MOV     R4,A            ;โอนค่า PV ให้เรจิสเตอร์ R4
        SUBB    A,DSP           ; PV - DSP -> A  :ซึ่ง DSP คือค่าเซตพอย์น
        JNC     NoC_P           ;error is carry as sign = 1
        SETB    SGNFg           ;Sgn 0 =Plus,Sgn 1=Minus
        CPL     A               ;คอมพลีเมนต์ A
        ADDC    A,#00H          ;นำค่าทดมาบวก
NoC_P:  MOV     DE,A            ;โอนค่า A ให้ DE
        MOV     R5,KP0          ;นำค่าพรอปพรอชันแนลเกนบิตนัยสำคัญต่ำ KP0 ให้กับ R5
        MOV     R6,KP1          ;นำค่าพรอปพรอชันแนลเกนบิตนัยสำคัญสูง KP1 ให้กับ R6
        MOV     A,DE            ;โอน DE -> A
        LCALL   MULT16C         ;KP*DE -> CORR หรือ R6R5*A -> R6R5 โดยเรียกใช้รูทีนคูณ
        MOV     R7,#N0_digit    ;กำหนดจำนวนหลักทศนิยมให้ R7
        ACALL   SHIFTDIG        ;ปัดทศนิยม
        ACALL   OverLimit1      ;เช็คโอเวอร์โฟล์ว
        MOV     R3,R0           ;โอนค่าที่ตรวจสอบโอเวอร์โฟล์วแล้วให้ R3 หรือ CORR
OutLimitK:  JNB SGNFg, minusP0  ;ถ้าแฟลกเครื่องหมายไม่เท่ากับ 1 กระโดดไปที่แอรดเดรสชื่อ minusp0
        CLR     C               ;
        MOV     A,DC0           ;load Zero Error DC0+CORR->DCP
        ADD     A,R3            ;
        MOV     R3,A
        JNC     NoaddP0         ;ถ้า DCP > FFh แล้ว DCP = FFh
maxP:   MOV     R3,#0FFH
        SJMP    NoAddP0
minusP0:CLR     C               ;SGNFg = 0 work minus
        MOV     A,DC0
        SUBB    A,R3            ;DC0-CORR->DCP
        MOV     R3,A
        JNC     NoAddP0         ;ถ้า DCP < 00h แล้ว DCP = 00h
minP:   MOV     R3,#00H
NoaddP0:MOV     DCP,R3          ;ได้ค่าคอนโทรลเอาต์พุต DCP
        MOV     DC0,R4          ;ได้ค่าเออเรอร์ศูนย์
        RET
```

ซึ่งได้นำสัปรูทีนเฉพาะเกี่ยวกับการควบคุมแบบพรอปพรอชันแนลมาแสดง โดยที่ซอฟต์แวร์แอสเซมบลี้ทั้งหมดแสดงอยู่ในรายการซอฟต์แวร์ อย่างไรก็ดีสัปรูทีนข้างต้นนั้นชี้ให้เห็นถึงอัลกอริธึมของมันซึ่งคล้องจองกับโฟล์วชาร์ตดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-12 *แสดงโฟล์วชาร์ตซึ่งสอดคล้องกับรูทีนแอสเซมบลีในการควบคุมแบบพรอปพรอชันแนล ซึ่งรายละเอียดของโฟล์วชาร์ตนี้อธิบายไว้ในบทที่ 3*

5.3.9 หน้าที่ของรูทีนโหมดการควบคุมแบบ PI

ซอฟต์แวร์สัปรูทีนของการควบคุมแบบ PI คือรูทีนชื่อว่า " PI_control " อยู่ในไฟล์ USEPLL.ASM รูทีนจะประกอบด้วยอัลกอลิธึมของการควบคุมแบบ P และการควบคุมแบบ I นำมาคาสเคด(cascade)กัน โดยจะนำเอาอัลกอลิธึมของการควบคุมแบบ I มาแสดงดังนี้

```
I_contl:                                ; การควบคุมแบบอินทีกรัล
        CLR     SGNFg                   ; ให้แฟลกเครื่องหมายเป็นศูนย์
        CLR     C                       ;
        MOV     A,ADBUFF        ;รับค่า PV จาก A/D
        SUBB    A,DSP                   ; PV - DSP -> DE
        JNC     NoC_I           ;error is carry sign 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        SETB    SGNFg           ;ถ้าเป็น PV < DSP ให้แฟลกเครื่องหมายเป็น  1
        CPL     A               ;complement
        ADDC    A,#00H
NoC_I:  MOV     DE,A            ; DE เป็น 2's คอมพลีเมนต์
        JNB     SGNIFg,  SlopDw ;แฟลกเครื่องหมายอินทีกรัลเป็นศูนย์  กระโดดไปที่ SlopDw
        JNB     SGNFg ,minusUp
        ACALL   ADDSUMUP        ;เรียกรูทีน ADDSUMUp เป็นการเพิ่มค่าตัวแปร SUM1,SUM0 ซึ่งเป็นค่าใน
อดีต
        SJMP    OutSlop         ;เสร็จแล้วก็ออก
minusUp: ACALL  SubSumC         ;เรียกรูทีน SubSumC เพื่อลดค่าตัวแปร SUM1,SUM0
        SJMP    outSlop
SlopDw: JB      SGNFg , minusDw
        ACALL   ADDSUMUP
        SJMP    outSlop
minusDw: ACALL  SubSumC         ;เรียกรูทีน SubSumC เพื่อลดค่าตัวแปร SUM1,SUM0
outSlop: NOP
        MOV     R5,SUM0         ;multiplicand หรือ กำหนดตัวตั้งคูณ 1
        MOV     R6,SUM1         ;multiplicand หรือกำหนดตัวตั้งคูณ  2
        MOV     A, KI0          ;multiplier  หรือ กำหนดตัวคูณ 1
        MOV     R4,KI1          ;multiplier หรือ กำหนดตัวคูณ 2
        LCALL   MULT16_16C      ;KI*SUM -> R2,R1,R0
        MOV     R7,#N0_Digit    ;กำหนดทศนิยมเท่ากับ #NO_Digit
        ACALL   ShiftDig        ;เลื่อนทศนิยมตามกำหนด
        ACALL   OverLimit2      ;จำกัดโอเวอร์โฟล์วที่ลิมิตบนเท่ากับ FFh
        MOV     R5,R0           ;ตัวตั้งคูณ1
        MOV     R6,R1           ;ตัวตั้งคูณ2
        MOV     A,DT            ;ตัวคูณ
        LCALL   MULT16C         ;KI*SUM*DT
        MOV     R7,#N0_DtDig
        ACALL   ShiftDig
        ACALL   OverLimit1
        MOV     DCI ,R0         ;ได้เอาต์พุตอินทีกรัลคอนโทรล  DCI
        RET
```

ที่ซึ่งโฟล์วชาร์ตของอัลกอลิธึมของการควบคุมแบบ I แสดงอยู่ในบทที่ 3 อย่างไรก็ดี ซอร์สโค๊ดที่แสดงในกรอบนั้นเป็นเพียงการควบคุมแบบ I ดังนั้นการควบคุมแบบ PI จึงต้องนำอัลกอ-ลิธึมของการควบคุมแบบ P ,มาคาสเคดดังกล่าว ซึ่งผลที่ได้จะอยู่ในรูทีน PI_Control ซึ่งดูได้ในราย-การของซอฟต์แวร์ ส่วนของโฟล์วชาร์ตของการควบคุมแบบ PI ดูได้ในบทที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.10 หน้าที่ของรูทีนโหมดการควบคุมแบบ PID

การควบคุมแบบต่อเนื่องที่รู้จักกันดีคือการควบคุมแบบ PID ซึ่งสัปรูทีนมีชื่อว่า " PID_control" อยู่ในไฟล์ USEPLL.ASM โดยที่อัลกอลิธึมการควบคุมแบบนี้นำเอาอัล์กอลิธึมของการควบคุมแบบ P อัลกอลิธึมการควบคุมแบบ I และ อัลกอลิธึมของการควบคุมแบบ D มาต่อกับแบบคาสเคด(cascade) ซึ่งในที่นี้จะยกตัวอย่างซอฟต์แวร์อัลกอลิธึมของการควบคุมแบบ D ดังนี้

```
D_Contl:                        ;สัปรูทีนการควบคุมแบบ D
;----------
; R3 : DDE                      ;DDE = DE - DE0 เป็นเออเรอร์ปัจจุบันลบด้วยเออเรอร์อดีต
; R4 : adbuff                   ;แทนค่า PV จาก A/D ด้วยเรจิสเตอร์ R4
;----------
        CLR     C
        CLR     SGNFg           ;เคลียร์แฟลกเครื่องหมาย
        MOV     R4,ADBUFF       ;ให้ค่า PV ได้จาก A/D กับเรจิสเตอร์ R4
        MOV     A,R4
        SUBB    A,DSP           ;DV-DSP->DE
        MOV     DE,R3
        JNC     NoC_D
        SETB    SGNFg           ;Sgn 0 =Plus,Sgn 1=Minus
        CPL     A               ;complement
        ADDC    A,#00H
        MOV     DE,R3           ;  DE  เป็นค่า 2's คอมพลีเมนต์  โดยมี SGNfg เป็นแฟลกเครื่องหมาย
NoC_D:  CLR     C
        CLR     SGNDFg          ;เคลียร์แฟลกเครื่องหมายดีริเวทีฟ
        MOV     A,R4
        SUBB    A,DV0           ;DE - DE0 -> DDE  หรือ DE - DV0 -> DDE
        MOV     R3,A            ;DDE-> R3
        JNC     NoCyD
        SETB    SGNDFg          ;ถ้า DE < DV0 ให้แฟลก SGNDFg เป็น 1
        MOV     A,R3
        CPL     A
        ADDC    A,#00H
        MOV     R3,A            ;ทำ DDE เป็น 2's คอมพลีเมนต์
NoCyD:  MOV     DV0,R4          ;DE -> DV0
        MOV     R5,KD0          ;ให้ KD0 เป็นตัวตั้งคูณ1
        MOV     R6,KD1          ;ให้ KD1 เป็นตัวตั้งคูณ2
        MOV     A,R3            ;KD*DDE -> R2,R1,R0
        LCALL   MULT16C         ;ผลลัพท์ใน  R2,R1,R0
        ACALL   OVERLIMIT2      ;จำกัดโอเวอร์โฟล์ว
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
OutChkD:  NOP
          CJNE    R0,#00H,toDt      ;ถ้าไม่เท่ากับศูนย์กระโดด
          CJNE    R1,#00H,toDt      ;
          CJNE    R2,#00H,toDt      ;
          SJMP    nonDt
toDt:     MOV     R7,DT             ;KD*DDE/DT
          LCALL   DIV16             ;product R1,R0
nonDt     CLR     C
          MOV     A,#N0_Digit       ;กำหนดทศนิยมปกติ
          SUBB    A,#N0_DTDig       ;ลบด้วยทศนิยมของแซมปลิง
          MOV     R7,A              ;
          ACALL   ShiftDIG          ;เลื่อนทศนิยมตามจริง
          LCALL   OverLimit1        ;ตรวจสอบโอเวอร์โฟล์ว
OutMore:  MOV     DCD,R0            ;เอาต์พุตคอนโทรล DCD
          RET
```

ส่วนโฟล์วชาร์ตของการควบคุมแบบ D และโฟล์วชาร์ตการควบคุมแบบ PID ดูได้ในบทที่ 3

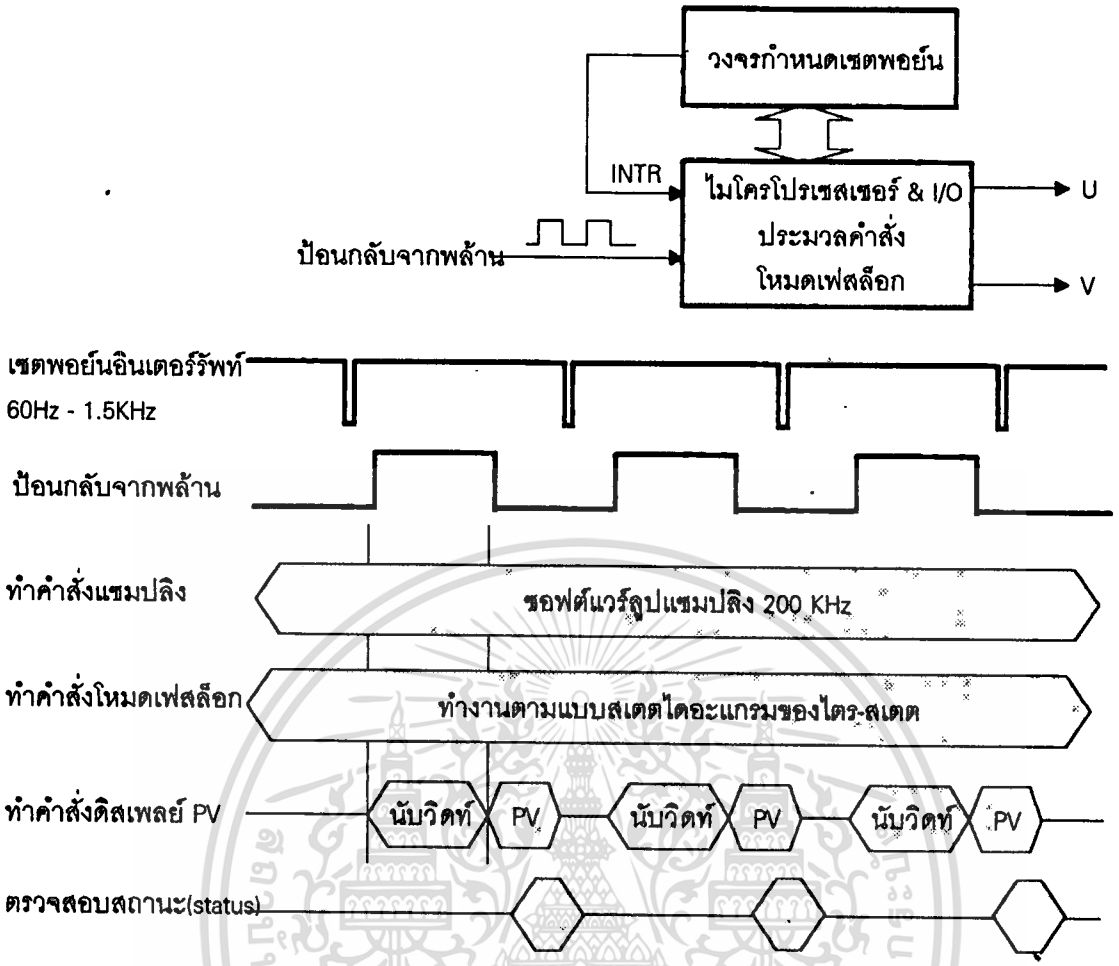
5.3.11 หน้าที่ของรูทีนโหมดการควบคุมแบบ PLL

ซอฟต์แวร์สับรูทีนของ PLL นั้นจะเลียนแบบการทำงานเฟสดีเทกเตอร์แบบ 4 หรือที่เรียกว่า เฟส/ความถี่ดีเทกเตอร์ รายละเอียดและคุณสมบัติของเฟสดีเทกเตอร์แบบนี้ดูได้ในบทที่ 2 และบทที่ 3 ลักษณะที่น่าสนใจของเฟสดีเทกเตอร์แบบนี้ คือเอาต์พุตจะมีลักษณะเป็นไตร-สเตต(tri-state) ซึ่งลักษณะนี้เองสามารถทำให้มันมีความไว(sensitive)กับการตรวจจับความถี่ และด้วยเหตุนี้เองทำให้มันมีช่วงล็อก(lock range)กว้างมาก จึงเป็นที่นิยมอย่างกว้างขวางกับการประยุกต์ใช้

อย่างไรก็ดีการใช้ซอฟต์แวร์ในการแทนเฟสดีเทกเตอร์นั้น ปัจจุบันเป็นเพียงโมเดลทางคณิต-ศาสตร์ เช่น ใช้การคูณแทนเฟสดีเทกเตอร์แบบ 1 ซึ่งเป็นแอนะล็อกเฟสล็อกลูป หรือใช้การแปลงของฮิลเบอร์ทดูได้ในบทที่ 2) เป็นต้น

ดังนี้จึงได้นำเสนอการแทนหรือการเลียนแบบเฟสดีเทกเตอร์แบบ 4 โดยใช้ซอฟต์แวร์ ซึ่งใช้งานร่วมกับอุปกรณ์ทางฮาร์ดแวร์ในบทที่ 4 ตามรูปที่ 4-14 โดยที่รูปที่ 5-13 แทนไดอะแกรมการทำงานของรูทีนซอฟต์แวร์ในโหมดเฟสล็อก แบบไตร-สเตตของเฟสดีเทกเตอร์แบบ 4 หรือเฟส/ความถี่ดี-เทกเตอร์ รายละเอียดของลักษณะต่างของสเตตเป็นอย่างไรแสดงอยู่ในบทที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-13 *ไดอะแกรมเวลาของการทำงานในโหมดเฟสล็อก*

```
PLL_V5:                          ;สับรูทีนเฟสดีเทกเตอร์ของระบบ PLL เวอร์ชั่นที่ 5
;-------------------
; subroutine PLL_V5
;-------------------
        PUSH    R3
        CLR     DutyFg    ;เป็นแฟลกสำหรับดิวตี้ไซเคิล  ถ้าแฟลกเป็น 1 แสดงว่าดิวตี้ไซเคิลของสัญญาณ เป็น 50 เปอร์เซ็น
ต์
BeginV5: ACALL  ChkF_sig          ;while(LOW) หรือกระบวนการในลักษณะที่สัญญาณอินพุตป้อนกลับเป็น low
        JB      Intr2ACTFg,IntrBefore ;ถ้ามีอินเตอร์รัพต์ EX1 จากวงจรเกิดขึ้น
        ;--- Check Duty ------ .      ;ระหว่างตรวจสอบดิวตี้ไซเคิล
        JB      HighFg,Chk1Duty          ;ถ้าสัญญาณอินพุตป้อนกลับเป็น 1 กระโดดไปที่ Chk1Duty
        SETB    DutyFg                   ;ให้แฟลกสำหรับดิวตี้ไซเคิลเป็น HIGH
        SJMP    BeginV5                  ;กระโดดไปที่ตำแหน่งเริ่มต้นรูทีน
Chk1Duty: NOP
        ;------ end duty ----
        ACALL   TurnOff           ;เอาต์พุตทุกสภานะ OFF ซึ่งจะกลายเป็นสภาวะอิมพลีแดนซ์สูง
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ACALL   TurnOnDw        ;.ให้เอาต์พุตเป็นสเตตที่ 1 ดูตามรูป 4-18
        MOV     R3,#00H         ;init count รีเซตตัวนับ
        CLR     C               ;เคลียร์แฟลกทด
        CLR     A               ;เคลียร์เรจิสเตอร์ A
Wait1Intr: ACALL  ChkF_sig      ;ตรวจสอบสัญญาณอินพุตป้อนกลับว่าเป็นลอจิก HIGH หรือ LOW
        JB      Intr2ACTFg ,    Outwait1 ;ถ้าเกิดอินเตอร์รัพต์ EX1 ให้กระโดดไปที่ Outwait1
        ; -- reserve for counter --
        MOV     A,R3            ;ให้เรจิสเตอร์ R3 เป็นบัฟเฟอร์ของเคาน์เตอร์
        ADD     A,#01H          ;เพิ่มค่าที่นับได้ทีละหนึ่ง
        MOV     R3,A            ;
        JNC     NocryV5         ;ถ้าไม่ค่านับไม่เกิน 0FFh กระโดดไปที่ตำแหน่ง NocryV5
        MOV     R3,#0FFH        ;ถ้านับเกินให้เป็น 0FFh
NocryV5: JB     Highfg,HighV5   ;ถ้าสัญญาณอินพุตป้อนกลับเป็นลอจิก HIGH กระโดดไปที่ตำแหน่ง HighV5
        MOV     A,R3            ;ต่อไปนี้เป็นการแปลงดิวตี้ไซเคิลเป็นความถี่ โดยใช้วิธีคอมพลีเมนต์
        CPL     A               ;convert DUTY to FREQ
        JNB     DutyFg,HighV5   ;ถ้าดิวตี้ไซเคิลไม่เท่ากับ 50 เปอร์เซ็นต์ แล้วกระโดด
        MOV     ADBuff,A        ;ส่งค่าความถี่ไปแสดงผลบนจอโดยผ่านตัวแปร ADBuff
HighV5: NOP                     ;No operation
        ; ------ end count ------
        SJMP    Wait1Intr       ;กระโดดไปที่ Wait1Intr เพื่อคอยอินเตอร์รัพต์
Outwait1: CLR   Intr2ACTFg      ;setb then clear
        ACALL   TurnOff         ;เอาต์พุต PFD เป็นสเตตที่ 2 หรือเป็นสภาวะอิมพลีแดนซ์สูง
Wait1low: ACALL ChkF_sig        ;while(HIGH) คือ กระบวนการในขณะที่อินพุตป้อนกลับเป็น HIGH
        JB      Intr2ACTFg,IntrBefore ;ถ้าเกิดอินเตอร์รัพต์ EX1 กระโดดไปที่ IntrBefore
        JB      HighFg,Wait1low ;ถ้าอินพุตป้อนกลับเป็น HIGH กระโดดที่ Wait1low (คอยลอจิกต่ำ)
        SJMP    EndV5           ;กระโดดไปที่จบ

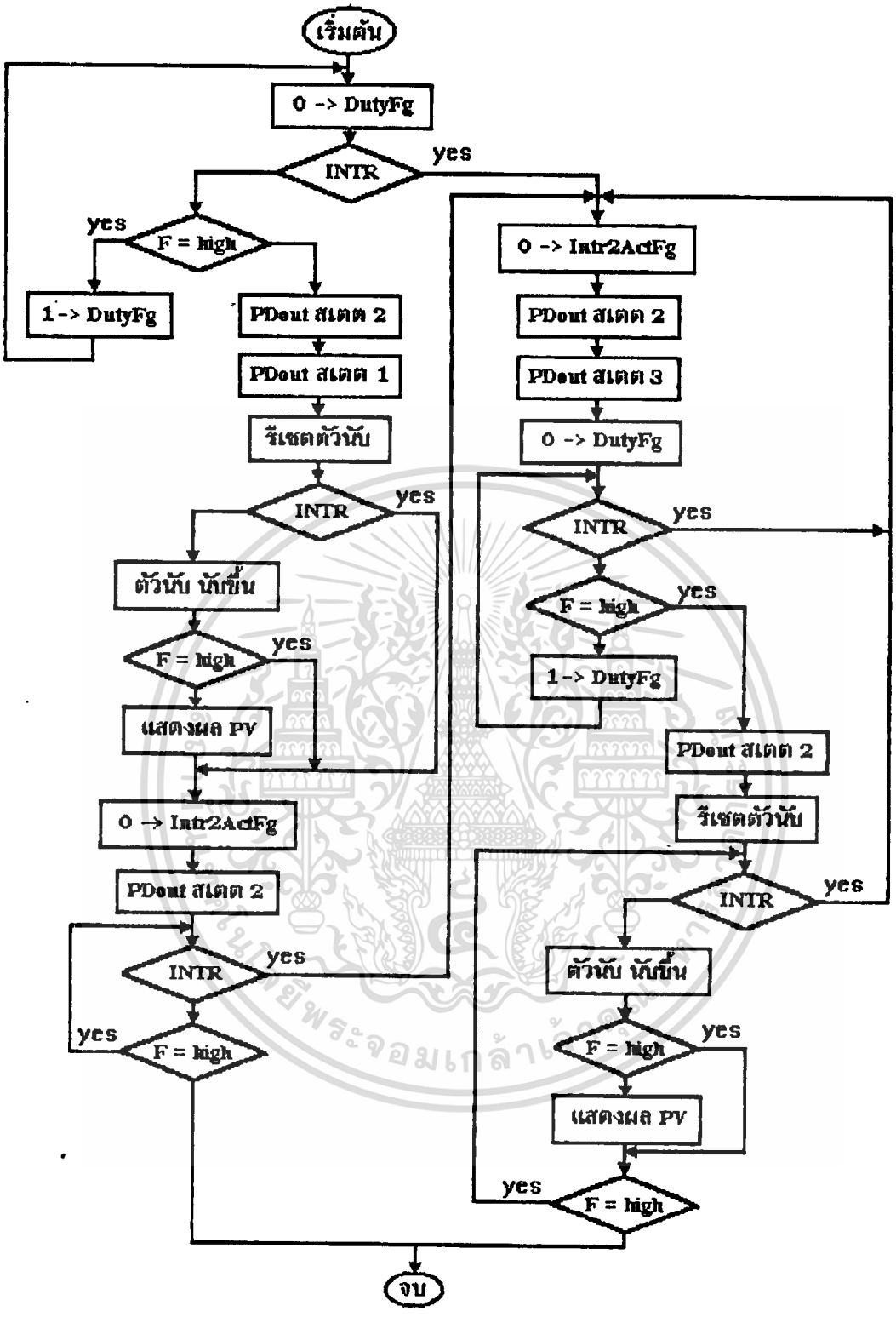
        ;------ if INTR signal occur before feedback signal ------
IntrBefore: CLR Intr2ACTFg      ;เคลียร์อินเตอร์รัพต์แฟลก
        ACALL   TurnOff         ;ให้เอาต์พุต PFD เป็นสเตต 2 หรือเป็นสภาวะอิมพีแดนซ์สูง
        ACALL   TurnOnUp        ;ให้เอาต์พุต PFD เป็นสเตต 3 หรือโวลต์เท่ากับ VCC
        CLR     DutyFg
WhileLow1: ACALL ChkF_sig       ;while(LOW) กระบวนการในขณะที่อินพุตป้อนกลับเป็นลอจิกต่ำ
        JB      Intr2ACTFg,IntrBefore ;ถ้าเกิดอินเตอร์รัพต์ EX1(คล้องจองกับฮาร์ดแวร์) กระโดดไปที่ IntrBefore
        ; ------ duty check —
        JB      HighFg,Chk2Duty ;ถ้าสัญญาณอินพุตป้อนกลับเป็นลอจิกสูง
        SETB    DutyFg          ;ตรวจสอบดิวตี้ไซเคิลจากดิวตี้แฟลก
        SJMP    WhileLow1       ;กระโดด นะจ๊ะ
Chk2Duty: NOP
        ;------ end duty chk ------
        ACALL   TurnOff         ;ให้เอาต์พุต PFD เป็นสเตต 2 หรือเป็นสภาวะอิมพีแดนซ์สูง
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     R3,#00H             ;รีเซตตัวนับ
        CLR     C
        CLR     A
Wait2low: ACALL   ChkF_sig          ;while(HIGH) กระบวนการในขณะที่อินพุตป้อนกลับเป็นลอจิกสูง
        JB      Intr2ACTFg,IntrBefore
        ;---- reserve for count ----------
        MOV     A,R3                ;นับขึ้นทีละหนึ่ง
        ADD     A,#01H
        MOV     R3,A
        JNC     Nocry2V5
        MOV     R3,#0FFH
Nocry2V5: JB    Highfg,High2V5
        MOV     A,R3
        CPL     A                           ;convert DUTY to FREQ
        JNB     DutyFg,High2V5
        MOV     ADBuff,A                    ;show PV
High2V5:  NOP
        ;---------- end count ----------
        JB      HighFg,Wait2low
EndV5:  POP     R3
        RET
```

ตามรูปที่ 5-14 เป็นโฟล์วชาร์ตซึ่งแทนรูทีนของเฟสดีเทกเตอร์ของระบบ PLL ซึ่งเขียนไว้ในตอนต้นเพื่ออธิบายการทำงานของการเลียนแบบเฟสดีเทกเตอร์แบบ 4 ด้วยซอฟต์แวร์ ซึ่งต้องใช้ร่วมกับวงจรซึ่งออกแบบไว้โดยใช้ ซีพียู 8031 เป็นตัวประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-14 โฟล์วชาร์ตของเฟสดีเทกเตอร์แบบไตร-สเตต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 วิธีการใช้งานตัวดิจิทัลคอนโทรลเลอร์

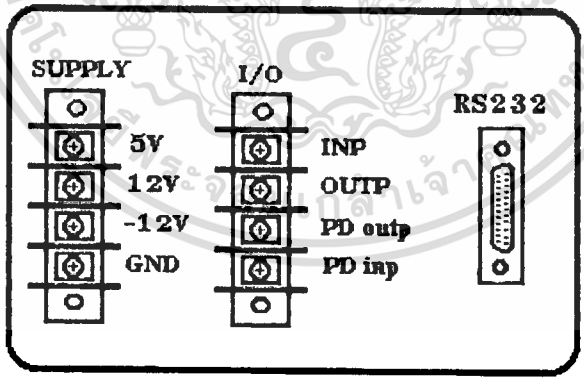
เริ่มแรกคือต่อแหล่งจ่ายไฟเข้าให้เรียบร้อยตามลำดับได้แก่ +5. โวลต์ +12 โวลต์ -12โวลต์ และกราวด์ให้เรียบร้อย และนำไปควบคุมลูปของกระบวนการโดยนำตัวดิจิทัลคอนโทรลเลอร์ไปต่อให้ครบลูป ซึ่งจะใช้แบบการควบคุมแบบต่อเนื่อง หรือแบบเฟสล็อกลูป ซึ่งขั้วต่อต่างกัน แสดงตามรูปดังนี้ 5-15 แสดงหน้าปัดด้านหลังของตัวดิจิทัลคอนโทรลเลอร์ ซึ่งประกอบด้วยขั้วต่อหลายแบบ ประกอบด้วยขั้วต่อ RS232 อยู่ด้านขวาใช้ติดต่อกับโฮสคอมพิวเตอร์ ขั้วต่อ i/0 ซึ่งประกอบด้วย INP,OUTP ใช้กับการควบคุมแบบต่อเนื่องโหมดปกติเช่น P,PI,PID และ PD outp ,PD inp ใช้การควบคุมแบบเฟสล็อกโหมด ขั้วต่อซัพพลาย(SUPPLY) ใช้สำหรับต่อแหล่งจ่ายแรงดันไฟฟ้าเข้า ขั้วทั้งหมดที่กล่าวมาสามารถจำแนกได้ดังนี้

INP เป็นขั้วสัญญาณอินพุตจากกระบวนการ มีแรงดันอยู่ระหว่าง 0 โวลต์ถึง 5 โวลต์

OUTP เป็นขั้วสัญญาณเอาต์พุตสู่กระบวนการ มีแรงดันอยู่ระหว่าง 0 โวลต์ถึง 5 โวลต์

PD outp เป็นขั้วสัญญาณเอาต์พุตของเฟสดีเทกเตอร์(ซอฟต์แวร์) มีลักษณะเป็นไตร-สเตต.

PD inp เป็นขั้วต่อสัญญาณอินพุตป้อนกลับจากกระบวนการ โดยที่ต้องเป็นสัญญาณรูปคลื่นสี่เหลี่ยม



รูปที่ 5-15 *แสดงหน้าปัดด้านหลังของตัวคอนโทรลเลอร์*

ต่อจากนี้จะกล่าวถึงขั้นตอนการปฏิบัติในการใช้งานของดิจิทัลคอนโทรลเลอร์ ดังนี้

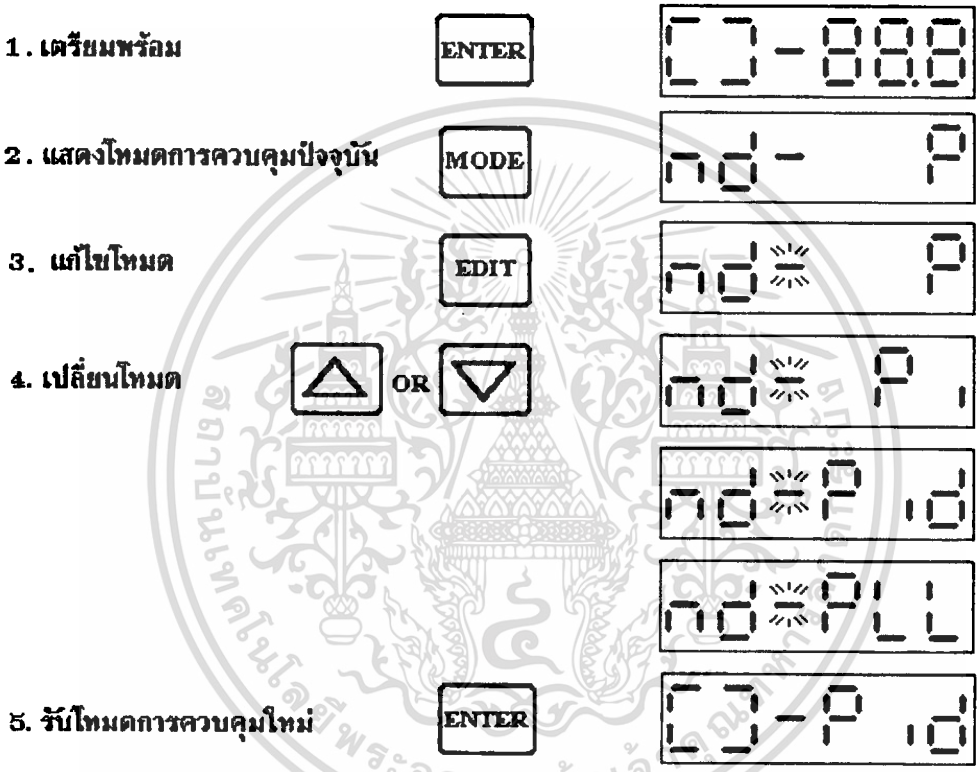
-เมื่อเริ่มเปิดเครื่อง(จ่ายแรงดันไฟฟ้าเข้าแล้ว) สังเกตว่าที่จอแสดงผลของตัวดิจิทัลคอนโทรลเลอร์จะแสดงคำว่า "READY"

-จากนั้นกดปุ่มคีย์ ENTER แล้วที่จอแสดงผลก็แสดงเป็นรูปวงเล็บใหญ่ "[ ]" หลักจากนั้นก็เป็นการเซตโหมดของการควบคุมต่างดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4.1 การเซตโหมดการควบคุม

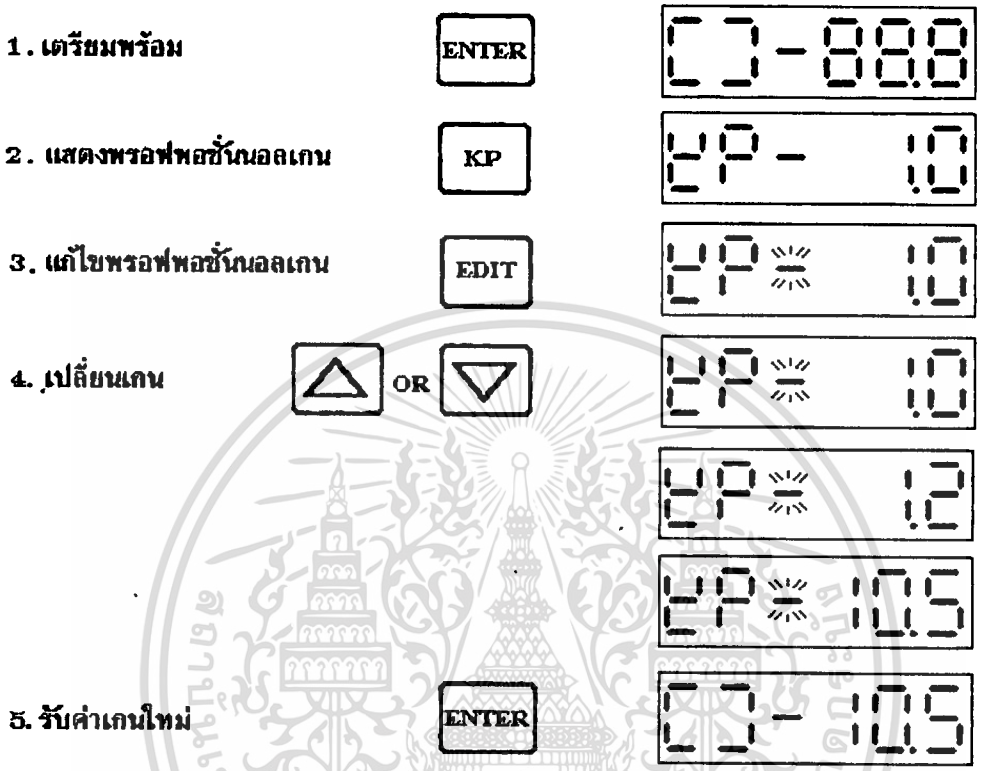
ตัวคอนโทรลเลอร์นั้นกล่าวได้ว่าส่วนที่สำคัญที่สุดคือวิธีการควบคุม เมื่อรีเซตเครื่องดิจิทัล-คอนโทรลเลอร์ จะดีฟอลต์(default)การควบคุมเป็นแบบโหมด P ถ้าต้องการเซตโหมดใหม่ทำได้โดยปฏิบัติตามการกดคีย์เรียงลำดับดังต่อไปนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4.2 การเซตพารามิเตอร์การควบคุม KP ,KI ,KD

1. วิธีเซตค่าพรอปพรอชันแนลเกน KP



ซึ่งค่าของ KP มีหน่วยเป็นจำนวนเท่า โดยถูกนิยามเหมือนกับอัตราส่วนการเปลี่ยนแปลงต่อค่าเออเรอร์กับช่วงอินพุต โดยสามารถเซตค่า KP ได้ตั้งแต่ 0.01 ถึง 999 เท่า เช่น ถ้า KP เป็น 1 จะไม่มีการขยายสัญญาณ CV (control variable)ของตัวดิจิทัลคอนโทรลเลอร์เป็นต้น

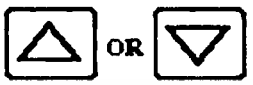
2. *วิธีเซตค่าอินทีกรัลเกน KI* ค่าพารามิเตอร์ KI จะมีผลการต่อควบคุมแบบ PI,และ PID เท่านั้น สมมุติว่าใช้การควบคุมแบบ P หรือแบบ PLL ถึงแม้จะเซตค่าพารามิเตอร์ KI เป็นอย่างไรก็ตามก็จะไม่มีผลในการควบคุมเหล่านั้น ค่า KI มีหน่วยเป็น วินาที$^{-1}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

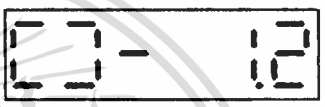
1. เตรียมพร้อม ENTER 

2. แสดงอินทีกรัลเกนปัจจุบัน KI 

3. แก้ไขอินทีกรัลเกน EDIT 

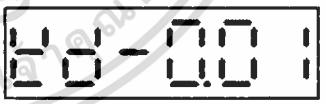
4. เปลี่ยนเกน  



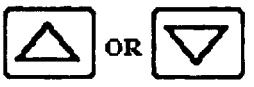
5. รับค่าเกนใหม่ ENTER 

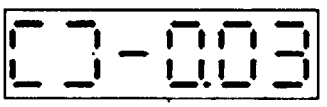
3.*วิธีเซตค่าดีริเวทีฟเกน KD* ค่าพารามิเตอร์ KD จะมีผลการต่อควบคุมแบบ PID เท่านั้น สมมุติว่าใช้การควบคุมแบบ P,PI หรือแบบ PLL ถึงแม้จะเซตค่าพารามิเตอร์ KD เป็นอย่างไรก็ตามก็จะไม่มีผลในการควบคุมเหล่านั้น ค่า KD มีหน่วยเป็น วินาที

1. เตรียมพร้อม ENTER 

2. แสดงดิริเวทีฟเกนปัจจุบัน KD 

3. แก้ไขดิริเวทีฟเกน EDIT 

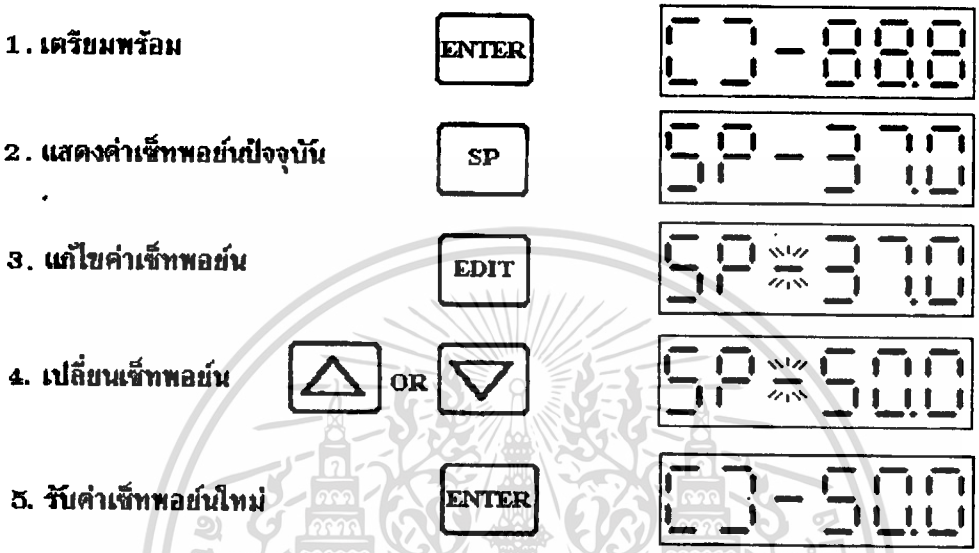
4. เปลี่ยนเกน  

5. รับค่าเกนใหม่ ENTER 

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4.3 การเซตค่าเซตพอย์น

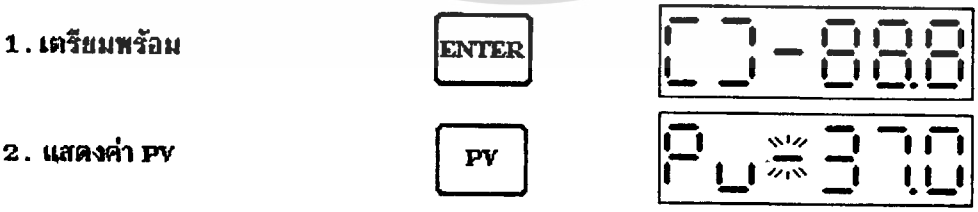
เราสามารถกำหนดเซตพอย์นให้กับตัวดิจิทัลคอนโทรลเลอร์ได้ ค่าเซตพอย์นนั้นจะต้องใส่ค่าเป็นเปอร์เซ็นต์ ตั้งแต่ 0 เปอร์เซ็นต์ ถึง 100 เปอร์เซ็นต์ ตามรูปแบบการกดคีย์ดังนี้



ตามตัวอย่างเป็นการเปลี่ยนค่าเซตพอย์นจาก 37 เปอร์เซ็นต์เป็น 50 เปอร์เซ็นต์ เป็นต้น

5.4.4 แสดงค่า process variable (PV)

โดยปกติเมื่อกระบวนการดำเนินไปอย่างต่อเนื่อง เราสามารถจะดูค่า PV (process variable) ของกระบวนการได้อย่างต่อเนื่องเช่นกัน โดยกดคีย์ตามดังนี้

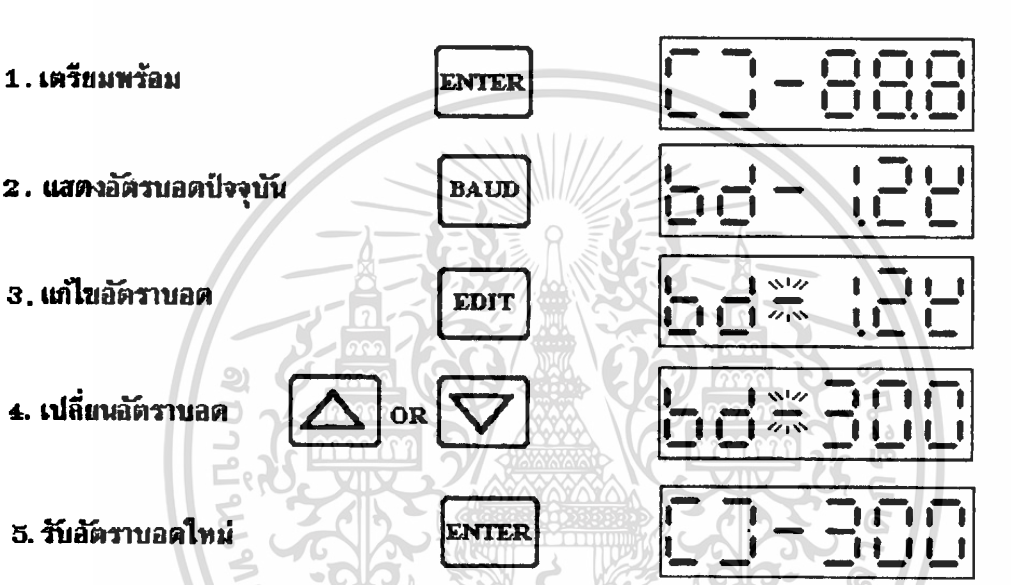


อย่างไรก็ดีค่า PV ที่แสดงบนจอแสดงผลนั้นจะถูกปรับทันกาล(update)ประมาณทุกๆ 1 วินาทีโดยสังเกตจากจังหวะการกระพริบของเคอร์เซอร์ ค่า PV ที่แสดงออกมาบนจอแสดงผลนั้น จะเป็นหน่วยเปอร์เซ็นต์ เช่น ตามตัวอย่าง 37.0 เท่ากับ 37.0 เปอร์เซ็นต์ ดังนั้นการแสดงค่า PV จึงเป็นไปได้อยู่ในช่วง 0 เปอร์เซ็นต์ ถึง 100 เปอร์เซ็นต์ ถ้าเคอร์เซอร์ไม่กระพริบในตอนนี้ แสดงว่ามีเออเรอร์เกิดขึ้น ซึ่งต้องรีเซตเครื่องใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4.5 การเซตอัตราบอด(Baud Rate)

อัตราบอด (baud rate) เป็นอัตราการส่งข้อมูลแบบอนุกรมแบบ RS232 ซึ่งจำเป็นสำหรับตัวดิจิทัลคอนโทรลเลอร์ต้องติดต่อสื่อสารกับโฮสคอมพิวเตอร์ ซึ่งการเซตค่าบอดกำหนดให้ผู้ใช้เซตค่าได้ ตั้งแต่ 150, 300, 600, 1200, 2400, 4800, 9600 บอด ดังนั้นการเซตอัตราบอดต้องเหมือนกับอัตราบอดของโฮส

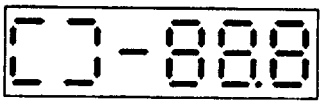


ตามตัวอย่างการกดคีย์ข้างต้น ลำดับที่ 2 ค่าที่ใช้ให้ปัจจุบันคือ 1200 บอดหรือ 1.2K บอด โดยเปลี่ยนแปลงใหม่เป็นอัตราบอด 300 บอด แล้วลำดับที่ 5 กดคีย์ ENTER รับอัตราบอดใหม่ไปใช้

5.4.6 การเซตหมายเลขสถานีแอดเดรส (Station Address)

ดังที่ทราบมาจากบทที่ 5 ตอนต้นนั้นการเชื่อมโยงตัวดิจิทัลคอนโทรลเลอร์หลายตัวกับโฮสคอมพิวเตอร์ นั้นจำเป็นต้องกำหนดรูปแบบเนตเวอร์กของการสื่อสารแบบอนุกรม ซึ่งได้เลือกใช้การต่อแบบลูปเนตเวอร์กไว้ดังกล่าว ดังนั้นเพื่อจะให้โฮสคอมพิวเตอร์รับรู้หมายเลขสถานีแอดเดรสของดิจิทัลคอนโทรลเลอร์ จึงเป็นต้องมีการเซตหมายเลขสถานีให้กับตัวดิจิทัลคอนโทรลเลอร์ โดยสามารถเซตได้ ตั้งแต่หมายเลข 1 ถึงหมายเลข 12 ซึ่งหมายความว่าในหนึ่งกิ่งสามารถต่อกับตัวคอนโทรล เลอร์ได้ 12 ตัวตามหมายเลข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เตรียมพร้อม ENTER 

2. แสดงค่าสถานีแอดเดรสปัจจุบัน ADDR 

3. แก้ไขสถานีแอดเดรส EDIT 

4. เปลี่ยนค่าสถานี  OR  

5. รับค่าสถานีแอดเดรสใหม่ ENTER 

ตามตัวอย่างเป็นการเปลี่ยนหมายเลขสถานีแอดเดรส จาก 4 เป็นหมายเลข 12 เป็นต้น

5.5 วิธีการติดตั้งโปรแกรมพล้านต์มอนิเตอริง

ในหัวข้อนี้เป็นการแนะนำการใช้ การติดตั้งโปรแกรมพล้านต์มอนิเตอริง ดังเสนอต่อไปนี้

5.5.1 การแนะนำการติดตั้งโปรแกรม

โปรแกรมพล้านต์มอนิเตอริง(plant monitoring program) ง่ายต่อการใช้งาน มีความสำคัญมากในการใช้กับงานประยุกต์ประเภทนี้ หัวข้อนี้จะบอกรายละเอียดกระบวนการติดตั้งและพื้นฐานการใช้เมาส์ การเรียนรู้เรื่องราวในการใช้โปรแกรมจะทำให้มีความเข้าใจและหลีกเลี่ยงปัญหาต่างๆ

1) การติดตั้งโปรแกรมพล้านต์มอนิเตอริง(Installation)

นำไฟล์ทั้งหมดกอบปี้ไปในไดเรกเทอรี่เดียวกัน ในฮาร์ดดิสค์ หรือฟลอปปี้ดิสก์ เพราะว่าขนาดของไฟล์ใช้งาน รวมทั้งหมดไม่เกิน 1.2 เมกะไบต์ สิ่งต่อไปที่ต้องทำคือตรวจดูฮาร์ดแวร์บน PC ทั้งหมดเป็นไป ตามรายละเอียดต่อไปนี้หรือไม่

2) ระบบฮาร์ดแวร์ที่ต้องการ

โปรแกรมพล้านต์มอนิเตอริงต้องกับใช้เครื่องคอมพิวเตอร์ส่วนบุคคล IBM-PC หรือเครื่องคอมแพตติเบิล ซึ่งต้องคุณสมบัติตามรายการ ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-หน่วยความจำหลักเท่ากับ 640 กิโลไบต์ หรืออย่างน้อย 560 กิโลไบต์สำหรับใช้รันโปรแกรมพล้านต์มอนิเตอริง

-ไม่ต้องการ math coprocessor

-ดิสก์ไดรว์ 5-1/4 นิ้ว หรือดิสก์ไดรว์ 3-1/2 นิ้ว หรือฮาร์ดดิสค์

-ใช้จอแสดงผล VGA อย่างเดียวเท่านั้น

-เมาส์แบบ 3 ปุ่มต่อที่ซ่อง COM1 เท่านั้น หรือต่อเข้าโดยทางเมาส์อินเตอร์เฟสการ์ด

-พอร์ตขนานสำหรับต่อกับเครื่องพิมพ์

3) คำแนะนำการเริ่มต้นรันโปรแกรมพล้านต์มอนิเตอริง(Plant Monitoring)

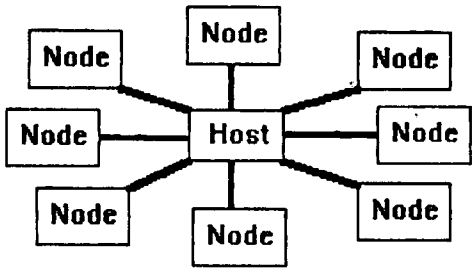
โปรแกรมนี้ใช้สำหรับเครื่องคอมพิวเตอร์ส่วนบุคคลของ IBM และเครื่องคอมแพตติเบิล โดยสร้างขึ้นมาสำหรับงานทางด้านควบคุมกระบวนการ ในลักษณะเป็นการแสดงผลของพล้านต์ เช่น แสดงเป็นกราฟบันทึกแนวโน้ม(trend recorder) แสดงเป็นกราฟบาร์(bar graph)ของระบบ Data aqusition เป็นต้น อีกทั้งมีฟังก์ชันแบบพูลดาว์นเมนู(pull down menu)ต่างๆเพื่อให้ใช้งานได้ง่ายมากขึ้น

ซอฟต์แวร์โปรแกรมตัวนี้เป็นซอฟต์แวร์สร้างมาจากซอร์สโค๊ต เขียนขึ้นจากภาษา C แบบ OOP หรือ ออปเจ็กโอเรียลเต็ด (object orientd program) ซึ่งใช้มาตราฐานของ BORLAND เป็นหลัก ซึ่ง OOP เป็นลักษณะการเขียนโปรแกรมซึ่งพัฒนามาจากภาษาแบบโครงสร้าง(sturcture)ทำให้การเขียนโปรแกรมขนาดใหญ่ดีได้ ซอฟต์แวร์โปรแกรมพล้านต์มอนิเตอริงนี้ไม่สามารถทำงานให้สมบูรณ์เพียงลำพังได้ แต่ต้องมีตัวดิจิทัลคอนโทรลเลอร์ภายนอกสนับสนุนการใช้งานด้วย ซึ่งดิจิทัลคอนโทรลเลอร์นี้ได้บรรยายมาแล้วในเนื้อหาบทต้น โดยออกแบบให้สามารถติดต่อกับเครื่อง ด้วยการสื่อสารแบบอนุกรมใช้พอร์ตมาตราฐานแบบ RS-232C บนสิ่งแวดล้อม(enveronment)ของโปรแกรมพล้านต์มอนิเตอริง ซึ่งเปรียบได้ว่าเครื่อง PC เป็น หน่วยหลัก(host) ที่รับข้อมูลจากตัวดิจิทัลคอนโทรลซึ่งจะติดตั้งอยู่ในพื้นที่ของกระบวนการมาแสดงผลแบบ ต่างๆ

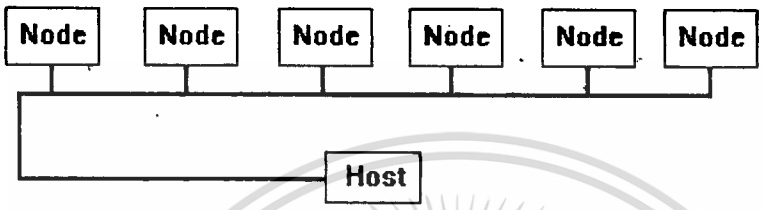
4) การเชื่อมโยงระหว่าง PC และคอนโทรลเลอร์

มีมาตราฐานสำหรับติดต่อระหว่าง PC กับตัวดิจิทัลคอนโทรลเลอร์ภายนอก ใช้มาตราฐานการเชื่อมโยงแบบ RS 232 ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



แบบ สตาร์



แบบ ลูป

รูปที่ 5-16 *การกำหนดรูปแบบการติดต่อสื่อสารแบบอนุกรม*

-พอร์ตอนุกรมหมายเลข COM1:(แอดเดรส 3F8) COM2:(แอดเดรส 2F8) โดยโยงไม่ควรกำหนด การติดตั้งซ้ำกับเมาส์

-กำหนดอัตราความเร็วในการส่งของมูล(baud)ให้มีค่าเดียวกันระหว่าง PC และตัวดิจิทัลคอนโทรลเลอร์ โดยมีขนาดของบอด(baud)เป็น 150 300 600 1200 2400 4800 9600 ให้ เลือกใช้

-การส่งข้อมูลผ่านแบบอนุกรมใช้ข้อมูลกำหนดตามแบบ ASCII ตามหลักการรับส่งข้อมูลแบบอะซิงโครนัสหรือเรียกย่อว่า UARTS ซึ่งขบวนสตีม(steam)เป็น 10 บิตโดยที่บิตแรกเป็นบิตสตาร์ต(start bit) บิตสุดท้ายเป็นบิตหยุด(stop bit)โดย 8 บิตที่เหลือเป็นบิตข้อมูลดังกล่าว

-ใช้เครือข่าย(network)) แบบลูปโดยสามารถเชื่อมโยงตัวดิจิทัลคอนโทรลเลอร์ได้ 12 หน่วย

5) การกำหนดโครงแบบเครือข่าย(Network Configulation)

ปัญหาอย่างแรกที่จะต้องเผชิญคือระบบเครือข่าย คือทำอย่างไรที่จะเกี่ยวโยงทางกายภาพของ คอมพิวเตอร์เข้าด้วยกันแสดงตามรูปที่ 5.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครือข่ายแบบสตาร์(star network) มีสายเดียวเชื่อมโยงระหว่างคอมพิวเตอร์ศูนย์กลางกับแต่ละคอมพิวเตอร์อื่น หรือเรียกอย่างง่ายว่าจากโฮส(host)สู่โหนด(node) การกำหนดรูปแบบเครือข่ายแบบนี้ซึ่งโดยปกติจะประยุกต์การใช้งานแบบไทม์แชริง(time sharing)

เครือข่ายแบบลูป(loop network) จะมีสายเดียวติดต่อกับโหนดทั้งหมด โดยมีโฮส(host)หนึ่งหน่วยจัดการรูปแบบควบคุมการทำงานบนลูป ซึ่งโฮสจะทำการอีนาเบิลให้โหนดตัวไหนในแต่ละเวลาที่กำหนดให้ โดยที่โค็ด(code)อินาเบิลของโหนดแต่ละตัวไม่เหมือนกัน ในเครือข่ายแบบลูปโดยปกติจะถูกใช้ในการติดตามเก็บสะสมข้อมูล(data-gathering) โดยที่จะสอบถามมาจากแต่ละโหนดตามลำดับแล้วมาสะสมข้อมูลไว้ที่โฮส(host) โดยแต่ละข้อมูลจะถูกอัฟเดท(update)ล่าสุด ในแต่ละโหนดทั้งหมดข้อมูลที่จะติดต่อสื่อสารระหว่างโฮสนั้นต้องทำเป็นส่วนตัว

การเลือกกำหนดรูปแบบของเครือข่าย ต้องขึ้นอยู่กับเงื่อนไขภายนอกหลายประการซึ่งขึ้นอยู่กับผู้ออกแบบเอง ซึ่งมีเครื่องชี้แนวทางโดยทั่วไปดังนี้

วัตถุประสงค์	แบบเนตเวอร์ก	ข้อคิดเห็น
ความเชื่อถือได้	สตาร์	สูญเสีย 1 โหนดต่อ 1 สาย
แยกความผิดพลาด	สตาร์	ติดตามความผิดพลาดได้
ความเร็ว	สตาร์	แต่ละโหนดสมบูรณ์ในตัว
ราคา	ลูป	ใช้สายเดียวกับโหมดทั้งหมด

ตารางที่ 5-2 *แสดงความแตกต่างระหว่างเนตเวอร์กแบบสตาร์ และแบบลูป*

เครือข่ายแบบสตาร์เป็นตัวเลือกที่ดีเมื่อมีจำนวนโหนดน้อย หรือระยะทางจากโฮสถึงโหนดสั้น แต่ถ้าต่อโหนดมากขึ้นราคาค่าใช้จ่ายของสายเคเบิลมากขึ้นเป็นเงาตามตัว ซึ่งเครือข่ายแบบลูปย่อมดีกว่า

ในเรื่องนี้ ซึ่งงานวิจัยนี้จะใช้เครือข่ายแบบลูปซึ่งโฮส(host)คือคอมพิวเตอร์ PC และ โหนดคือตัวดิจิทัลคอนโทรลเลอร์ ซึ่งเชื่อมโยงตัวดิจิทัลคอนโทรลเลอร์ 12 ตัว

5.5.2 ระบบงานของโปรแกรม(Run Time System)

รูปแบบโดยทั่วไปของโปรแกรมจะเตรียมฟังก์ชันและความสามารถต่างๆไว้ดังต่อไปนี้

-ระบบ OS ใช้ DOS 3.2 ขึ้นไป

-ระบบติดต่อกับผู้ใช้เป็นลักษณะของพูลดาวน์เมนู(pull-down menu)

-สนับสนุนทั้งคีย์บอร์ด และ เมาส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-เซตเงื่อนไขเริ่มต้น รวมทั้ง Help

-ไม่มี พาสเวิอร์ด

-ไม่มี ซิมูเลชัน(simulation) แต่ต้องมีฮาร์ดแวร์สนับสนุน

-ค่าพารามิเตอร์ของระบบ

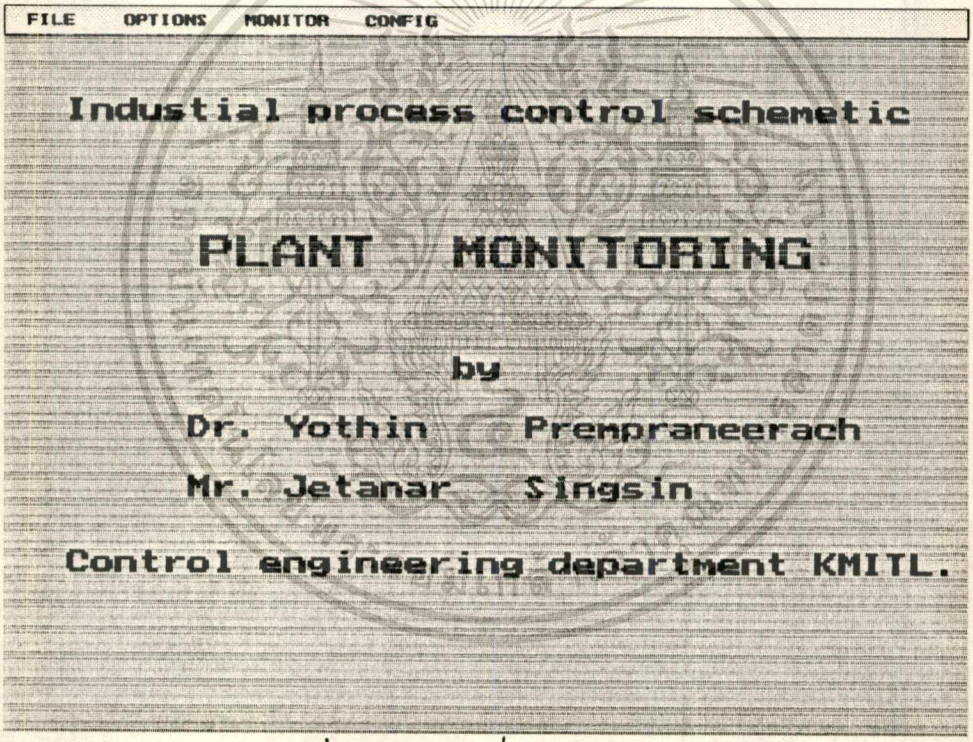
-เลือกไฟล์สำหรับผู้ใช้

-เซตป้าย(tag edit)

-แสดงผลแบบเวลาจริง (real time)

-เทรนเรคคอร์ดดิง(rrend recording)

-ข่าวสารการทำงาน(operator messages)



รูปที่ 5-17 *ไตเติลเริ่มต้นของโปรแกรม*

5.5.3 เริ่มต้นรันโปรแกรมพล้านต์มอนิเตอริง

เมื่อได้รับแผ่นโปรแกรมมาแล้วเมื่อจะประกอบด้วยหลายไฟล์ด้วยกัน มีทั้งซอร์สไฟส์ และไฟล์อื่นๆ โดยที่ไฟล์ใช้รันโปรแกรมประกอบด้วยไฟล์ PNTMNT.EXE , PLANT.CFG , *.TAG (ไฟล์ทั้งหมดที่มี นามสกุล TAG) โดยเรียกรันโปรแกรมดังนี้

C:\>PNTMNT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

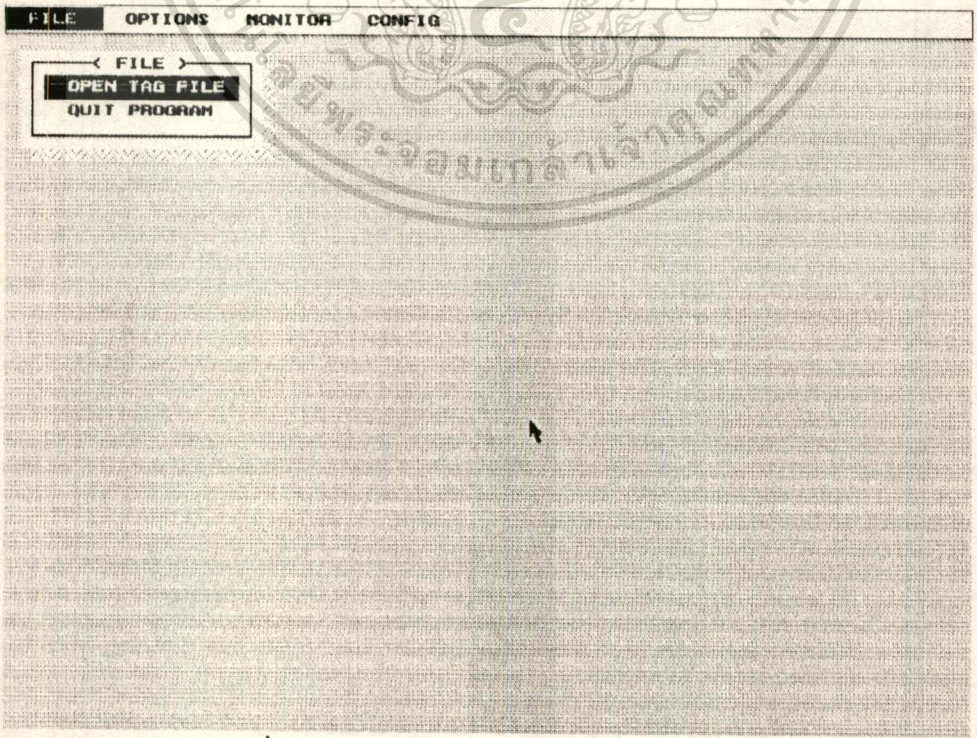
เมื่อเริ่มต้นทำงานของโปรแกรมพล้านต์มอนิเตอริงแสดงตามรูปที่ 5-17 ซึ่งจะเห็นว่าประกอบไปด้วย ไตเติลของผู้เขียน อาจารย์ที่ปรึกษาและชื่อโปรแกรม ต่อไปกดรีเทอร์นจะเป็นคำแนะนำเบื้องต้น ซึ่งที่จอภาพด้านบนมีเมนูบาร์หลัก(main menu bar)ปรากฏอยู่ มีตัวเลือกอยู่ด้านบนซึ่งเมนูบาร์หลัก ประกอบด้วยตัวเลือกดังนี้

1. เมนูหลัก <FILE>
2. เมนูหลัก <OPTIONS>
3. เมนูหลัก <DISPLAY>
4. เมนูหลัก <CONFIG>

เมนูบาร์หลักเหล่านี้เป็นเมนูหลัก ซึ่งในแต่ละเมนูหลักแบ่งเป็นเมนูย่อย(Submenu)แยกลงไปได้อีกดังนี้

5.5.4 การใช้เมนูหลักชื่อ <FILE>

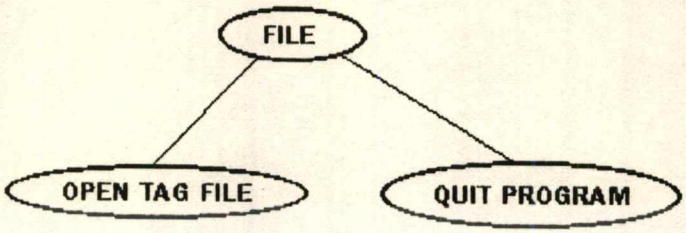
เป็นเมนูหลัก <FILE> นี้เกี่ยวข้องกับการทำงานของไฟล์ต่างซึ่งโปรแกรมต้องการโหลดมาใช้ซึ่งผู้ใช้ต้องกำหนดเอง แบ่งเป็นเมนูย่อยสองเมนูคือ <OPEN TAG FILE> และ <QUIT PROGRAM> แสดงดังรูปที่ 5-18 ซึ่งมีรายละเอียดดังนี้



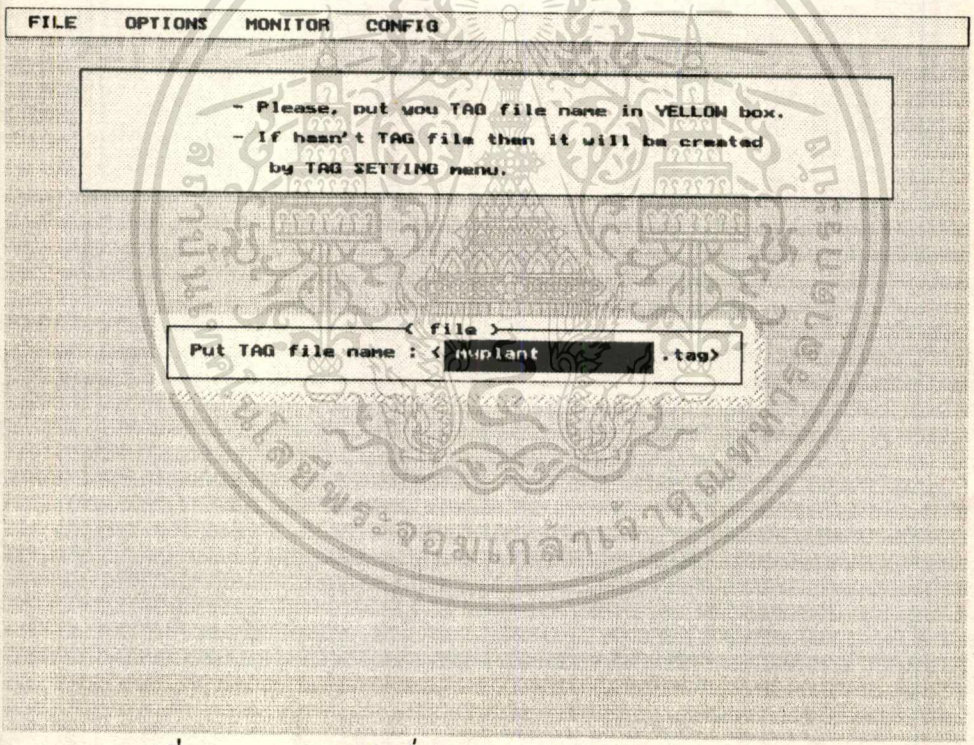
รูปที่ 5-18 *แสดงเมนูหลัก <FILE> และเมนูย่อย*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงเป็นบล็อกไดอะแกรมดังนี้



ซึ่งได้อธิบายเมนูย่อย(submenu)ของ FILE ซึ่งเป็นเมนูหลักดังนี้



รูปที่ 5-19 *แสดงการใส่ชื่อไฟล์ของเมนูย่อย<OPEN TAG FILE>*

1) เมนูย่อย <OPEN TAG FILE>

ในการรันโปรแกรม.จะต้องเปิด TAG ไฟล์คือไฟล์ต่างซึ่งมี นามสกุล *.tag ซึ่งไฟล์เหล่านี้จะเก็บค่าพารามิเตอร์ซึ่งเกี่ยวข้องกับกราฟของ trend และกราฟ บาร์ เช่น แอมปลิจูดของกระบวนการ(process) ช่วงของแอมปลิจูด ชื่อของลูปกระบวนการ หรือชื่อของกราฟบันทึกแนวโน้มข้อมูล และ คาบของเวลา ไว้ซึ่งเคยถูกเก็บ(save)ไฟล์ไว้ก่อนหน้านี้ โดยพารามิเตอร์ดังกล่าว ผู้ใช้สามารถเข้าไปแก้ไข หรือสร้างใหม่ในเมนูย่อย<TAG EDIT> ของเมนูหลักที่มีชื่อว่า <MONITOR> เมนูย่อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

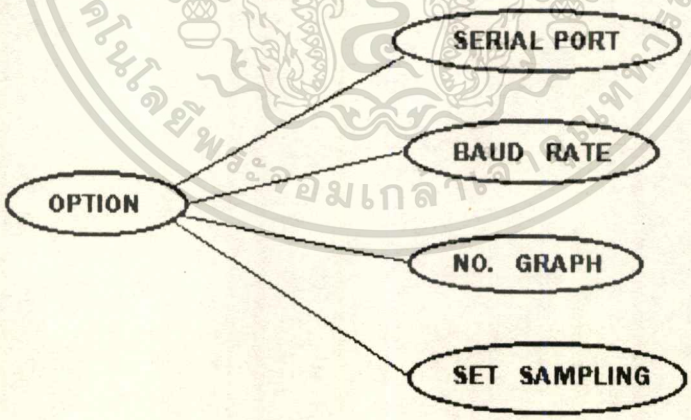
<OPEN TAG FILE> เมื่อถูกเลือกจะแสดงดังรูปที่ 5-19 ซึ่งมีการดำเนิน การถามชื่อไฟล์ให้ผู้ใช้ใส่เข้าไป โดยผู้ใช้กดคีย์ตัวอักษร ซึ่งตัวอักษรนั้นจะไปปรากฏที่บล็อกสีเหลือง ตามลำดับ ชื่อไฟล์ที่ใส่นั้นจะต้องมีนามสกุล(extention)เป็น *.TAG โดยที่ใส่เฉพาะชื่อไฟล์เท่านั้นไม่ต้องใส่นามสกุล เมื่อไม่มีชื่อไพล์ตามที่ผู้ใช้กำหนดโปรแกรมจะทำการตักเตือน เมื่อเสร็จเรียบร้อยแล้วโปรแกรมพล้านต์มอนิ-เตอริง นี้พร้อมที่จะทำงานอย่างสมบูรณ์ อย่าลืมถ้าไม่มี *.TAG ไฟล์จะต้องสร้างไฟล์ *.TAG ก่อน

2) เมนูย่อย <QUIT PROGRAM>

เป็นเมนูย่อยเรียกใช้เพื่อต้องการออกจากโปรแกรมไปสู่สิ่งแวดล้อมดอส(DOS environment) อย่างถาวร ซึ่งโปรแกรมพล้านต์มอนิเตอริงจะเลิกทำงานโดยสมบูรณ์ โดยที่ตัวดิจิทัลคอนโทรลเลอร์ซึ่งอยู่ในพล้านต์(plant) หรือเรียกว่า "โลคอลคอนโทรลเลอร์" (local controler)จะทำงานตามปกติเพียงแต่จะตัดขาดจากโฮสคอมพิวเตอร์เท่านั้น

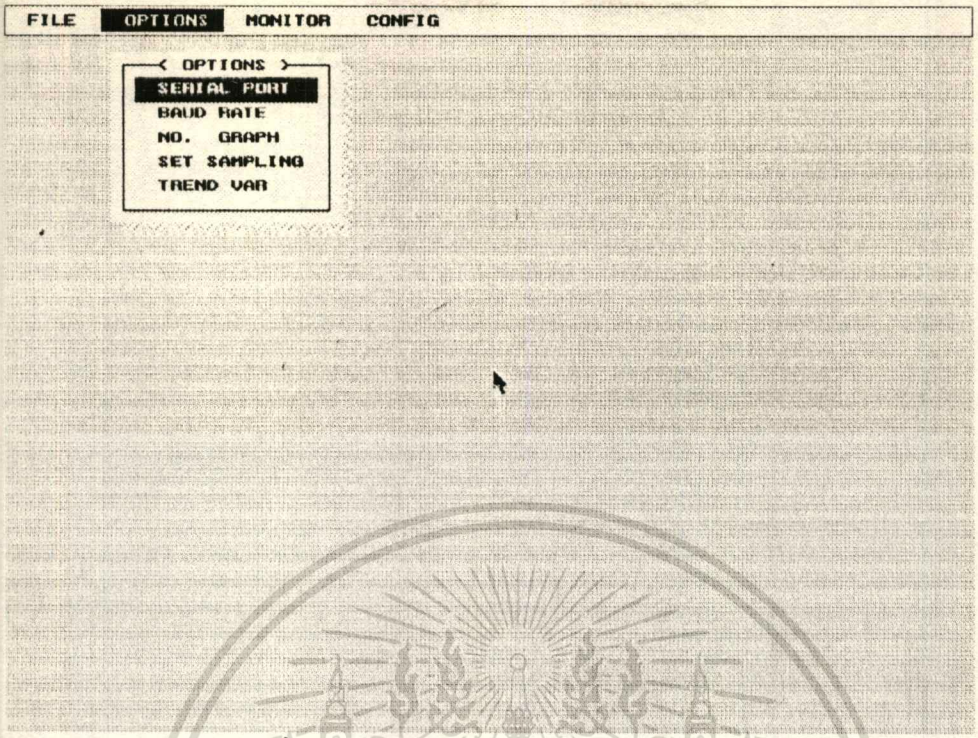
5.5.5 การใช้เมนูหลักชื่อ <OPTIONS>

เป็นเมนูเกี่ยวกับการเซตพารามิเตอร์ต่างๆ จะเกี่ยวข้องกับการเชื่อมต่อแบบอนุกรม(serial interface) การกำหนดจำนวนกราฟ การกำหนดแซมปลิง ซึ่งมีเมนูย่อยดังนี้

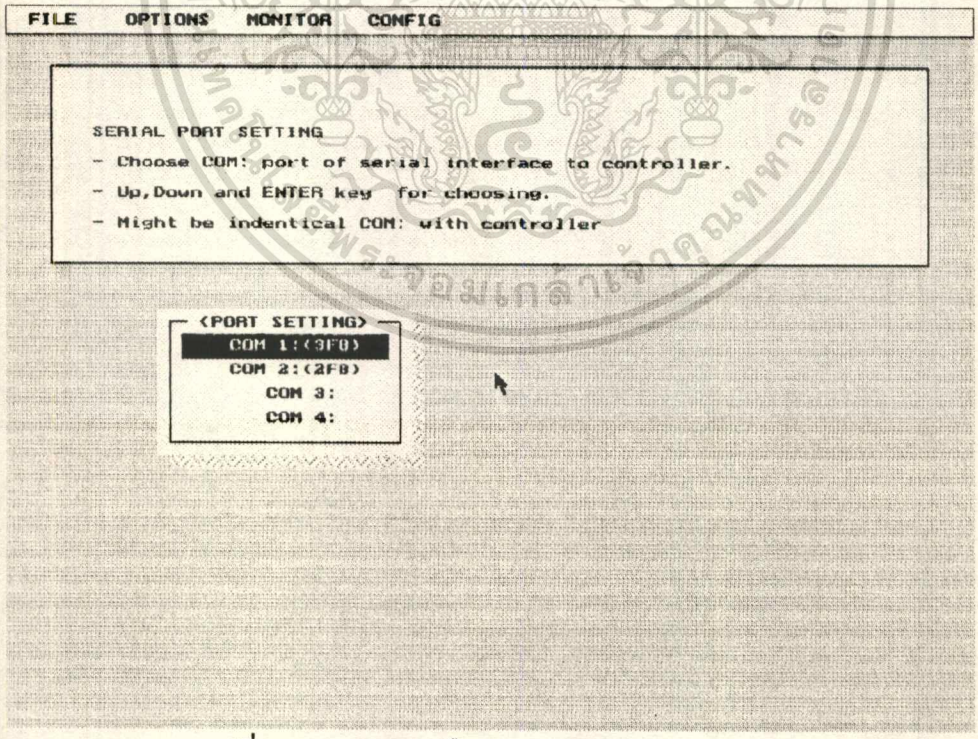


จากรูปที่ 5-20 แสดงเมนูย่อยของเมนูหลัก <OPTIONS> โดยเมนูย่อยเรียงจากบนไปล่างตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-20 *เมนูย่อยของเมนูหลัก<OPTIONS>*



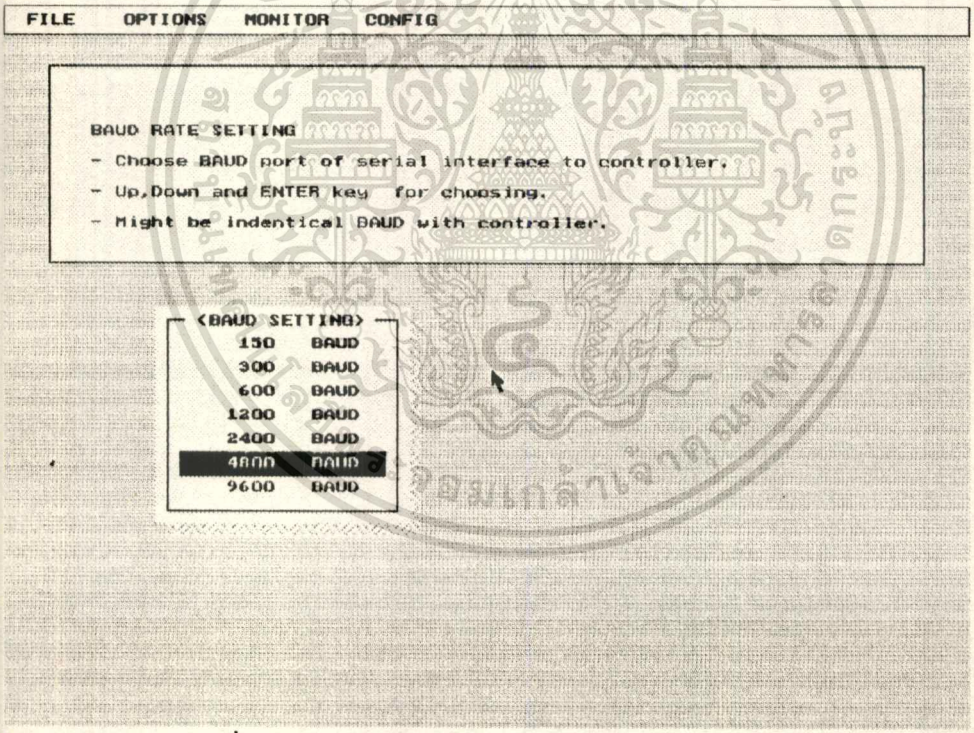
รูปที่ 5-21 *แสดงตัวเลือกช่องการติดต่อแบบอนุกรม*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1)เมนูย่อย<SERIAL PORT>

ตัวโฮสคอมพิวเตอร์จะติดต่อกับดิจิทัลตัวคอนโทรลเลอร์ โดยการสื่อสารเชื่อมโยงแบบอนุกรมดังกล่าว ซึ่งใช้มาตราฐานของ RS-232C ดังนั้นการติดต่อกันดังกล่าวโฮสคอมพิวเตอร์ จะมีคอนเนคเตอร์ใช้กับการติดต่อแบบซีเรียลหลายช่องเช่น COM1,COM2 เป็นต้น ซึ่งจำเป็นจะต้องเซตช่องการติดต่อแบบซีเรียลให้ถูกต้องตามที่ผู้ใช้ติดตั้งเพื่อติดต่อกับตัวคอนโทรลเลอร์ โดยโปรแกรมนี้ใช้ช่อง COM1 มีหมายเลขถอดรหัสพอร์ตเป็น 3F8 และช่อง COM2 เป็น 2F8 ถ้าสมมุติว่า COM1 และ COM2 ไม่ตรงกับหมายเลขถอดรหัสพอร์ตที่กำหนดให้แล้ว ให้ยึดถือหมายเลขถอดรหัสเป็นหลัก

เมื่อเข้าสู่เมนูย่อย<SERIAL PORT>แล้วจอภาพจะแสดงผลดังรูป 5-21 บนจอภาพ มีแสดงเป็นคำแนะนำ และมีตัวเลือกของช่องอนุกรม COM1,COM2,COM3 และ COM4 โดยผู้ใช้กดคีย์เคอร์เซอร์ขึ้นหรือลงแล้วกดคีย์รีเทอร์นเพื่อรับค่า หรือสะดวกจะใช้เมาส์ก็ได้ แสดงดังรูป 5-21



รูปที่ 5-22 *แสดงตัวเลือกของเมนูย่อย<BAUD RATE>*

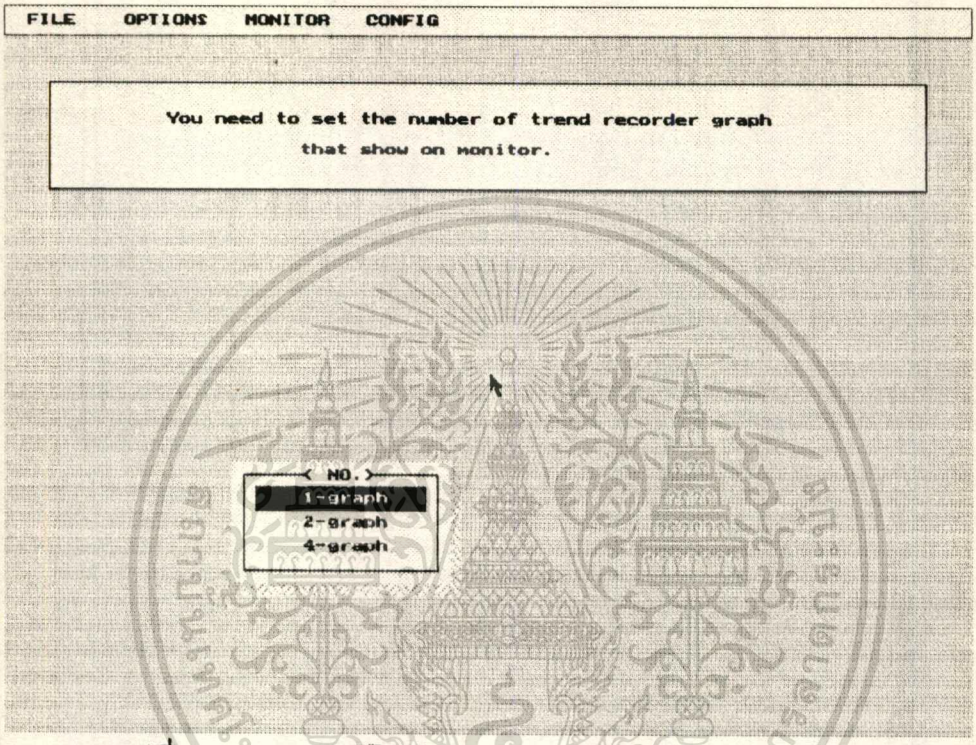
2)เมนูย่อย <BAUD RATE>

เมนูย่อยนี้สำหรับเซตอัตราความเร็วในการส่งข้อมูลทาง RS-232C มีหน่วยเป็น baud มีให้เลือกเซตตั้งแต่ 150,300,640,1200,2400,4800 และ 9600 ผู้ใช้เองต้องเลือกให้เหมาะสมกับการติดต่อกับตัวคอนโทรลเลอร์ ตัวอย่างเช่นถ้าระยะทางไกลหรือในสภาวะแวดล้อมมีสัญญาณรบกวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มากใช้อัตรา baud ต่ำเป็นต้น อย่างไรก็ดีควรจะให้มีอัตราความเร็วในการส่งข้อมูลมากที่สุดเท่าที่เป็นไปได้ เมื่อเข้าสู่เมนูย่อย<BAUD RATE>ถูกเลือก จอภาพจะถูกแสดงผลเป็นดังรูป 5-22

3)เมนูย่อย <NO. GRAPH>



รูปที่ 5-23 *แสดงตัวเลือกจำนวนกราฟของเมนูย่อย<NO. GRAPH>*

เมนูย่อยนี้จะเกี่ยวข้องกับ จำนวนการแสดงผล(display)ของกราฟแนวโน้มข้อมูล ในจอภาพ ซึ่งในเมนูย่อยของ<SET GRAPH>มีเมนูย่อยลงไปอีก เป็นเมนูสำหรับเลือกจำนวนของกราฟเป็น 1, 2 และ 4 กราฟ อย่างไรก็ดีเมื่อเลือกตามจำนวนที่ต้องการแล้วโปรแกรมจะเก็บพารามิเตอร์ไว้อัตโนมัติ เพื่อนำไปใช้ร่วมในเมนูย่อย<TREND RED> ของเมนูหลัก <MONITOR>

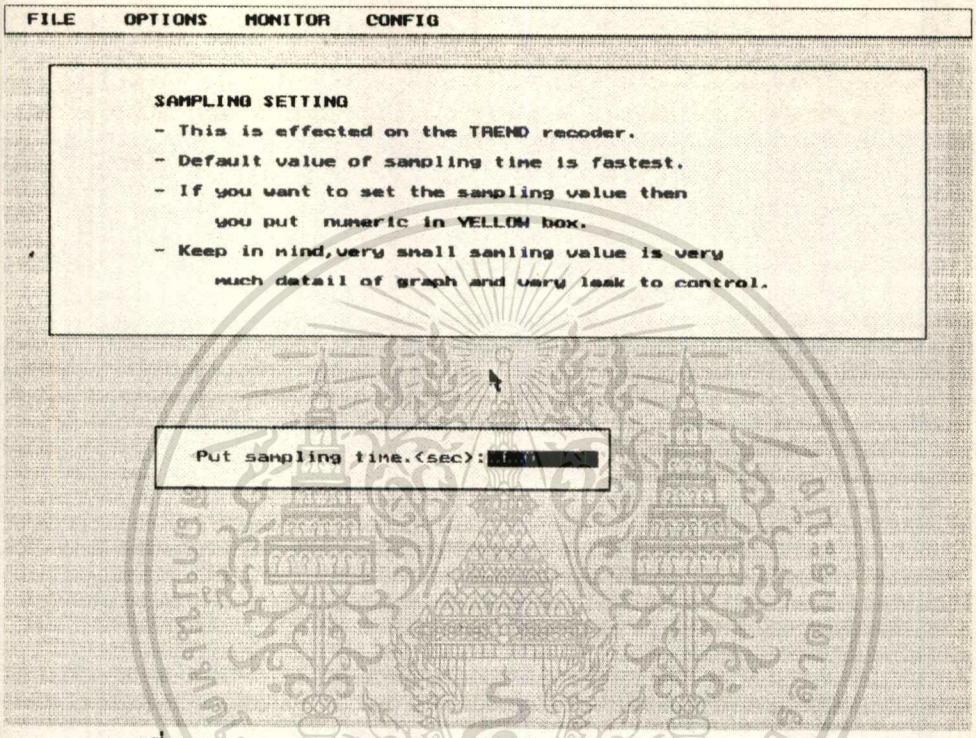
เมื่อเมนูย่อย <NO. GRAPH> ถูกเลือกแล้วจอภาพจะแสดงผลดังรูปที่ 5-23 เพื่อให้ผู้ใช้เลือกจำนวนกราฟตามต้องการ

4) เมนูย่อย <SET SAMPLING>

เป็นเมนูย่อยซึ่งตั้งค่าเวลาแซมปลิงในหน่วยของวินาที เวลาแซมปลิงของโปรแกรมพล้านต์มอนิเตอริงของคอมพิวเตอร์ PC คือช่วงเวลาแต่ละครั้งในการติดต่อกับตัวดิจิทัลคอนโทรลเลอร์ เช่น ถ้าใส่ค่าเวลาแซมปลิงเป็น 0.0 วินาที นี้หมายความว่าโฮส จะติดต่อกับตัวดิจิทัลคอนโทรลเลอร์ ตลอดเวลาที่ต้องการแสดงผลกราฟบันทึกแนวโน้ม(trend) หรือ กราฟบาร์ อย่างไรก็ตามการตั้งค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เวลาแซมปลิงเป็น 0.0 วินาที การแสดงผลกราฟบันทึกแนวโน้ม trend จะมีเส้นกราฟละเอียดต่อเนื่อง แต่ก็มีผลเสียบ้างเล็กน้อยคือ รบกวนการทำงานของตัวดิจิทัลคอนโทรลเลอร์ตามปกติ โดยที่ตัวดิจิทัลคอนโทรลเลอร์ต้องแบ่งเวลามาเพื่อตอบรับการสื่อสารระหว่างกัน อย่างไรก็ตามในระบบที่มีความเร็วการตอบสนองช้า แนะนำให้ใช้เวลาแซมปลิงมากกว่า 5.0 วินาทีเป็นต้น



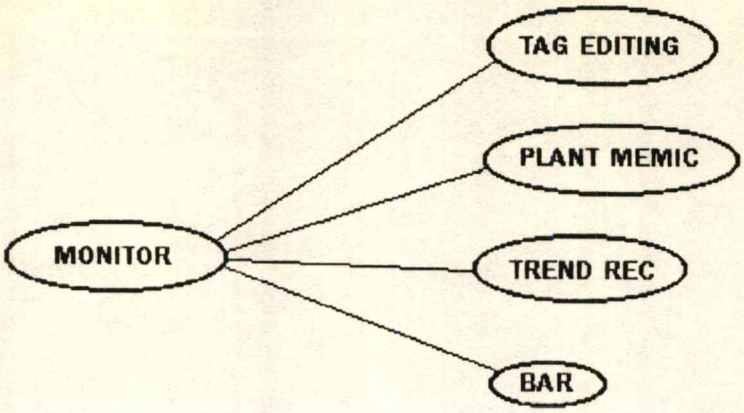
รูปที่ 5-24 *แสดงการใส่ค่าแซมปลิงในเมนูย่อย <SET SAMPLING>*

เมื่อเข้าสู่เมนูย่อย <SET SAMPLING> จอภาพจะแสดงผลดังรูปที่ 5-24 ตรงกรอบสี่เหลี่ยมด้านล่างในบล็อกสีเหลืองจะแสดงค่าของเวลาแซมปลิงที่เคยเซตไว้เดิม ซึ่งสามารถแก้ไขเปลี่ยนตัวเลขใหม่ได้ทันที โดยกดคีย์ตัวเลขที่ต้องการก็สามารถแทนที่ตัวเลขเดิมอัตโนมัติ เมื่อใส่ค่าตัวเลขเรียบร้อยแล้วกดปุ่ม ENTER เป็นเสร็จอันกระบวนการ

5.5.6 การใช้เมนูหลักชื่อ <MONITOR>

เป็นเมนูเกี่ยวข้องกับการแสดงผลค่าตัวแปรควบคุม PV (process variable) ,ค่าเซตพอย์น SP (setpoint) ตัวแปรควบคุม CV (control variable เป็นต้น

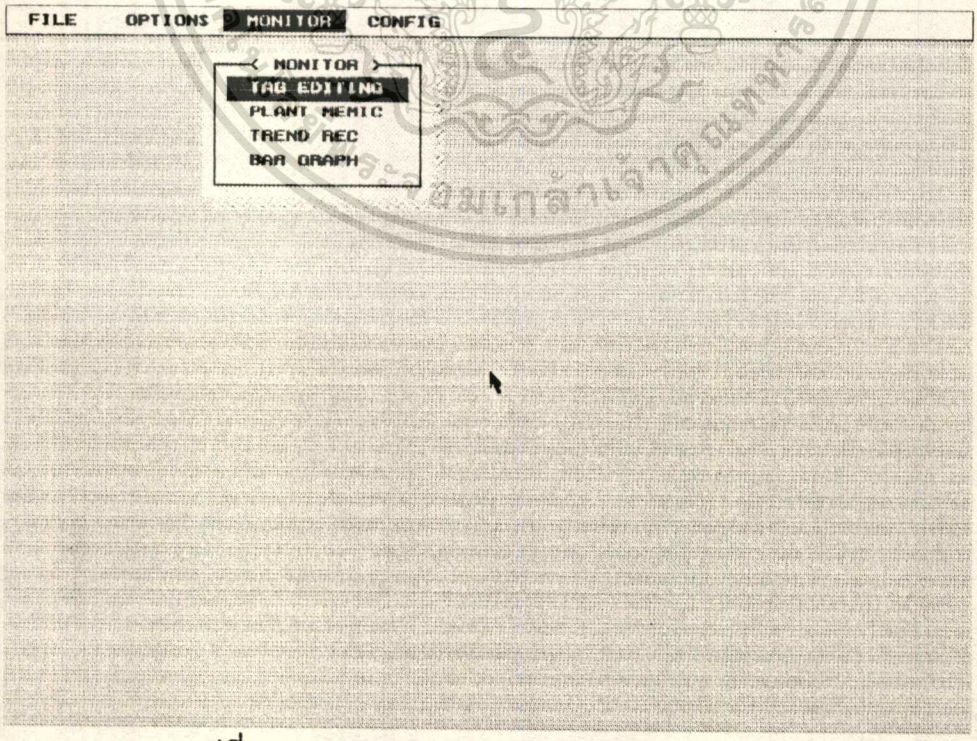
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



จาก รูปที่ 5-25 แสดงเมนูย่อยของเมนูหลัก <MONITOR> บนจอภาพ ซึ่งเป็นเมนูแบบพลูดาวน์ โดยสามารถกดคีย์ขึ้นหรือลงเพื่อเลือกเมนูย่อยตามต้องการ

1)เมนูย่อย <TAG EDITING>

เป็นเมนูย่อยซึ่งมีหน้าที่กำหนดชื่อของแต่ละกราฟ พารามิเตอร์ ต่างของกราฟ ซึ่งคำว่า TAG คำนี้มีความหมายว่า สลาก หรือป้ายชื่อ ดังนั้นเมนูย่อย<TAG SEETTING> เป็นการใส่ชื่อระหัสต่างๆของข้อมูลที่ได้รับจากตัวคอนโทรลเลอร์ในช่อง editing tag file แล้วผลที่ได้จะแสดงอยู่บนตาราง TAG tableในจอภาพ ดังนี้ ดังรูปที่ 5-26



รูปที่ 5-25 *แสดงเมนูย่อยของเมนูหลัก <MONITOR>*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

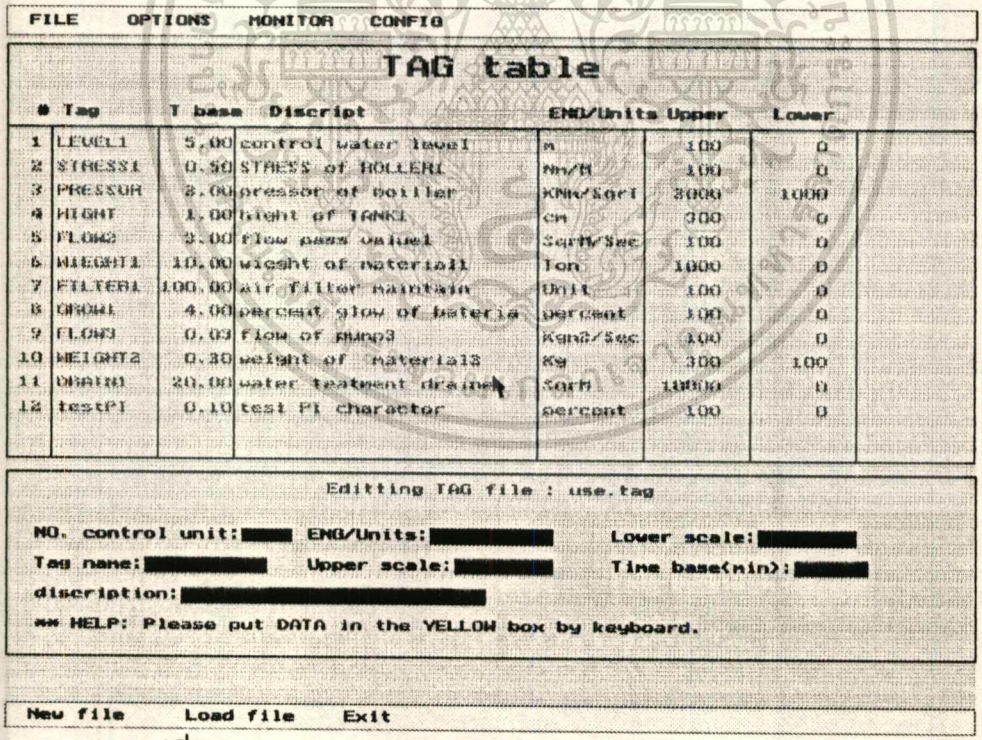
เมื่อเลือกเมนูย่อย <TAG EDITING>แล้ว บนจอภาพจะปรากฏเป็นดังรูปที่ 5-26 สังเกตได้ว่าเป็นเมนูบาร์ ซึ่งแถบสว่าง(HIGHT LIGHT) แทนที่จะอยู่ขอบบนของจอภาพกลับมาปรากฏอยู่ขอบล่างของจอภาพ ซึ่งเป็นตัวเลือก 3 ลักษณะคือ

เมนูบาร์ <New file> คือ สร้าง TAG ไฟล์ใหม่ และกำหนดพารามิเตอร์ใหม่ทั้งหมด

เมนูบาร์ <Load file> คือ เปิด TAG ไฟล์เดิม โดยนำมาแก้ไขพารามิเตอร์หรือเพิ่มเติม

เมนูบาร์ <Exit> คือ ออกจากเมนูย่อยนี้ไป เข้าสู่เมนูหลัก

ในกรณีที่เลือก <New file> ผู้ใช้ต้องใส่ชื่อไฟล์ที่ต้องการสร้างใหม่เสียก่อน แล้วจะนำเข้าสู่การแก้ไข TAG ไฟล์ ตามรูปที่ 5-26 ตาราง TAG (TAG table)จะว่างเปล่าเพราะว่าเป็น ไฟล์ซึ่งถูกสร้างขึ้นใหม่ยังจะไม่มีข้อมูล สังเกตว่าจะมีหมายเลขปรากฏทางด้านซ้ายของตารางซึ่งคือ หมายเลขของตัวคอนโทรลเลอร์ตั้งแต่ 1 ถึง 12 ซึ่งอยู่ในเครือข่ายเดียวกัน ซึ่งผู้ใช้จะทำการใส่ข้อมูลตามจริงของตัวคอนโทรลเลอร์ เช่นตั้งชื่อดังนี้



รูปที่ 5-26 *แสดงการป้อนพารามิเตอร์ของเมนูย่อย<TAG edit>*

ตัวอย่าง

ตัวคอนโทรลเลอร์หมายเลข(#) 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีชื่อ(TAG)ว่า "FLOW1"

ฐานเวลา(time base)เป็น 5.00 นาที

บรรยายลักษณะ(discription) อัตราการไหลของปั๊มพ์ตัวที่1

หน่วยทางวิศวกรรม(ENG/Unit) ลูกบาศก์เมตรต่อวินาที (m^3 /sec)

สเกลบน(upper scale) 20 m^3 /sec

สเกลล่าง(Lower scale) 0 m^3 /sec

เมื่อได้ข้อมูลของตัวดิจิทัลคอนโทรลเลอร์ตัวที่ 1 ดังกล่าวข้างต้นแล้ว ต่อไปจะทำการป้อนพารามิเตอร์แต่ละอย่างไปตามลำดับดังนี้ สังเกตกรอบทึบสีเหลืองว่างเปล่าอยู่ตำแหน่งตามลำดับดังนี้

1. *NO. control unit* ใส่หมายเลขของหน่วยของคอนโทรลเลอร์ที่ต้องการแสดงผล ซึ่งต้องการแก้ไข(edit) โดยใส่หมายเลขได้ตั้งแต่ 0 ถึง หมายเลข 12 ซึ่งหมายเลข 0 หมายถึงไม่แก้ไขพารามิเตอร์ตัวทุกตัวที่แสดงไว้ในตาราง ตามตัวอย่างให้ใส่หมายเลข 1 เมื่อใส่หมายเลขหน่วยที่ ต้องการแล้วกดรีเทอร์นเพื่อทำงานในหัวข้อต่อไป(โดยสังเกตว่าแถบสีเหลืองเลื่อนไปอยู่บรรทัดถัดไป
2. *TAG name* เป็นหัวข้อต่อไปนี้ซึ่งเป็นการใส่ข้อมูลถัดจากใส่ข้อมูล No.control unit เสร็จเรียบร้อย คือหัวข้อ TAG name เป็นการใส่ป้ายชื่อให้กับการแสดงผลของตัวคอนโทรลเลอร์โดยใส่เป็นตัวอักษร โดยที่ป้ายชื่อนี้จะไปปรากฏอยู่ที่ขอบด้านบนของกราฟบันทึกแนวโน้มข้อมูล เพื่อให้ผู้ใช้ไม่เกิดการสับสนของการดูผลการตอบสนองของระบบที่ได้มาจากตัวคอนโทรลเลอร์ เมื่อแก้ไข TAG name เรียบร้อยแล้วกดรีเทอร์นเพื่อทำงานถัดไป ตัวอย่างด้านบนก็ใส่คำว่า "FLOW1" โดยกดคีย์ตัวอักษรเรียงทีละตัว โดยที่สามารถลบอักษรออกได้โดยกดคีย์ back space
3. *Description* เป็นการใส่คำบรรยายสั้นๆ เพื่อบ่งบอกลักษณะที่สามารถเข้าใจกิจกรรมของคอนโทรลเลอร์ได้ เช่น "FLOW OF PUMP NO.1" เป็นต้น ถ้าคิดว่าไม่จำเป็นต้องใส่คำ อธิบายไว้ก็ได้โดยกดรีเทอร์นผ่านไปทำหัวข้อต่อไป คำบรรยายที่ใส่ไว้ในช่องdescription นี้จะไม่ไปแสดงบนกราฟแต่อย่างใด ตามตัวอย่างข้างต้นให้ใส่คำว่า "Flow of pump1" เป็นต้น
4. *ENG/Unis* เป็นหัวข้อสำหรับการใส่หน่วยวัดทางวิศวกรรมเพื่อชี้บอกขนาด เช่น เมตร PSI กิโลกรัม เป็นต้น โดยที่หน่วยเหล่านี้จะนำไปแสดงบนแกน Y ของกราฟบันทึกแนวโน้มข้อมูล และบาร์กราฟ เพื่อบ่งบอกขนาดแอมพลิจูดตามจริงของกระบวนการ ตามตัวอย่างนั้นให้ใส่ "M3/Sec" เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.*Upper scale* เมื่อใส่หน่วยวัดทางวิศวกรรมไปแล้ว ต้องใส่ตัวเลข เพื่อชี้บอกขนาด (scale) ซึ่งจะไปปรากฏบนแกน Y ของกราฟบันทึกแนวโน้มข้อมูล และบาร์กราฟ โดยที่การใส่สเกลบนนี้จะใส่ค่าสูงสุดเป็นจำนวนตัวเลข 6 หลักรวมจุดทศนิยมเช่น 1500.5 เป็นต้น ซึ่งโปรแกรมนี้จะเทียบ Upper scale เป็น 100 เปอร์เซ็นต์กับ PV (process variable) เองโดยอัตโนมัติ ถ้าเป็นตามตัวอย่างก็ใส่เป็น "20" เป็นต้น

6. *Lower scale* หัวข้อนี้เหมือนกับหัวข้อที่ 5. โดยที่จะเป็นขนาดต่ำสุดโดยจะเทียบ เป็น 0 เปอร์เซ็นต์ตามโปรแกรม

7.*Time base* เป็นช่วงเวลาสิ้นสุดของแกนเวลา ซึ่งคือแกน X ที่แสดงบนกราฟ trend ดังนั้นจึงจำเป็นต้องใส่ฐานเวลาของกราฟบันทึกแนวโน้มข้อมูล ด้วย ซึ่งสามารถใส่มากที่สุดได้ 99999 นาทีหรือ 69 วัน ที่ซึ่งเกินพอสำหรับกระบวนการใดๆ โดยที่จำนวนตัวเลขที่ใส่เข้าไปจะถูก ให้เป็นหน่วยของนาทีแบบเดียว เช่นใส่ 2 จะเป็น 2 นาที หรือต้องการฐานเวลาเป็น 30 วินาทีก็ใส่ค่าเป็น 0.5 เป็นต้น ขนาดของฐานเวลาจะเป็นตัวกำหนดเวลาในการแสดงผลของกราฟ เต็มช่วงของแกน X พอดี แล้วโปรแกรมจะทำการวาดกราฟใหม่(refresh) ทันทีทันใด ตัวตัวอย่าง ใส่เลข "5" เป็นต้น

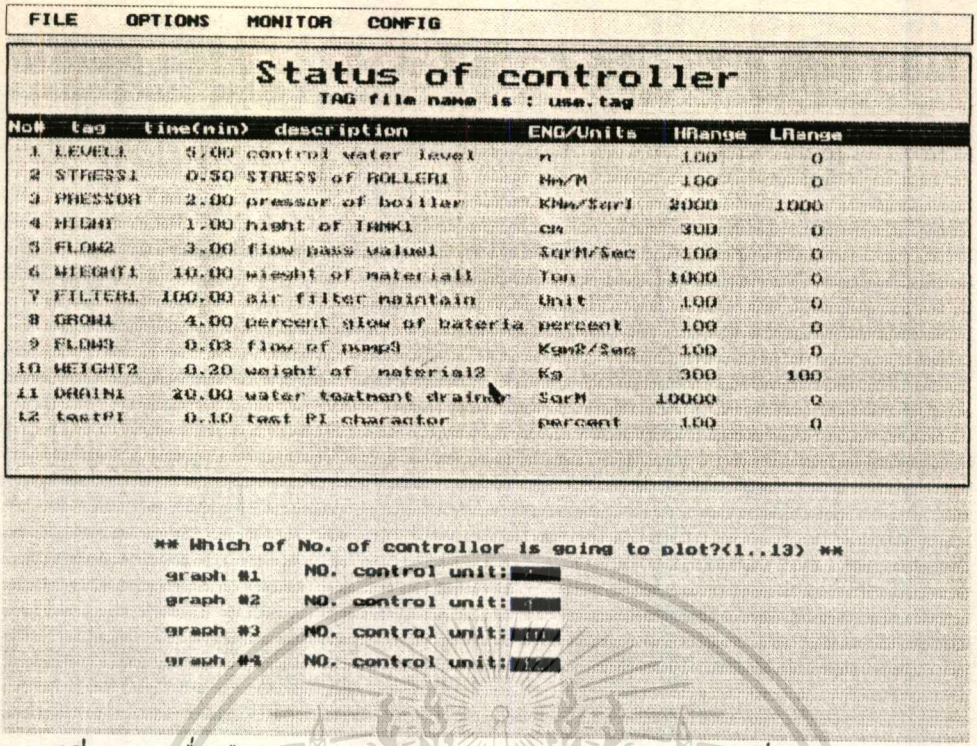
2) เมนูย่อย <PLAN MEMIC>

เป็นเมนูย่อยเพื่อแสดงภาพจำลองของพล้านต์ในโรงงานอย่างง่าย แสดงเป็นบล็อก ซึ่งโปรแกรมส่วนนี้ยังไม่เสร็จต้องดำเนินการพัฒนาต่อไป

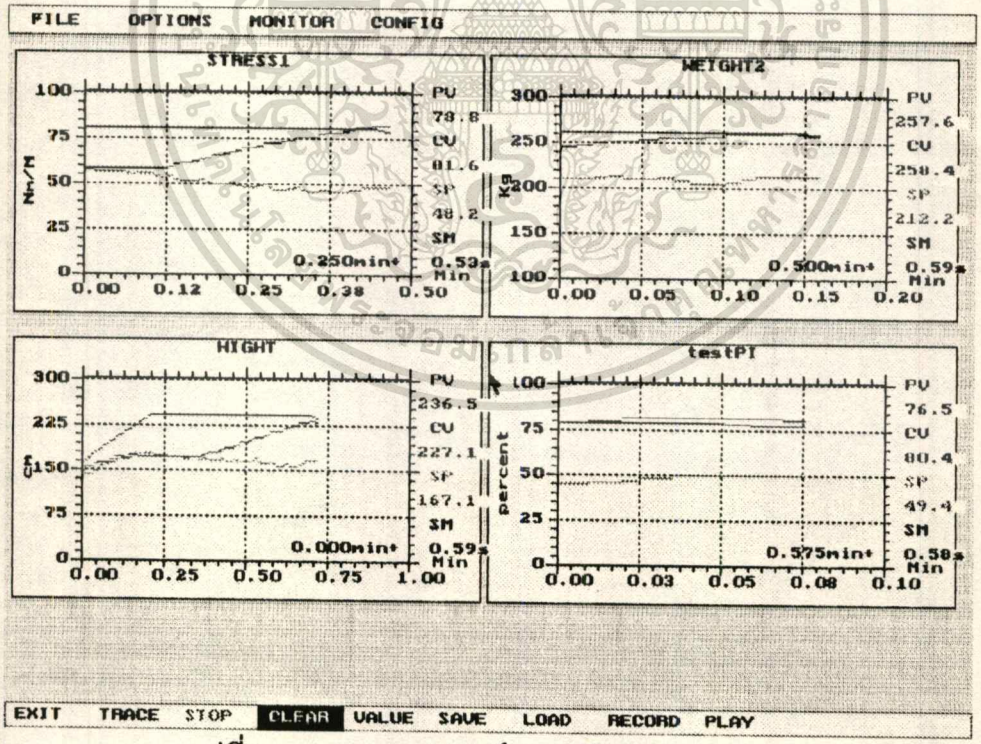
3)เมนูย่อย <TREND REC>

ชื่อเต็มคือ trend recording การที่จะเลือกทำงานในเมนู ย่อยนี้ต้องเปิด TAG ไฟล์ก่อน มิฉะนั้นจะเมนูย่อยนี้จะไม่ทำงาน โดยจำนวนกราฟเพื่อแสดงผลบนจอภาพ จะขึ้นอยู่กับการเลือกตัวแปรในเมนูย่อย <NO. GRAPH> ในเมนูหลัก<OPTIONS> เมื่อเข้าไปทำงานในเมนูย่อยนี้แล้วจอภาพจะแสดงเป็น TAG-table โดยด้านล่างของตารางจะเป็นการถามหมายเลขของยูนิตของตัวดิจิทัลคอนโทรลเลอร์ที่ต้องการแสดงผลให้ผู้ใช้ใส่แสดงดังรูปที่ 5-27 ซึ่งถ้าแสดงผลจำนวน 4 กราฟบนจอภาพก็จะถาม 4 ครั้งเป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-27 *เมื่อเลือก เมนู <trend rec> จอภาพจะแสดงดังรูป เพื่อให้ใส่หมายเลขตัวดิจิทัลคอนโทรลเลอร์ที่ต้องการแสดงผล*



รูปที่ 5-28 *แสดงกราฟบันทึกแนวโน้มข้อมูล 4 กราฟ*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเรียบร้อยแล้วจอภาพจะเปลี่ยนการแสดงผลเป็นกราฟ แต่ยังไม่แสดงผลกราฟทันทีแต่ต้องเลือกเมนูบาร์ซึ่งแสดงไว้ด้านล่างของจอภาพ ดังรูปที่ 5-28 แสดงเป็นกราฟบันทึกแนวโน้มข้อมูล 4 กราฟ และมีเมนูเลือกการทำงานต่อไปทางด้านล่าง เป็นต้น

เมนูบาร์<EXIT> คือการออกจากเมนูย่อยของ TREND REC เข้าสู่เมนูหลัก(main menu)เพื่อที่ผู้ใช้ต้องการใช้เมนูย่อยอื่น เช่น <BAR> เป็นต้น

เมนูบาร์ <TRACE> คือการพล็อตกราฟการตอบสนองของระบบ ซึ่งได้มาจากตัวคอนโทรลเลอร์ หมายเลขที่กำหนด โดยกราฟบันทึกแนวโน้มข้อมูล จะประกอบด้วยเส้นกราฟ 3 เส้นด้วยกันเส้นสีเขียวคือ PV (process variable) เส้นสีฟ้าคือ CV(control variable) และเส้นสีแดงคือ SP(setpoint) โดยที่เส้นกราฟจะถูกพล็อตไปจนกระทั่งสุดฐานเวลา(หรือเวลาสูงสุดของแกน X) และเส้นกราฟทั้งหมด จะถูกเลื่อนไปด้านหน้าเพื่อที่จะดำเนินการพล็อตเส้นกราฟต่อไป ซึ่งเวลาแซมปลิงซึ่งถูกกำหนดค่าในเมนูย่อย<SET SAMPLING> จะมีผลต่อความละเอียดของเส้นกราฟด้วย

เมนูบาร์ <STOP> เป็นเมนูเพื่อหยุดการพล็อตของกราฟชั่วคราวจนกระทั่ง ใช้เมนู TRACE อีก ครั้ง อย่างไรก็ตามเส้นกราฟจะไม่ต่อเนื่องกับกราฟเดิม เพราะว่าช่วงเวลาที่ทำงานหยุดพล็อตตัวโปรแกรมจะหยุดการติดต่อกับตัว controller

เมนูบาร์ <CLEAR> เป็นเมนูสำหรับลบเส้นกราฟทั้งหมดออกจากตารางกราฟ ซึ่งเมนูนี้จะไม่ทำงานเมื่อกราฟกำลังดำเนินการพล็อตอยู่แต่จะทำงานได้เมื่อใช้เมนูย่อย STOP เสียก่อน

เมนูบาร์ <VALUE> เป็นเมนูสำหรับใช้อ่านค่ากราฟ ขณะเวลาใดๆ ออกมาปรากฏเป็นตัวเลข โดยจะมีเส้นบอกตำแหน่งปรากฏอยู่บนเส้นกราฟ ณ ตำแหน่งเวลา ที่ซึ่งผู้ใช้ต้องการอ่านค่า

เมนูบาร์ <SAVE> เซ็ฟ(save)ค่าแนวโน้มข้อมูลที่ปรากฏบนกรอบของกราฟปัจจุบัน เป็นไฟล์ลงในดิสก์

เมนูบาร์ <LOAD> โหลดไฟล์ออกมาแสดงผล

เมนูบาร์ <RECORD> เป็นเมนูเพื่อบอกให้รู้ว่าการแสดงผลกราฟ จะถูกเซ็ฟ(save)ลงในไฟล์ตลอดเวลา

เมนูบาร์ <PLAY> เรียกข้อมูลซึ่งถูกบันทึก(record) จากไฟล์ออกมาดู

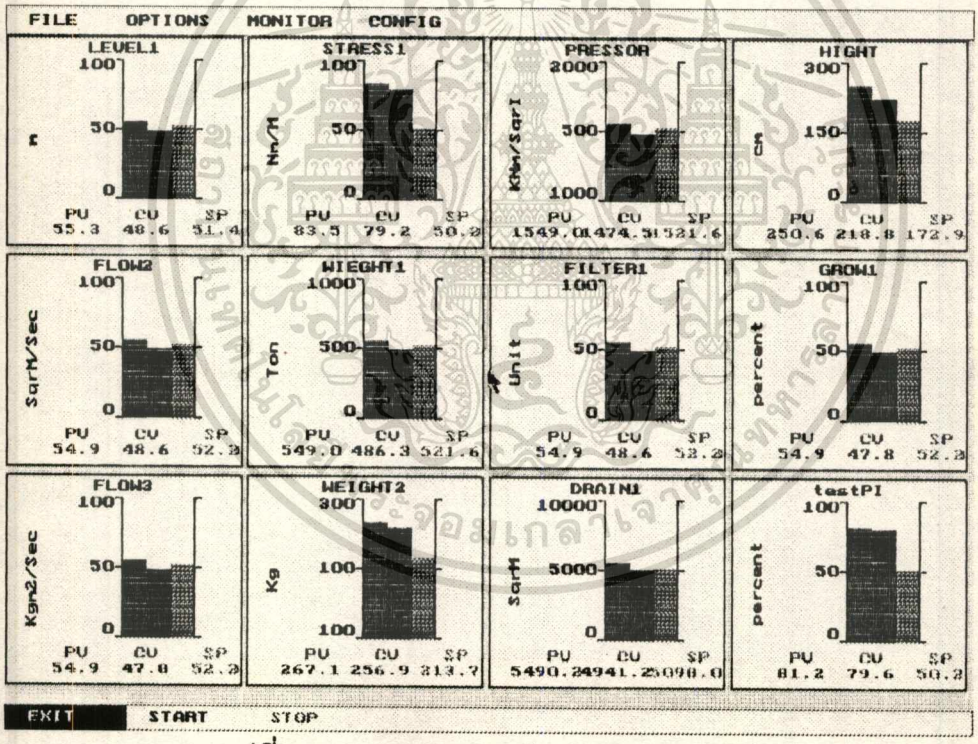
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4)เมนูย่อย <BAR>

เป็นเมนูย่อยสำหรับแสดงผลของการตอบสนอง(response)ของระบบโดย แสดงในลักษณะกราฟแท่งตามแนวตั้ง(vertical bar graph) ซึ่งจะแสดงกราฟแท่งจำนวน 12 กราฟบนจอภาพ โดยที่จะแสดงผลขณะเดียวพร้อมกันหมด 12 กราฟ ในแต่ละกราฟจะประกอบด้วย แท่งกราฟสามแท่งแบ่งเป็น 3 สี เช่นสีเขียวแทน PV สีฟ้าแทน CV และสีแดงแทน SP โดยที่ความสูงของแต่ละแท่งคือขนาด(magnitude)ของตัวแปร เมื่อเข้าเมนู BAR GRAPH เริ่มแรก จะไม่ทำงานซึ่งจะต้องเลือกการทำงานตามเมนูบารเพิ่มเติมมาอยู่ด้านล่าง ตามรูป 5-29 ดังนี้

เมนูบาร์<EXIT> ออกจากเมนูย่อย<BAR> เข้าสู่เมนูหลัก

เมนูบาร์<START> เริ่มทำงานสร้างแท่งกราฟ PV CV และ SP ขึ้นมาโดยกราฟจะมีการเปลี่ยนแปลงการตอบสนองตามระบบ



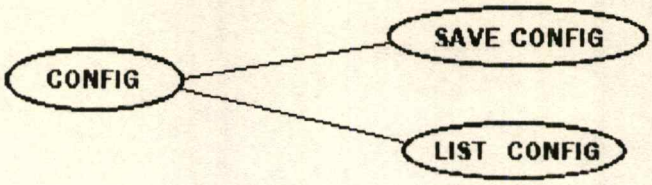
รูปที่ 5-29 *แสดงกราฟบาร์ของเมนูย่อย<BAR>*

เมนูบาร์<STOP> การทำงานซึ่ง จะทำให้แท่งกราฟมีขนาดความสูงคงที่ ณ เวลาสุดท้าย ที่หยุด ซึ่งสามารถเริ่มต้นใหม่โดยใช้เมนูบาร์<START>

5.5.7 เมนูหลัก <CONFIG>

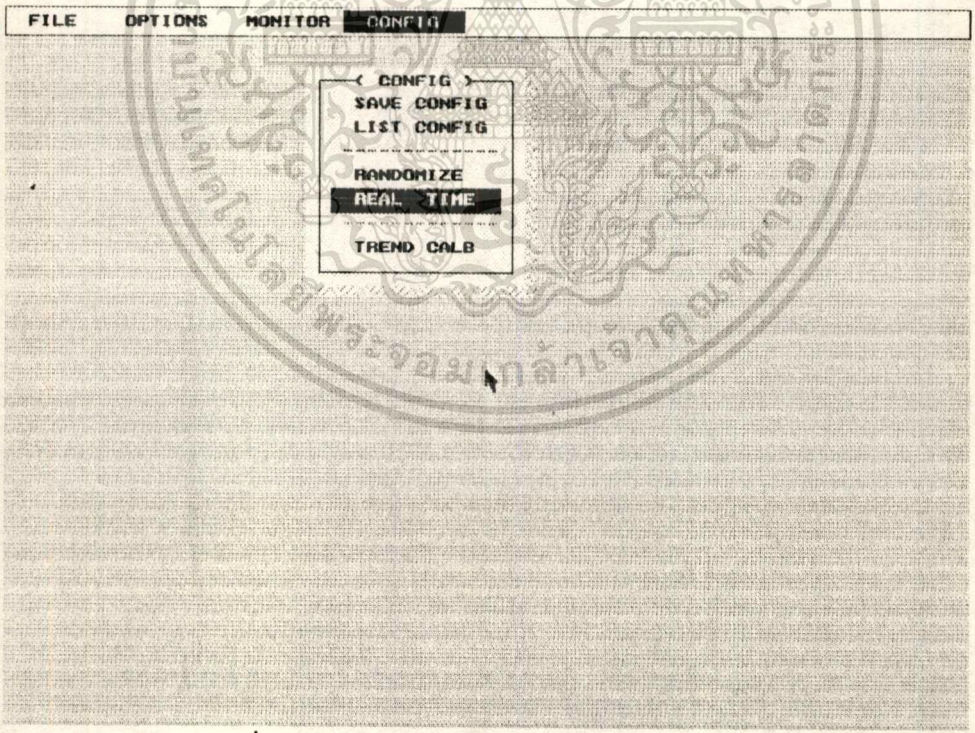
ในเมนูหลักนี้ ใช้เกี่ยวกับการแสดงค่าโครงแบบ(configulation)ของผู้ใช้ตั้งไว้ และเก็บค่าตัวแปรแสดงบล็อกการทำงานของเมนูย่อยดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า แสวงหากำไร ทั้งทางตรงและทางอ้อม ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เมนูย่อยของเมนูหลัก<CONFIG>แสดงตามรูปที่ 5-30

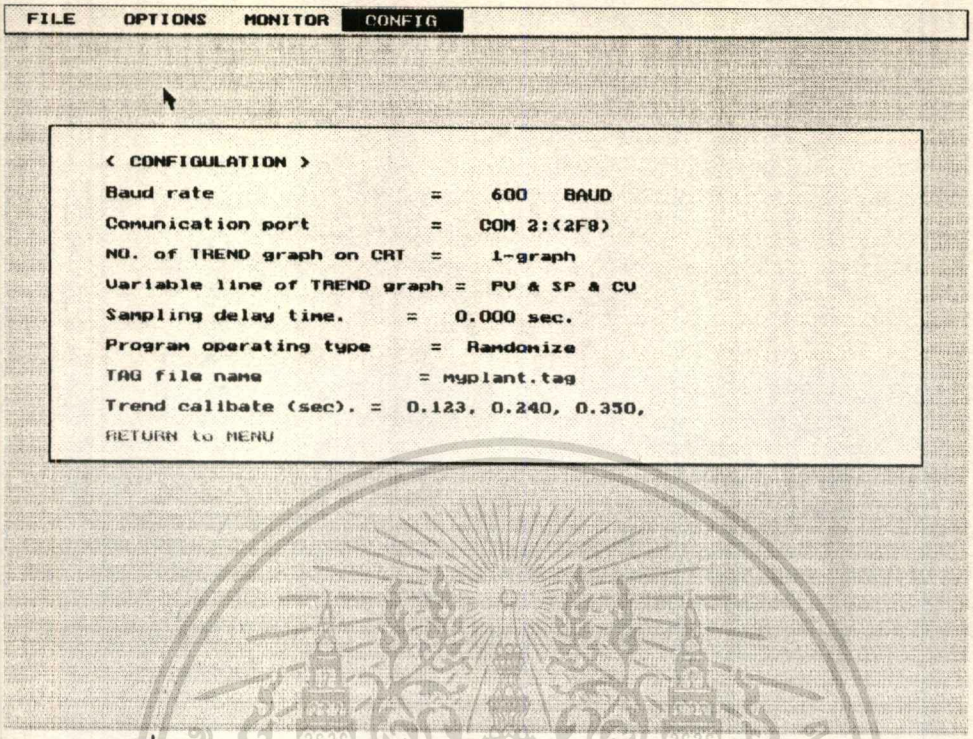
1)เมนูย่อย <SAVE CONFIG > เป็นการเก็บ(save)ค่าตัวเลือกเก็บไว้ในไฟล์ชื่อว่า plant.cfg ซึ่งโปรแกรมจะสร้างไฟล์นี้เองอัตโนมัติ โดยจะเซฟตัวเลือกในเมนู<OPTIONS>ทั้งหมด ได้แก่ ช่องการเชื่อมต่อแบบซีเรียล ความเร็วของการส่งข้อมูล(baud) จำนวนของกราฟ และค่า แซมปลิง ไฟล์ที่ถูกเก็บชื่อว่า plant.cfg จะถูกโหลดขึ้นมาใช้ขณะเริ่มต้นโปรแกรมพล้านต์มอนิเตอริง ซึ่งผู้ใช้ไม่จำเป็นต้องเซตค่าตลอดเวลาเมื่อเข้าใช้โปรแกรม



รูปที่ 5-30 *แสดงเมนูย่อยของเมนูหลัก<CONFIG>*

2)เมนูย่อย <LIST CONFIG> แสดงค่าพารามิเตอร์ปัจจุบันเช่นช่องการเชื่อมต่อ ความเร็วการ ส่งข้อมูล จำนวนกราฟ และค่าแซมปลิง แสดงดังรูปที่ 5-31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-31 *แสดงค่าโครงแบบของโปรแกรมในเมนูย่อย <list config>*

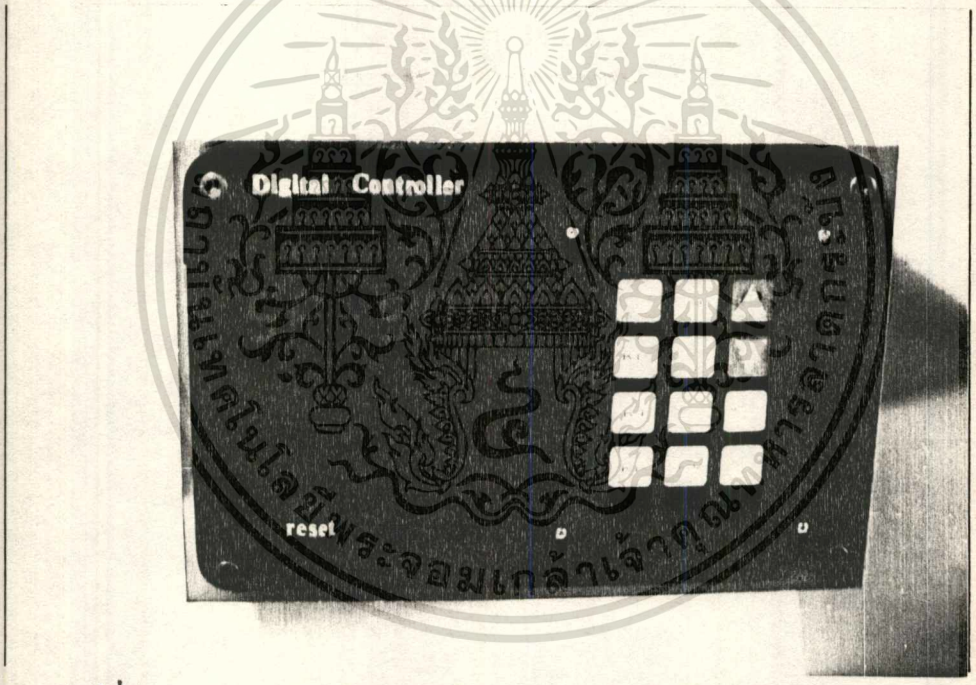
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

การทดสอบตัวดิจิทัลคอนโทรลเลอร์และโปรแกรมพล้านต์มอนิเตอริง

6.1 ลักษณะโดยทั่วไปของตัวคอนโทรลเลอร์

ด้านหน้าของตัวดิจิทัลคอนโทรลเลอร์จะมีหน้าตาดังรูป 6-1 บนแผงหน้าปัดด้านหน้าจะประกอบด้วยแผงคีย์บอร์ด จอแสดงผลแบบตัวเลข 7 ส่วน และปุ่มรีเซต มีขนาดความกว้าง 27 cm ความสูง 20 cm และความลึก 40 cm โดยมีน้ำหนัก 1.5 Kg

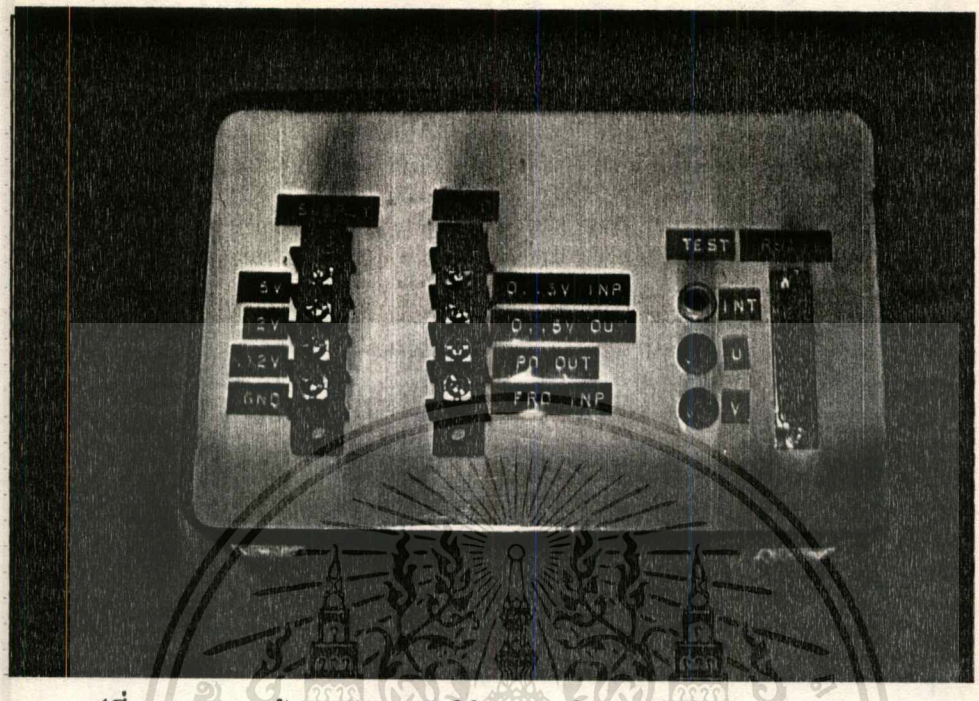


รูปที่ 6-1 *แสดงด้านหน้าของตัวดิจิทัลคอนโทรลเลอร์ต้นแบบ*

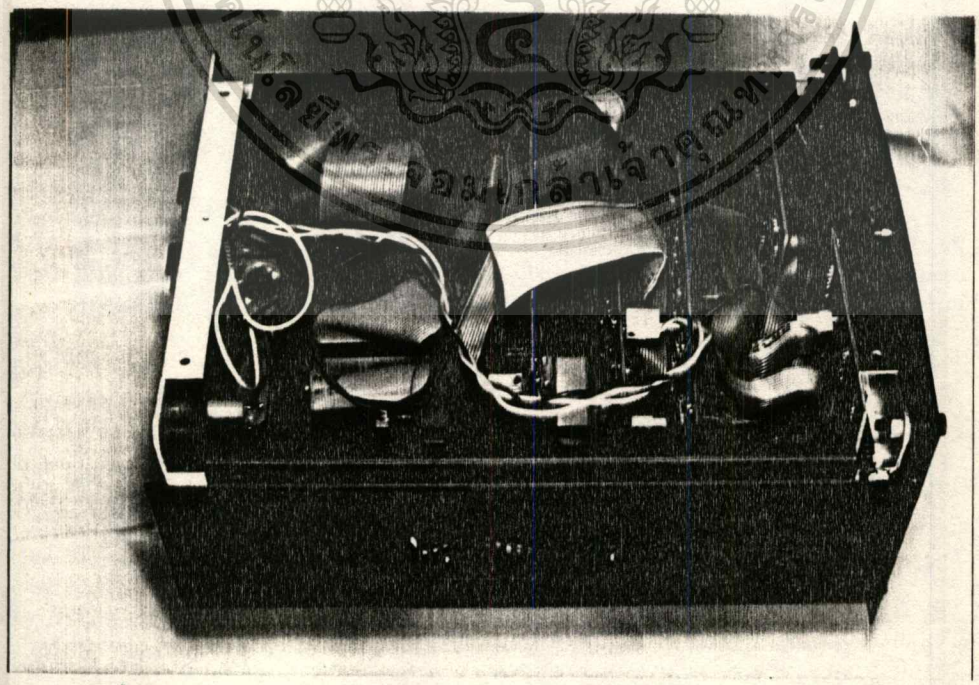
ด้านหลังของตัวดิจิทัลคอนโทรลเลอร์จะมีหน้าตาดังรูป 6-2 จะประกอบด้วยช่องการสื่อสารแบบอนุกรม RS232 มีขั้วต่ออินพุตและเอาต์พุตของตัวคอนโทรลเลอร์ทั้งในโหมดปกติ และโหมดเฟสล็อกลูป ขั้วต่อแหล่งจ่ายไฟเข้า

แสดงลักษณะภายในของดิจิทัลคอนโทรลเลอร์ตามรูป 6-3 จะประกอบด้วยแผงวงจรหลายส่วนด้วยกัน แผงวงจรที่อยู่ด้านล่างสุดคือ บอร์ดไมโครโปรเซสเซอร์เบอร์ 8031 แผงวงจรตามแนวตั้งติดด้านหน้าสุดคือ แผงวงจรควบคุมคีย์บอร์ด ต่อมา แผงควบคุมแสดงผล แผงอินพุตเอาต์พุต และแผงของวงจรเฟสล็อก

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-2 *แสดงด้านหลังของตัวดิจิทัลคอนโทรลเลอร์ต้นแบบ*



รูปที่ 6-3 *แสดงอุปกรณ์ภายในของตัวดิจิทัลคอนโทรลเลอร์ต้นแบบ*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2 ผลการทดสอบดิจิทัลคอนโทรลเลอร์ในโหมดต่าง ๆ

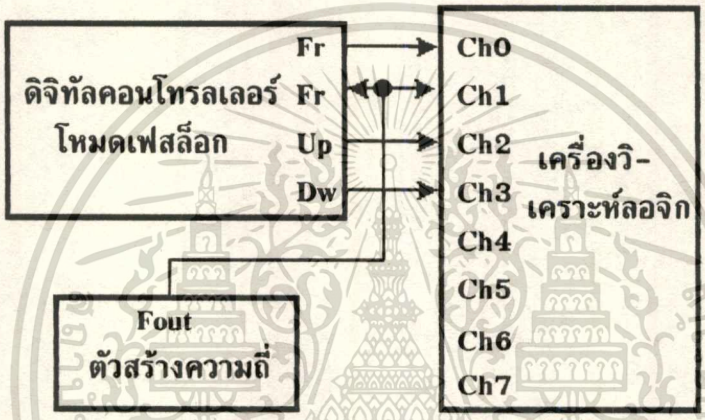
6.2.1 ทดสอบในโหมดการควบคุมแบบเฟสล็อก

-หาช่วงอินพุต ซึ่งเป็นคลื่นความถี่สี่เหลี่ยมเทียบเป็นเปอร์เซ็นต์ของ PV ดังนี้

0% PV คือความถี่ 65 เฮิรตซ์

100% PV คือความถี่ 1.5 กิโลเฮิรตซ์

-ทดสอบการทำงานลักษณะเอาต์พุตไตร-สเตตของซอฟต์แวร์เฟสดีเทกเตอร์



รูปที่ 6-4 *บล็อกไดอะแกรมเพื่อการทดสอบลักษณะเฉพาะของ PD*

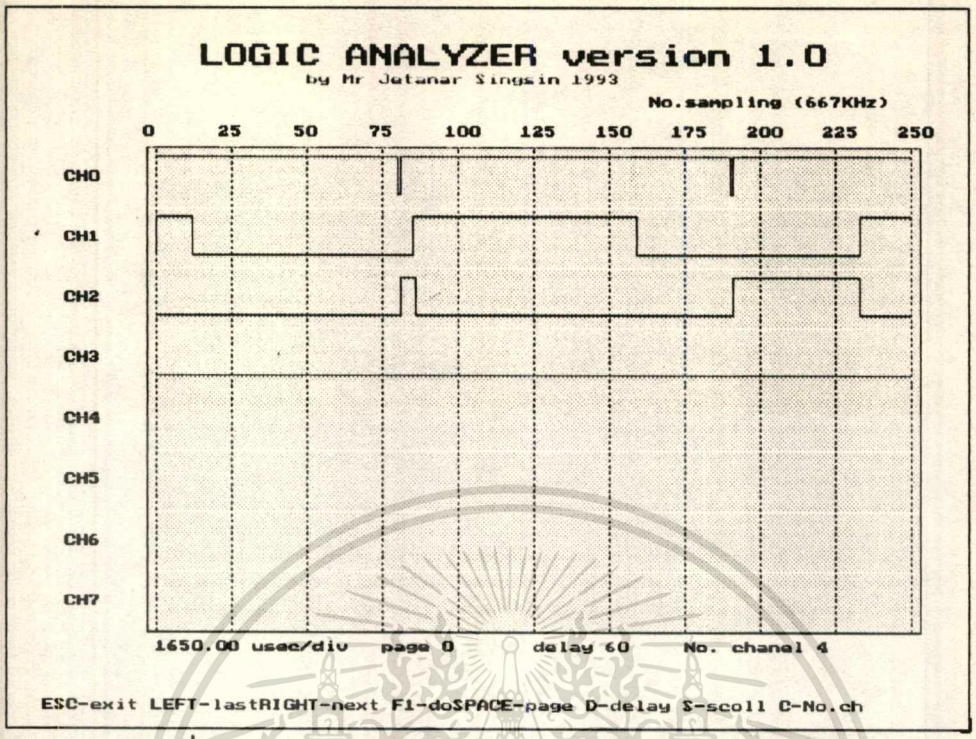
1) *ความถี่อ้างอิง F_r น้อยกว่าความถี่ป้อนกลับ F_b* ผลลัพธ์แสดงผลเป็นไดอะแกรมจังหวะเวลา(timing diagram) แสดงตามรูปที่ 6-5 โดยใช้โปรแกรมเครื่องวิเคราะห์ลอจิก ซึ่งรับค่าลอจิกภายนอก ซึ่งโปรแกรมเครื่องวิเคราะห์ลอจิกได้ถูกออกแบบขึ้นมา เพื่อใช้งานในลักษณะต่างๆซึ่งมีความเร็วไม่สูงมากนักโดยมีอัตราแซมปลิง 667 กิโลเฮิรตซ์

จากรูปที่ 6-5 แสดงได้ว่า ทางกรอบภาพด้านซ้ายมือจะแสดงการบอกช่อง ซึ่งมีด้วยกัน 8 ช่องด้วยกันคือ CH0 ถึง CH0 โดยแต่ละช่องแทนสัญญาณดังต่อไปนี้

-CH0 เป็นสัญญาณอ้างอิง หรือเป็นเซตพอย์น ซึ่งมีลักษณะเป็นพัลส์คงที่ตามเซตพอย์นที่กำหนด เช่น ที่เซตพอย์น 50 เปอร์เซ็นต์ จะเป็นความถี่พัลส์ประมาณ 3.5 กิโลเฮิรตซ์

-CH1 เป็นสัญญาณป้อนกลับ หรือสัญญาณที่นำมาเปรียบเทียบ ซึ่งในที่นี้ความถี่ของสัญญาณจะน้อยกว่าความถี่อ้างอิง และต้องมีดิวตี้ไซเคิล 50 เปอร์เซ็นต์ เพราะว่ามีผลต่อซอฟต์แวร์รูทีน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-5 *แสดงสภาวะไตร-สเตตของซอฟต์แวร์เฟสดีเทกเตอร์ ถ้า F_r น้อยกว่า F_b จากเครื่องวิเคราะห์ลอจิก*

-CH2 เป็นลักษณะเอาต์พุตของซอฟต์แวร์เฟสดีเทกเตอร์ ซึ่งเป็นคุณสมบัติของไตร-สเตต (tri-stage) ผลลัพธ์ที่ได้ตามรูปของ CH2 นั้นจะปฏิบัติได้ตามทฤษฎีของเฟสล็อกลูป ซึ่งใช้เฟสดีเทกเตอร์แบบที่ 4 ซึ่งอ้างอิงได้ตามบทที่ 2 และ บทที่ 4 ซึ่งเมื่อเปรียบเทียบแล้ว ผลที่ได้จะเหมือนกันทุกประการโดยที่ CH2 คือ V_D ซึ่งไดอะแกรมจังหวะเวลา(timing diagram)ที่ได้ทั้งหมด จะเป็นลอจิกศูนย์

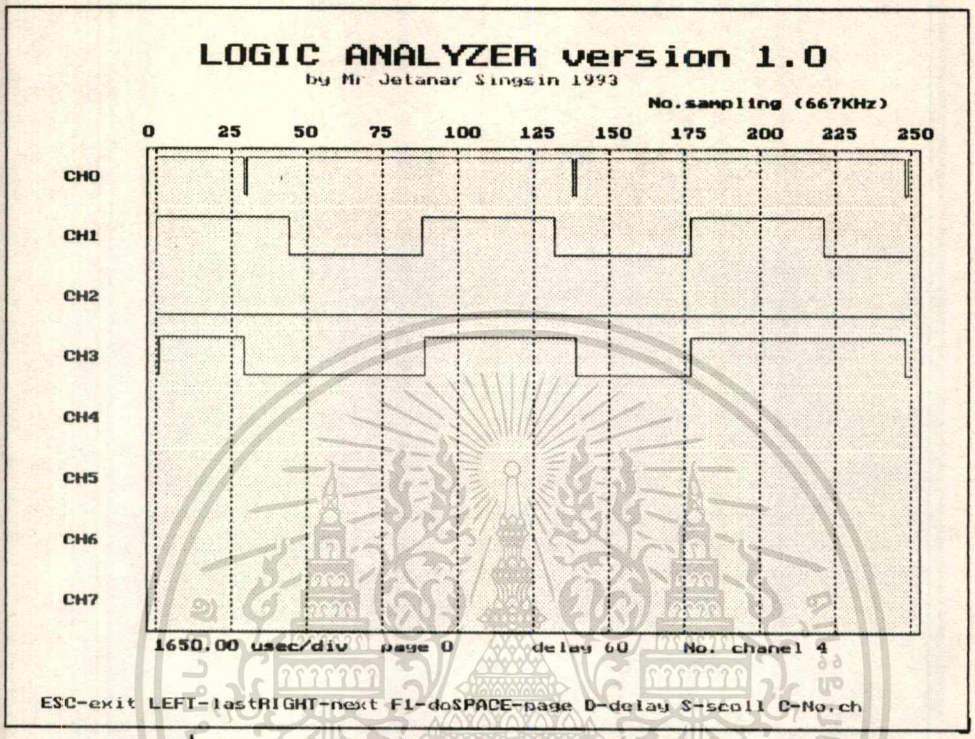
-CH3 เป็นลักษณะเอาต์พุตของซอฟต์แวร์เฟสดีเทกเตอร์ เหมือนดัง CH2 ซึ่งเมื่อเปรียบเทียบกับรูปที่ 4-16 ก็จะได้ว่า CH3 เท่ากับ V_U ตามรูปที่ 6-2 นั้น CH3 จะได้เป็นรูปคลื่นสี่เหลี่ยมซึ่งความกว้างต่างกันไป ซึ่งเป็นคุณสมบัติการเปรียบเทียบของตัวเฟสดีเทกเตอร์แบบ 4 ซึ่งรายละเอียดโปรดอยู่ในบทที่ 2

ตัวเลขที่อยู่บนกรอบแสดงหมายเลขแซมปลิงเริ่มตั้งแต่ 0 ถึง 80000 ซึ่งไม่สามารถแสดงบนจออันเดียวได้หมด โดยสามารถเลื่อนไปดูแซมปลิงต่อไป ได้โดยกดคีย์ลูกศร

และตัวเลขที่อยู่ด้านล่างของกรอบคือสเกลของเวลาเช่น 1650.00 ไมโครวินาทีต่อช่อง คือในหนึ่งช่องตามแนวตั้งที่เป็นเส้นประมีคาบเวลาเป็น 1650 ไมโครวินาทีหรือ 1.65 มิลิวินาทีต่อช่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) *ความถี่อ้างอิง F_r มากกว่าความถี่ป้อนกลับ F_b* ผลลัพธ์ที่แสดงไดอะแกรมจังหวะเวลา(timing diagram) โดยโปรแกรมเครื่องวิเคราะห์ลอจิก แสดงตามรูปที่ 6-6



รูปที่ 6-6 *แสดงสภาวะไตร-สเตตของซอฟต์แวร์เฟสดีเทกเตอร์ ถ้า F_r มากกว่า F_b จากเครื่องวิเคราะห์ลอจิก*

จากรูป 6-6 แสดงได้ว่า ทางกรอบด้านซ้ายมือจะแสดงการบอกช่อง ซึ่งมีด้วยกัน 8 ช่อง ด้วยกันคือ CH0 ถึง CH0 โดยแต่ละช่องแทนสัญญาณดังต่อไปนี้

-CH0 เป็นสัญญาณอ้างอิง หรือเป็นเซตพอย์นซึ่งมีลักษณะเป็นพัลส์คงที่ตามเซตพอย์นที่กำหนด เช่น ที่เซตพอย์น 50 เปอร์เซ็นต์ จะเป็นความถี่พัลส์ประมาณ 3.5 กิโลเฮิรตซ์

-CH1 เป็นสัญญาณป้อนกลับ หรือสัญญาณที่นำมาเปรียบเทียบ ซึ่งในที่นี้ความถี่ของสัญญาณจะมากกว่าความถี่อ้างอิง และต้องมีดิวตี้ไซเคิล 50 เปอร์เซ็นต์ เพราะว่ามีผลต่อซอฟต์แวร์รูทีน

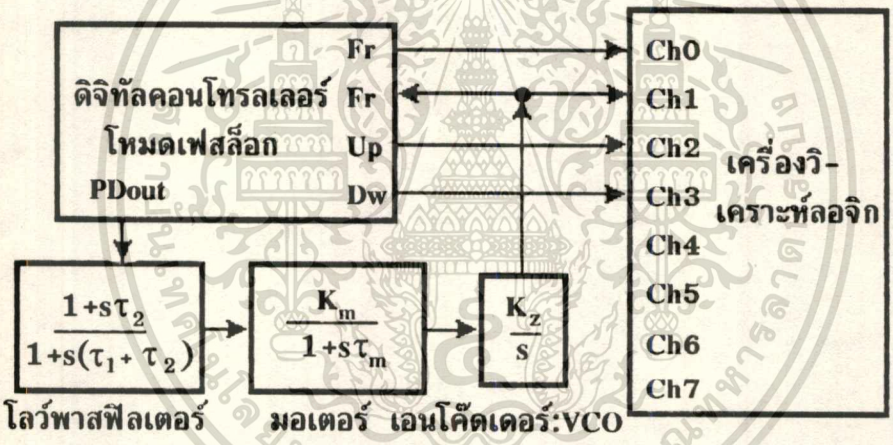
-CH2.เป็นลักษณะเอาต์พุตของซอฟต์แวร์เฟสดีเทกเตอร์ ซึ่งเมื่อเปรียบเทียบกับรูปที่ 4-16 ก็จะได้ว่า CH3 เท่ากับ V_D ตามรูปที่ 6-6 นั้น CH2 จะได้เป็นรูปคลื่นสี่เหลี่ยมซึ่งความกว้างต่างกันไปซึ่งเป็นคุณสมบัติการเปรียบเทียบของตัวเฟสดีเทกเตอร์แบบ 4 ซึ่งรายละเอียดโปรดอยู่ในบทที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-CH3 เป็นลักษณะเอาต์พุตของซอฟต์แวร์เฟสดีเทกเตอร์ ซึ่งเป็นคุณสมบัติของไตร-สเตต (tri-stage) ผลลัพธ์ที่ได้ตามรูปของ CH2 นั้นจะปฏิบัติได้ตามทฤษฎีของเฟสล็อกลูป ซึ่งใช้เฟสดีเทกเตอร์แบบที่ 4 ซึ่งอ้างอิงได้ตามบทที่ 2 และ บทที่ 4 ซึ่งเมื่อเปรียบเทียบแล้วผลที่ได้จะเหมือนกันทุกประการโดยที่ CH3 คือ V_U ซึ่งไดอะแกรมจังหวะเวลา(timing diagram)ที่ได้ทั้งหมด จะเป็นลอจิกศูนย์

3) *ความสามารถในการล็อกความถี่ที่เซตพอย์นที่กำหนด*

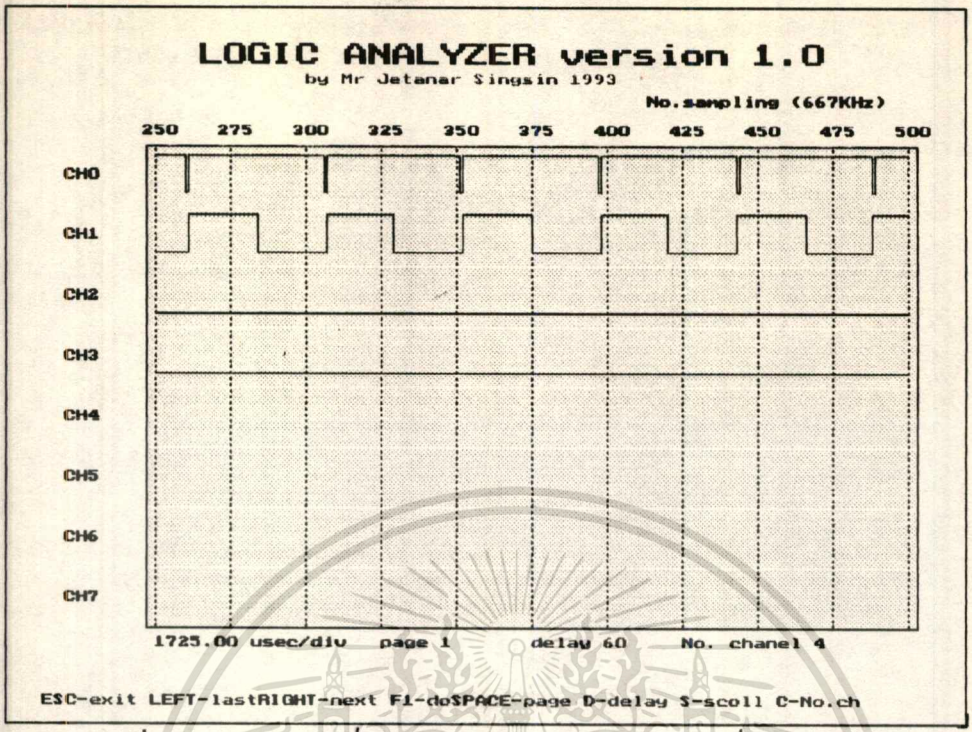
การทดสอบนี้เป็นลักษณะที่สำคัญของระบบเฟสล็อกลูป โดยการทดสอบแบบนี้จะต่อการใช้งานเป็นแบบลูปปิด โดยกำหนดให้กระบวนการเป็น VCO และ โลว์พาสฟิลเตอร์ ดังรูป 6-7 ต่อไปนี้



รูปที่ 6-7 *ไดอะแกรมทดสอบลูปปิดของระบบเพื่อทดสอบการล็อก*

การทดสอบการล็อกของตัวดิจิทัลคอนโทรลเลอร์ในโหมดเฟสล็อก ต่อวงจรการทดสอบตามรูปที่ 6-7 โดยใช้มอเตอร์ซึ่งมีฟังก์ชันถ่ายโอนดังรูป และ ออฟติคัลเอนโค๊ดเดอร์แทนโดยตัว VCO เป็นกระบวนการ ซึ่งต้องกำหนดค่าพารามิเตอร์ของโลว์พาสฟิลเตอร์เพื่อให้ระบบเสถียรภาพ ส่วน τ_m เป็นค่าคงที่เวลาของมอเตอร์ โดยที่ค่า τ_m จะเปรียบเสมือนค่าโมเมนต์ความเฉื่อยของโรเตอร์และฟลายวีล(flywheel)รวมกัน ดังนั้นจึงต้องให้ค่าที่เหมาะสมเพื่อพร้อมที่จะใช้งานจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-8 แสดงความถี่สภาวะล็อกของระบบ PLL ตามเงื่อนไขข้างต้น

CH0: ความถี่อ้างอิง CH1: ความถี่ป้อนกลับจากพล้านต์

ที่ซึ่งกำหนดเงื่อนไข

$$\tau_m = 0.33\text{sec}$$
$$\tau_1 = 0.33\text{sec}$$
$$\tau_2 = 0.39\text{sec}$$

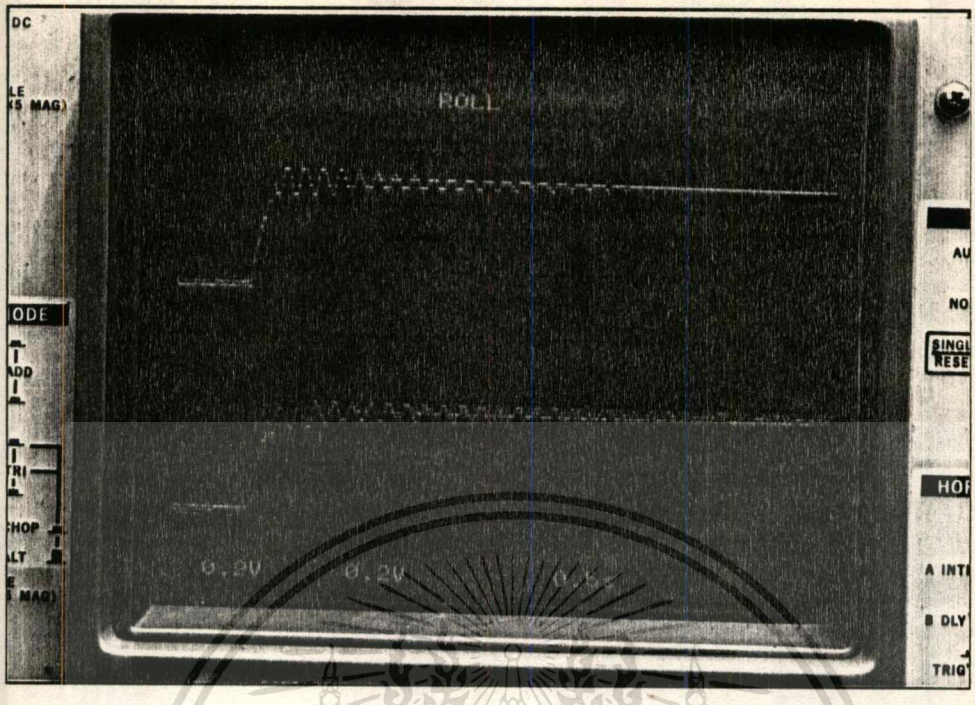
เหตุผลที่กำหนดค่าเช่นนี้ จะนำเสนอในบทวิเคราะห์

ผลการทดสอบการล็อกของความถี่อ้างอิงกับความถี่ป้อนกลับจากระบบ จะเป็นดังรูปที่ 6-8

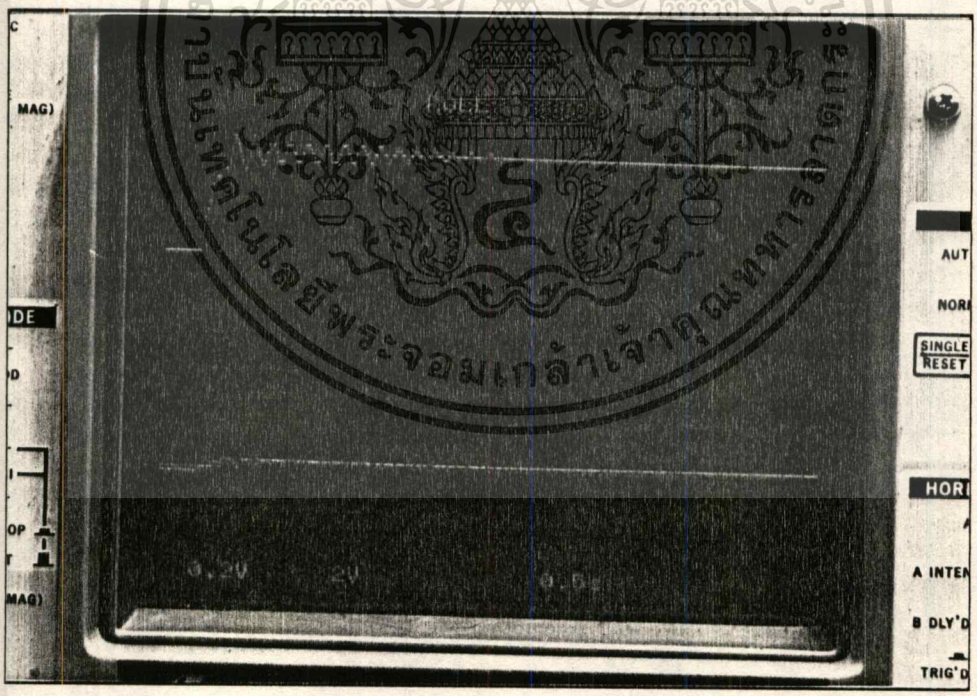
4) *ผลการตอบสนองการเปลี่ยนสเต็ปเซตพอย์น*

โดยต่อวงจรเป็นเหมือนดังรูปที่ 6-7 เพื่อทดสอบการเปลี่ยนค่าสเต็ปเซตพอย์นขนาดต่างๆ กัน เพื่อหาค่าการตอบสนอง ดังปรากฏผลตามรูปดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(a)



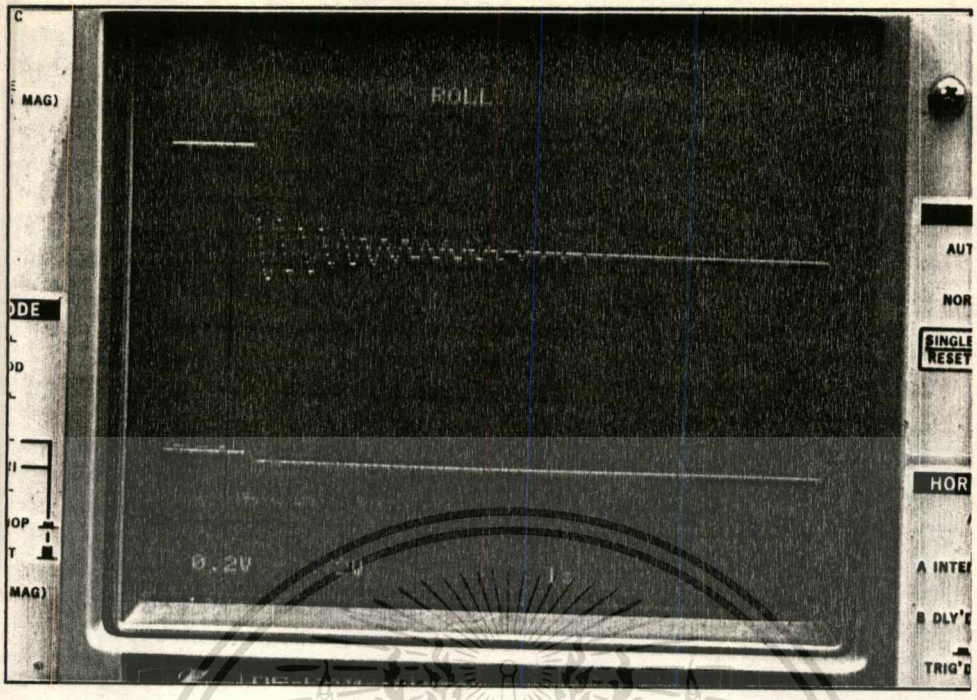
(b)

รูปที่ 6-9 *การตอบสนองสเต็ปขึ้น 0% - 50% เซตพอยน์*

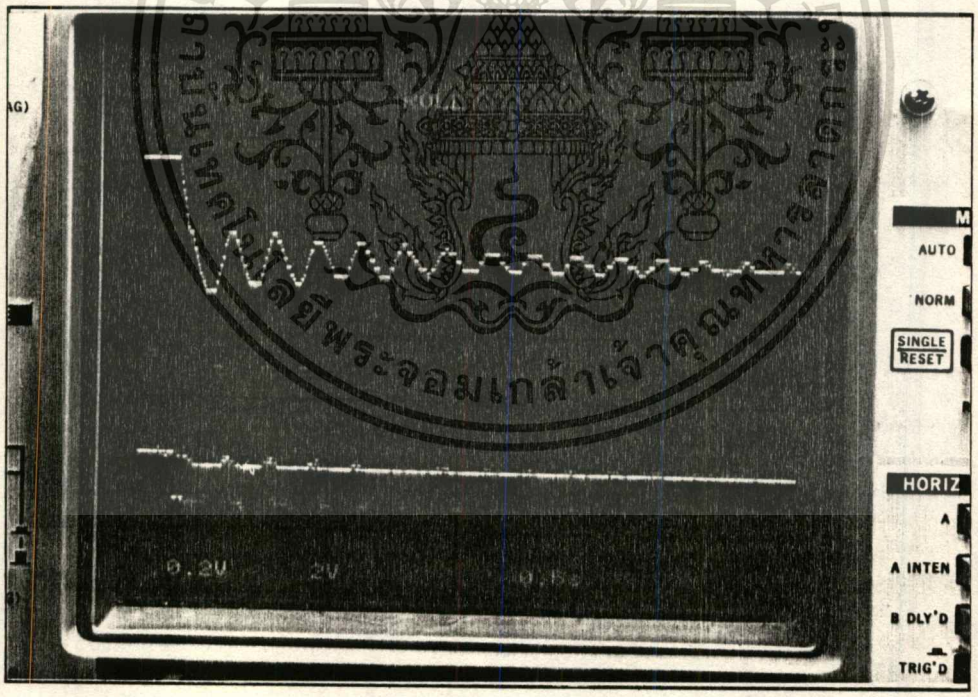
เงื่อนไข: LF = (1+0.39s)/(1+0.72s) มอเตอร์ = 1/(1+0.33s)

บน: เอาต์พุตของมอเตอร์ ล่าง: เอาต์พุตลูปฟิลเตอร์(LF)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(a)



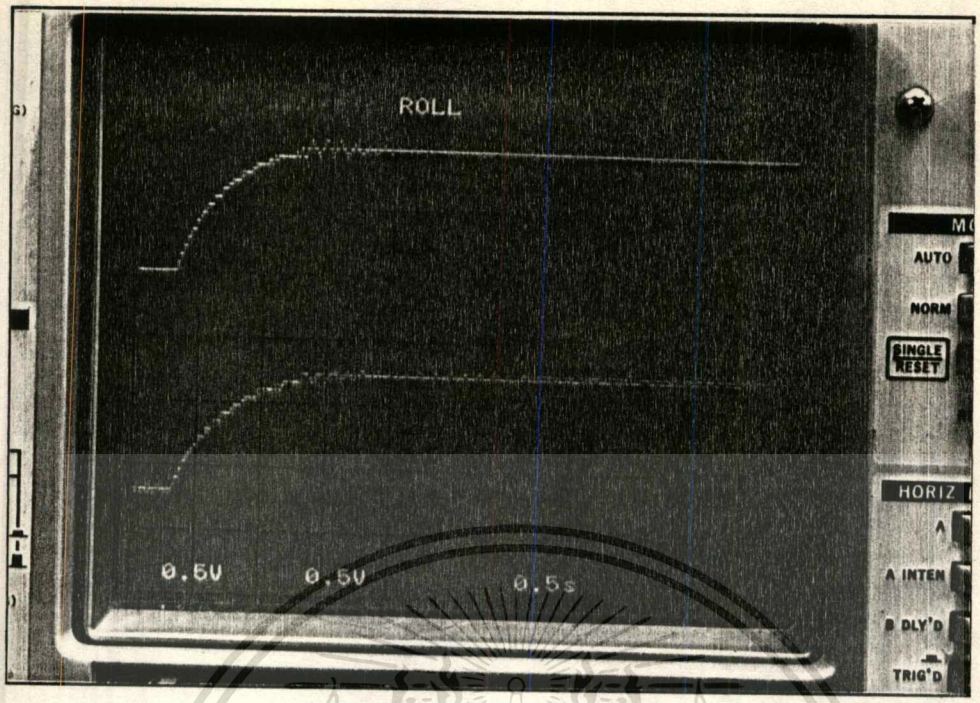
(b)

รูปที่ 6-10 *การตอบสนองสเต็ปลง 50% - 0% เซตพอยน์*

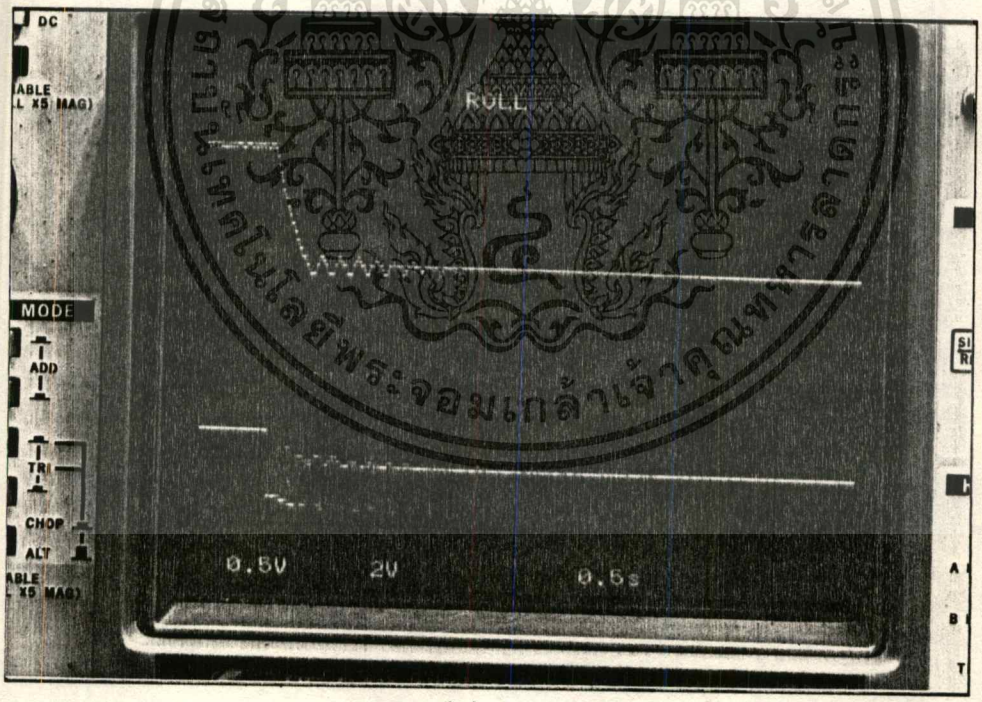
เงื่อนไข: LF = (1+0.39s)/(1+0.72s) มอเตอร์ = 1/(1+0.33s)

บน: เอาต์พุตของมอเตอร์ ล่าง: เอาต์พุตลูปฟิลเตอร์(LF)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



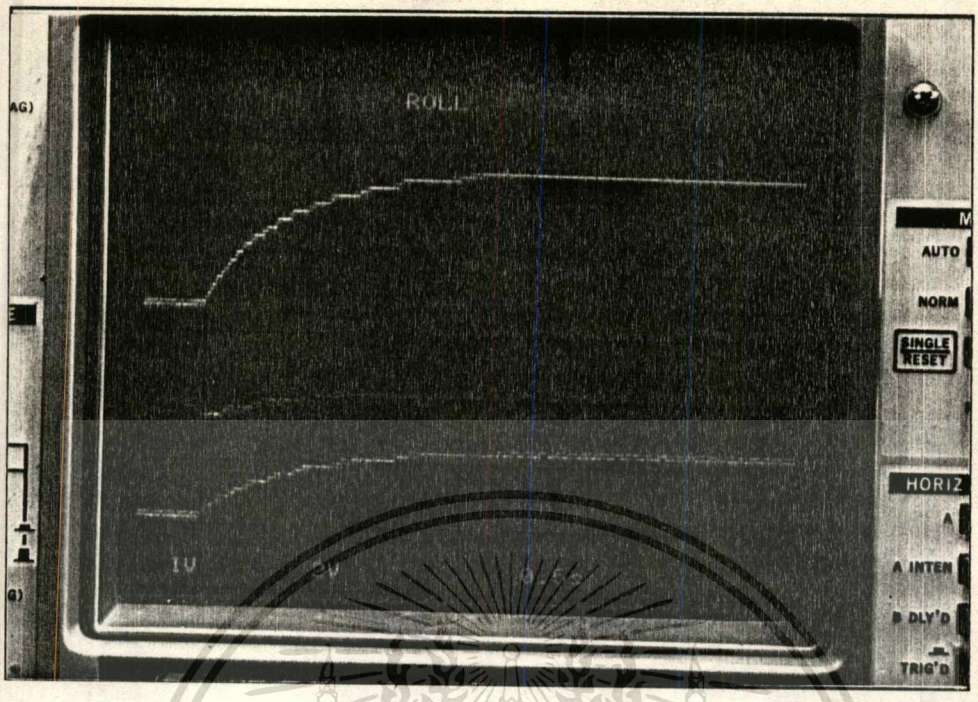
(a)



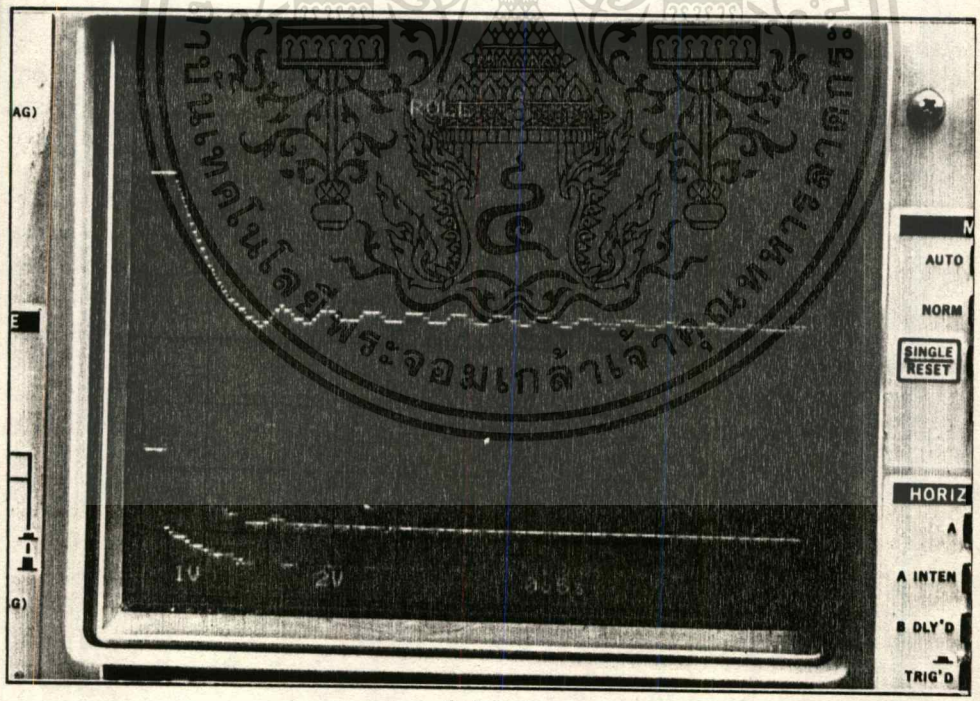
(b)

รูปที่ 6-11 *รูป (a) การตอบสนองสเต็ปขึ้น 50% - 95% เซตพอยน์*
รูป (b) การตอบสนองสเต็ปลง 95% - 50% เซตพอยน์
เงื่อนไข: LF = (1+0.39s)/(1+0.72s) มอเตอร์ = 1/(1+0.33s)
บน: เอาต์พุตของมอเตอร์ ล่าง: เอาต์พุตลูปฟิลเตอร์(LF)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



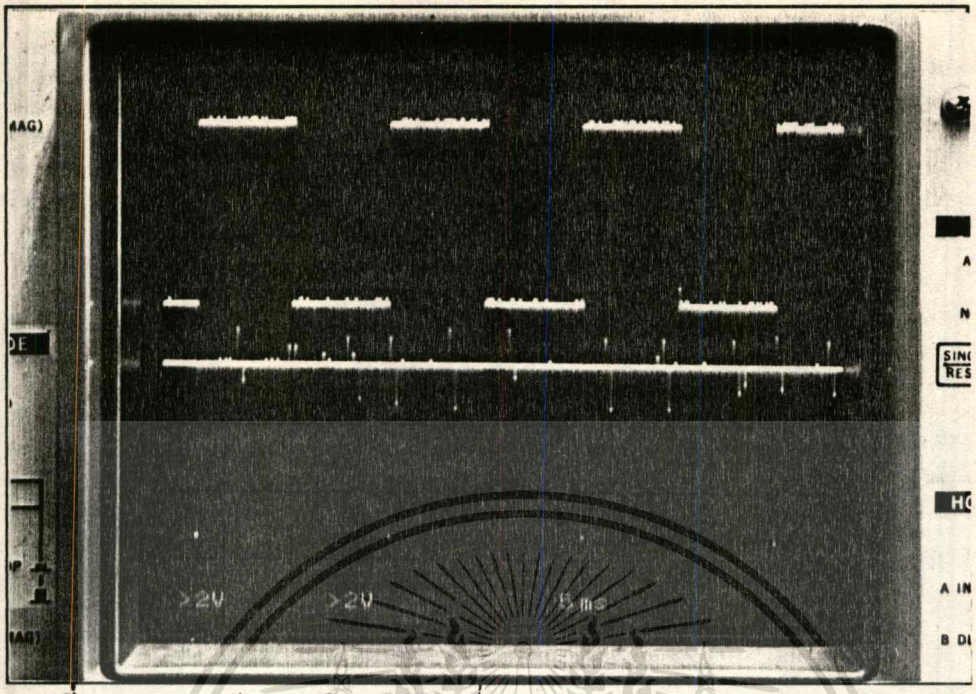
(a)



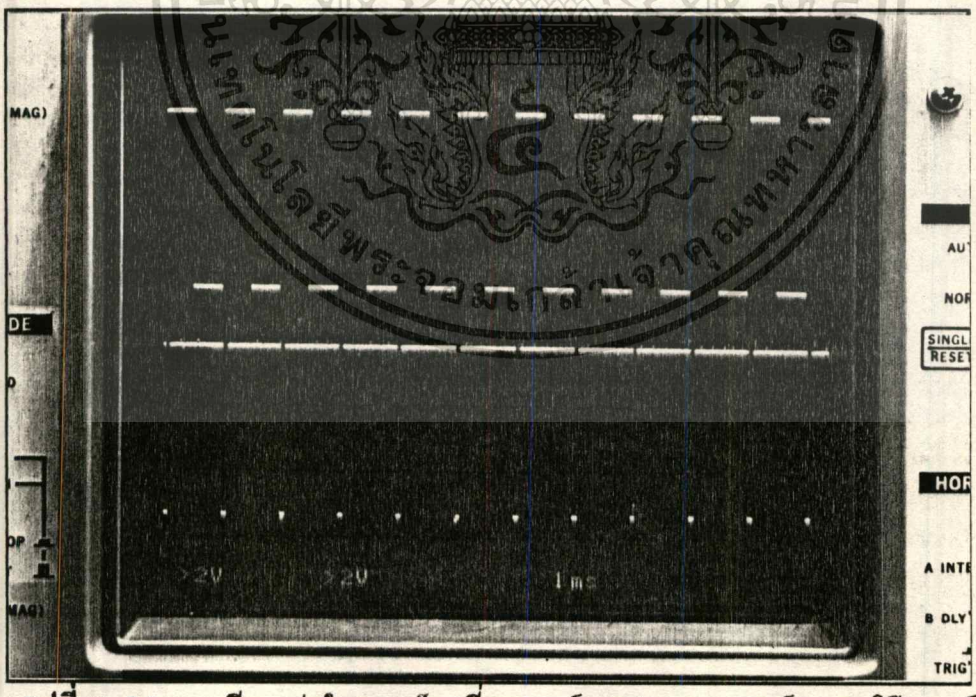
(b)

รูปที่ 6-12 *รูป (a) การตอบสนองสเต็ปขึ้น 0% - 100% เซตพอยน์*
รูป (b) การตอบสนองสเต็ปลง 100% - 0% เซตพอยน์
เงื่อนไข: LF = (1+0.39s)/(1+0.72s) มอเตอร์ = 1/(1+0.33s)
บน: เอาต์พุตของมอเตอร์ ล่าง: เอาต์พุตลูปฟิลเตอร์(LF)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-13 *เทียบเฟสในขณะล็อกที่เซตพอย์น 0% บน:เอาต์พุตเอนโค๊ดเดอร์ ล่าง: เอาต์พุตเซตพอย์น เงื่อนไข: LF = (1+0.39s)/(1+0.72s) มอเตอร์ = 1/(1+0.33s)*



รูปที่ 6-14 *เทียบเฟสในขณะล็อกที่เซตพอย์น 95% บน:เอาต์พุตเอนโค๊ดเดอร์ ล่าง: เอาต์พุตเซตพอย์น เงื่อนไข: LF = (1+0.39s)/(1+0.72s) มอเตอร์ = 1/(1+0.33s)*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กล่าวนำการควบคุมแบบทั่วไป

ฟังก์ชันถ่ายโอนของการควบคุมแบบทั่วไป โดยใช้สมการของ PID มีสมการดังนี้

$$D(s) = K_p\left(1 + s\tau_d + \frac{1}{s\tau_i}\right)$$

แต่ในตัวดิจิทัลคอนโทรลเลอร์จะไม่มีการเซตค่าตัวแปร τ_d และตัวแปร τ_i อย่างไรก็ดีสามารถใช้ค่าเป็นตัวแปร K_d และ K_i แทน โดยจะเปลี่ยนฟังก์ชันถ่ายโอนของสมการการควบคุมใหม่คือ

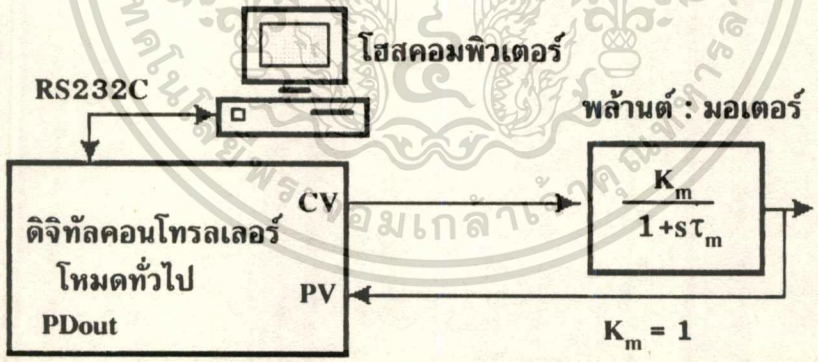
$$D(s) = K_P + sK_PK_d + \frac{K_PK_i}{s}$$

ที่ซึ่ง K_P = พรอปพรอชันแนลเกน

$K_d = 1 \bullet \tau_d$ = ดีริเวทีฟเกน

$K_i = 1/\tau_i$ = อินทีกรัลเกน

โดยทำการทดสอบตามวงจรต่อไปนี้ ดังรูปที่ 6-15



รูปที่ 6-15 ***แสดงวงจรทดสอบของตัวดิจิทัลคอนโทรลเลอร์ในโหมดทั่วไป***

6.2.2 การทดสอบการควบคุมในโหมดพรอปพรอชันแนล

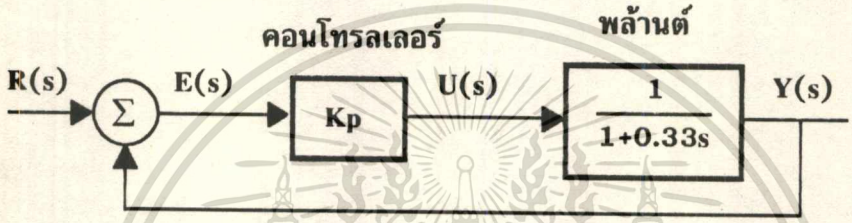
หัวข้อนี้เป็นการทดสอบโหมดการทำงานแบบ P ของดิจิทัลคอนโทรลเลอร์ โดยใช้ซอฟต์แวร์เลียนแบบการทำงาน ซึ่งรูทีนสำคัญของอัลกอลิธึมของโหมดนี้คือ รูทีน P_control โดยที่จะใช้เวลาคำนวณแต่ละครั้งจะใช้เวลา โดยวิธีการจับเวลาดังนี้

ให้ดิจิทัลคอนโทรลเลอร์ทำงานในโหมด P จำนวน 1280 วงรอบ ใช้เวลา 22 วินาที ดังนั้นในแต่ละครั้งหรือ 1 วงรอบจะใช้เวลา 17 มิลลิวินาที ซึ่งสามารถสรุปได้ว่า พล้านต์จำลอง หรือ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ห้ามมิให้ผู้ใดนำไปใช้ในการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

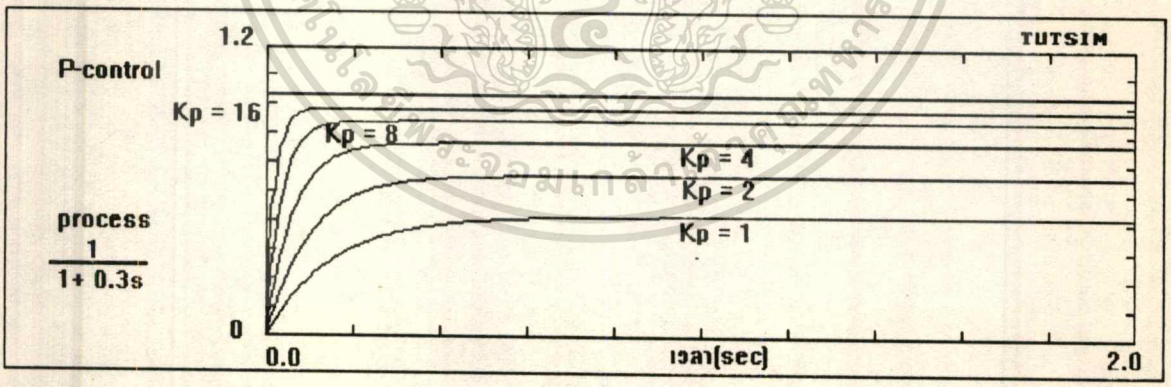
กระบวนการจำลองซึ่งใช้เป็นตัวทดสอบคอนโทรลเลอร์จะต้องมีค่าคงที่เวลามากกว่า 17 มิลลิวินาที หรือถ้าจะให้ดีจะต้องมีค่าคงที่ของเวลา มากกว่าเป็นสองเท่า คือ 34 มิลลิวินาที กล่าวได้ว่าในการใช้งานจริงของดิจิทัลคอนโทรลเลอร์ในโหมดนี้ จะมีข้อจำกัดที่สำคัญคือ ค่าคงที่ของเวลาของตัวพล้านต์ หรือระบบที่จะต้องไปควบคุมต้องมีค่ามากกว่า 34 มิลลิวินาที ในระบบอุตสาหกรรมความเป็นจริงส่วนใหญ่จะมีค่าคงที่ของเวลามีค่าสูง ดังนั้นแฟกเตอร์ของเวลาดังกล่าวจึงไม่มีผลต่อการควบคุม

- การทดสอบการตอบสนองสเต็ปในโหมด P โดยกำหนด พล้านต์จำลองเป็น ฟังก์ชันถ่ายโอนแบบออร์เดอร์หนึ่ง โดยมีบล็อกไดอะแกรมของระบบดังนี้



รูปที่ 6-16 *บล็อกไดอะแกรมของการควบคุมแบบ P โดยให้พล้านต์เป็นออร์เดอร์หนึ่ง*

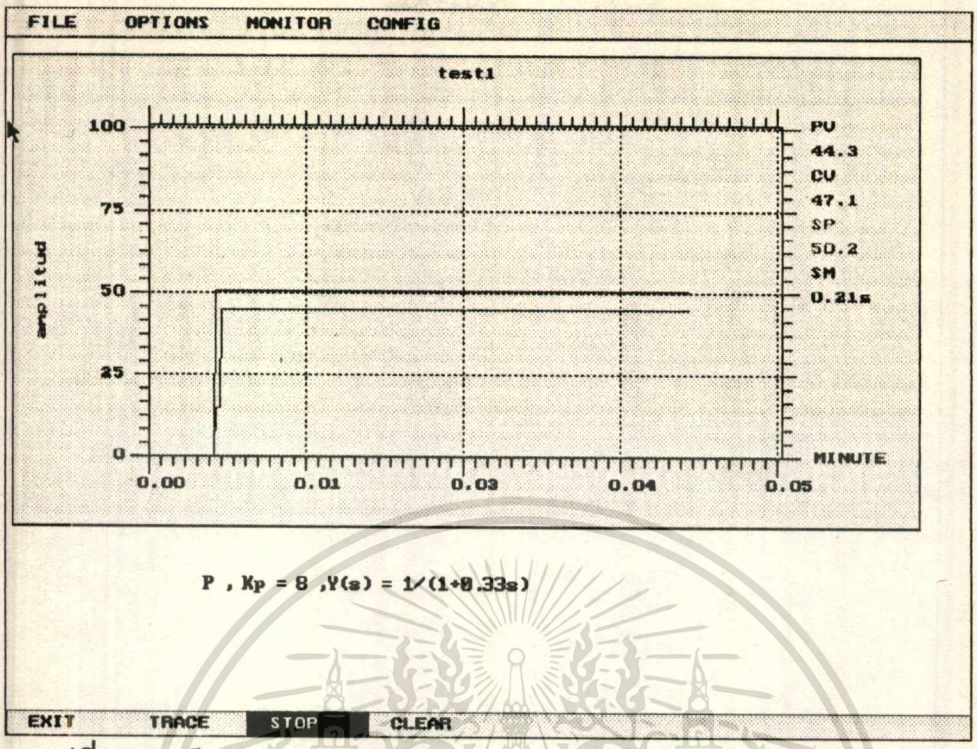
ตามบล็อกไดอะแกรมรูปที่ 6-16 ข้างต้นเราสามารถดูการซิมูเลชันของระบบ โดยใช้โปรแกรมทางคณิตศาสตร์ที่ชื่อว่า "ทุทซิม"(TUTSIM) ได้ผลการตอบสนองสเต็ปของระบบดังนี้



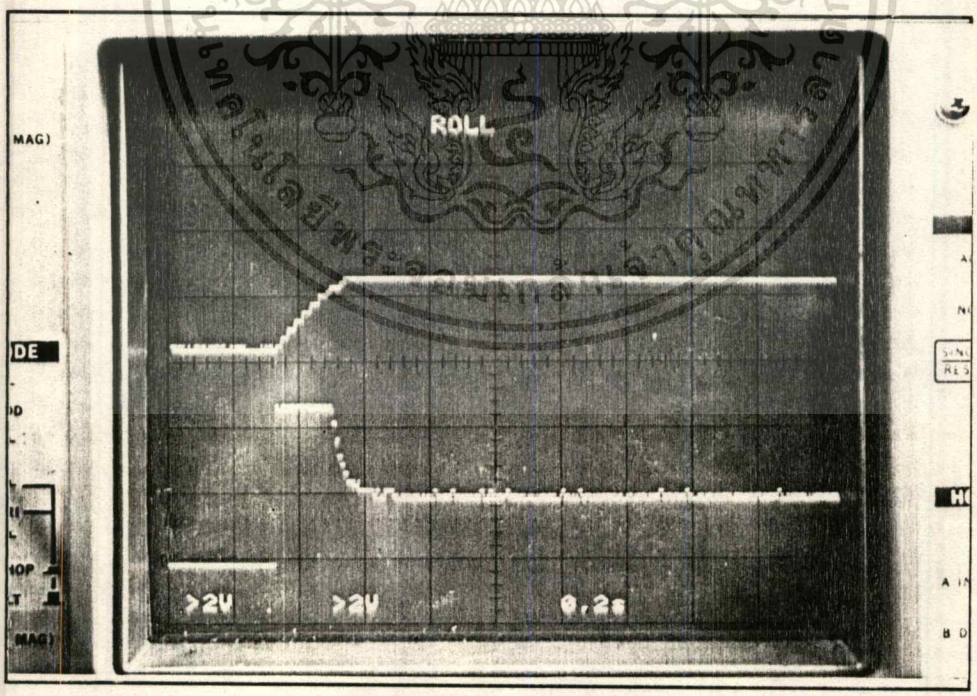
รูปที่ 6-17 *การซิมูเลชันของระบบในการควบคุมแบบ P*

จากนั้นเราทดสอบจากตัวดิจิทัลคอนโทรลเลอร์ โดยใช้พล้านต์ที่มีฟังก์ชันถ่ายโอนตัวเดียวกันคือ 1/(1+0.33s) เพื่อทดสอบว่าการตอบสนองเป็นไปตามทฤษฎีทางคณิตศาตร์หรือไม่ ซึ่งได้ผลการตอบสนองจากการสังเกตได้จากโฮสคอมพิวเตอร์ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-18 *เป็นการตอบสนองสเต็ปของการควบคุมแบบ P โดยสังเกตจากโฮส พล้านต์=1/(1+0.33s) , Kp = 8.0 บน: SP ล่าง: PV*



รูปที่ 6-19 *เป็นการตอบสนองสเต็ปของการควบคุมแบบ P โดยวัดจากสโคป พล้านต์ = 1/(1+0.33s) , Kp = 8 บน: PV ล่าง: CV*

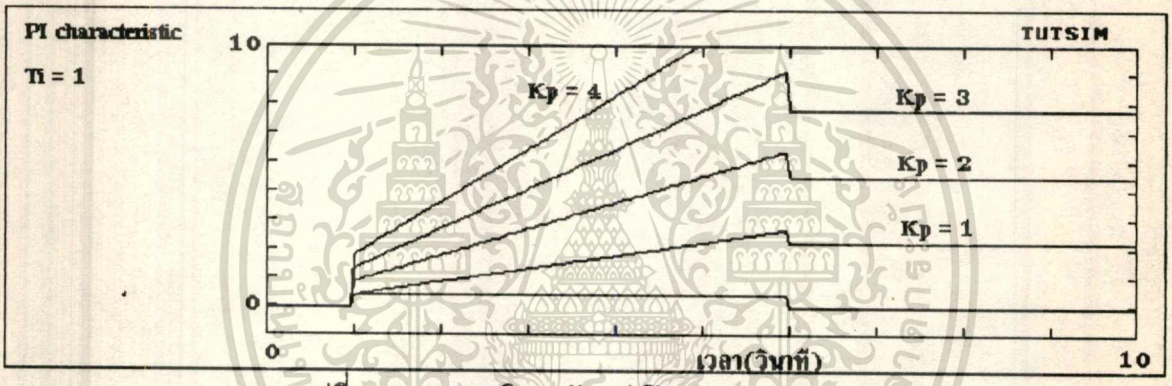
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.3 การทดสอบการควบคุมในโหมดพรอปพรอชันแนล-อินทีกรัล

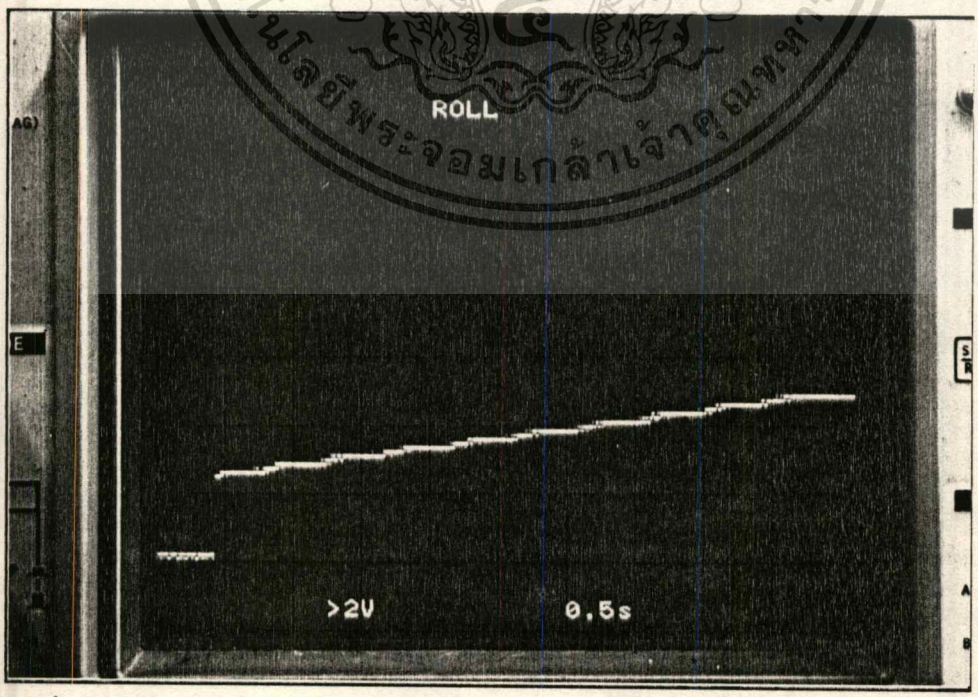
หัวข้อนี้เป็นการทดสอบโหมดการทำงานแบบ PI ของดิจิทัลคอนโทรลเลอร์ โดยใช้ซอฟต์แวร์เลียนแบบการทำงาน ซึ่งรูทีนสำคัญของอัลกอลิธึมของโหมดนี้คือ รูทีน PI_control โดยที่จะใช้เวลาคำนวณแต่ละครั้งจะใช้เวลาสิ้นเปลือง โดยวิธีการจับเวลาการทำงานดังนี้

ให้ดิจิทัลคอนโทรลเลอร์ทำงานในโหมด PI จำนวน 960 วงรอบ ใช้เวลา 19 วินาที ดังนั้นในแต่ละครั้งหรือ 1 วงรอบจะใช้เวลา 20 มิลลิวินาที ซึ่งสามารถสรุปได้ว่า พล้านต์จำลอง จะต้องมีค่าคงที่ของเวลา มากกว่าเป็นสองเท่า คือ 20 วินาที,

- การทดสอบคุณลักษณะ(characteristic) ของการควบคุมโหมด PI ในลูปเปิด โดยป้อนอินพุตสเต็ป และหาผลการตอบสนองเฉพาะของคอนโทรลเลอร์ โดยให้ PV เป็นศูนย์ โดยมีผลการตอบสนอง ตามรูป 6-20 และ รูป 6-21ดังนี้



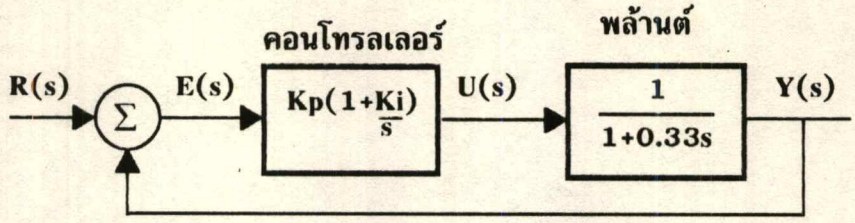
รูปที่ 6-20 *ผลการซิมูเลชันลูปเปิดของ* PI-control



รูปที่ 6-21 *เป็นการตอบสนองสเต็ป 25 % อินพุตการควบคุมแบบ PI โดยวัดจากออสซิลโลสโคป โดย Kp = 1.0 ,Ki = 0.2 และ PV = 0%*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การทดสอบการตอบสนองสเต็ปในโหมด PI โดยกำหนด พล้านต์จำลองเป็น ฟังก์ชันถ่ายโอนเป็นแบบออร์เดอร์หนึ่งเท่ากับ 1/(1+0.33s) โดยมีบล็อกไดอะแกรมของระบบ ตามรูป 6-22 ดังนี้

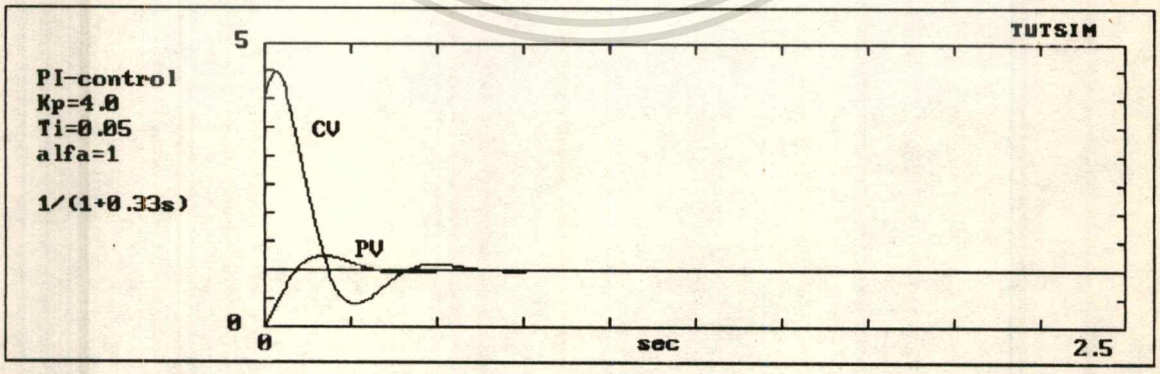


รูปที่ 6-22 *บล็อกไดอะแกรมของการควบคุมแบบ PI โดยให้พล้านต์เป็นออร์เดอร์หนึ่ง*

ตามบล็อกไดอะแกรมข้างต้นเราสามารถซิมูเลชัน ของการควบคุมแบบ PI ซึ่งใช้พล้านต์เป็นสมการออร์เดอร์หนึ่ง โดยกำหนดพารามิเตอร์แบบต่างๆ



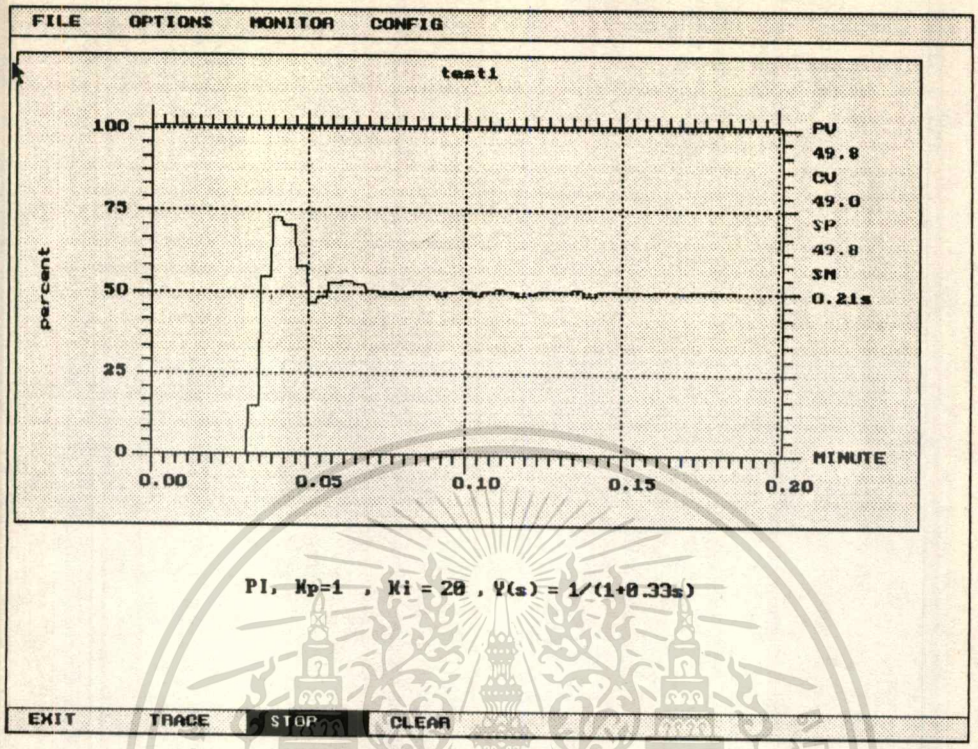
(a) Kp = 1.0 Ti = 0.05



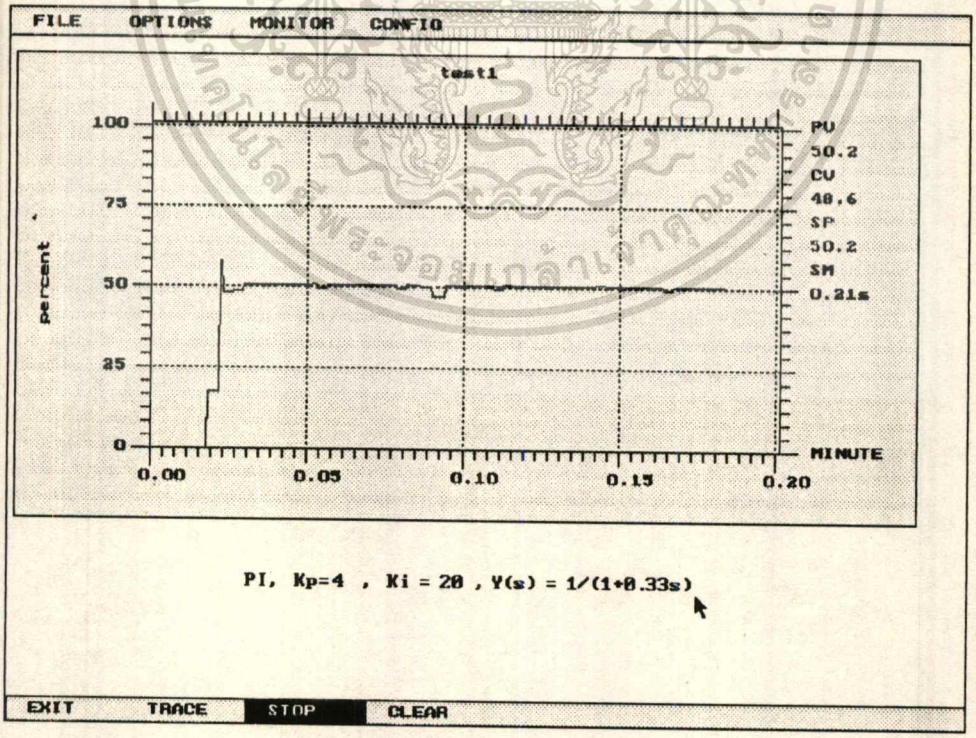
(b) Kp = 1.0 และ Kp = 4.0 Ti = 0.05

รูปที่ 6-23 (a),(b),*ผลการซิมูเลชันโดยการควบคุมแบบ PI โดยให้พล้านต์เป็นสมการออร์เดอร์หนึ่ง ในขณะค่า Kp,Ki ต่างกัน โดยโปรแกรมคณิตศาสตร์*

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ห้ามนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



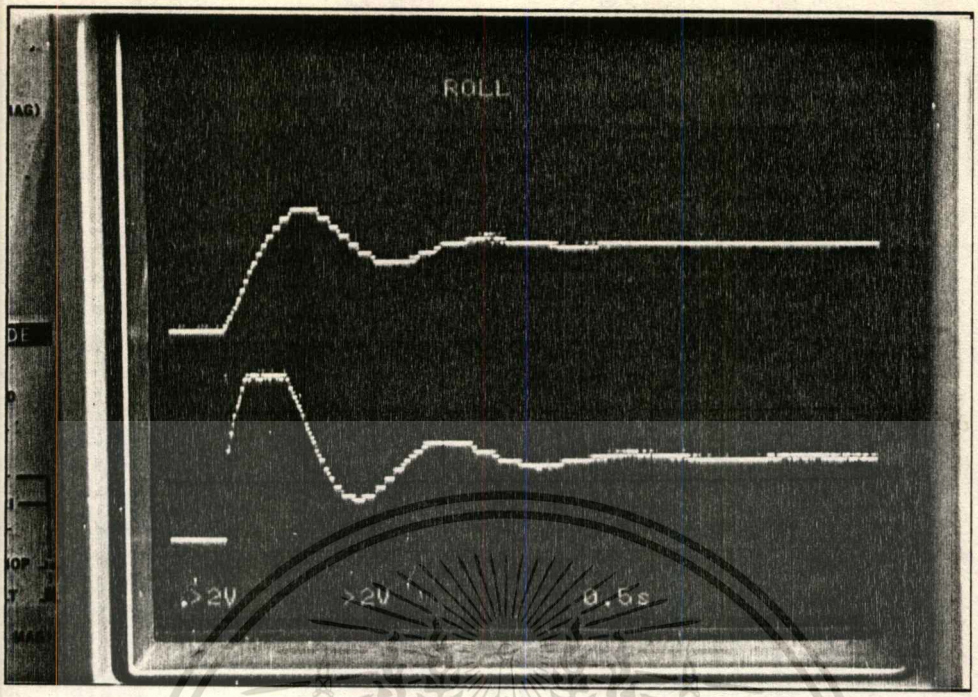
(a) Kp = 1.0 , Ki = 20



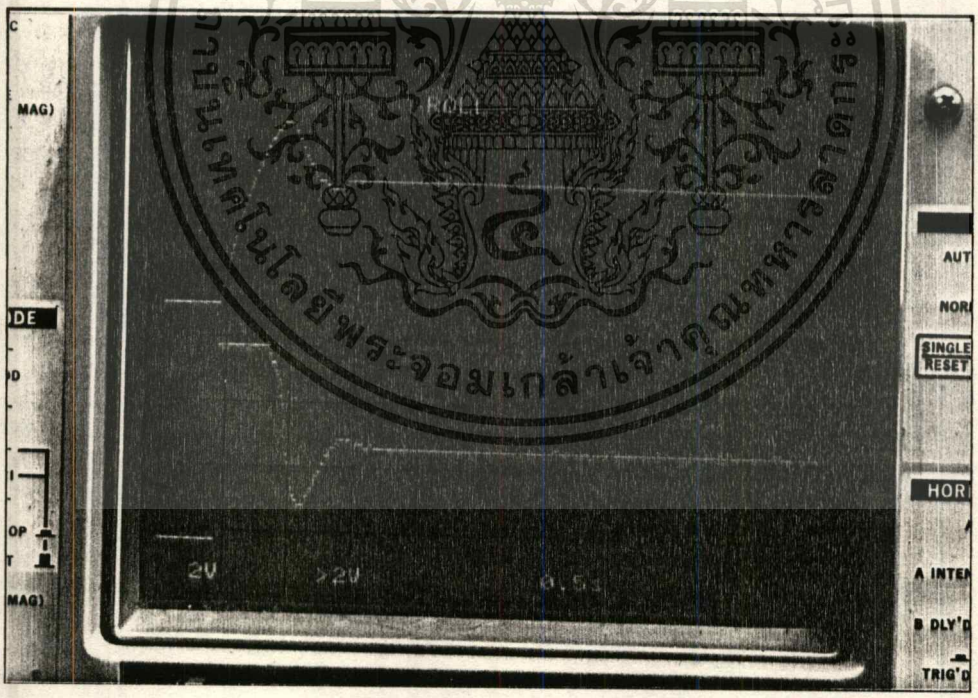
(b) Kp=4.0 , Ki = 20.0

รูปที่ 6-24 *ผลการตอบสนองสเต็ปของการควบคุมแบบ PI โดยสังเกตจากโฮส*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(a) Kp = 1.0 ,Ki = 20.0



(b) Kp = 4.0 ,Ki = 20.0

รูปที่ 6-25 *ผลการตอบสนอง 50 % สเต็ปของการควบคุมแบบ PI โดยวัดจากสโคป*

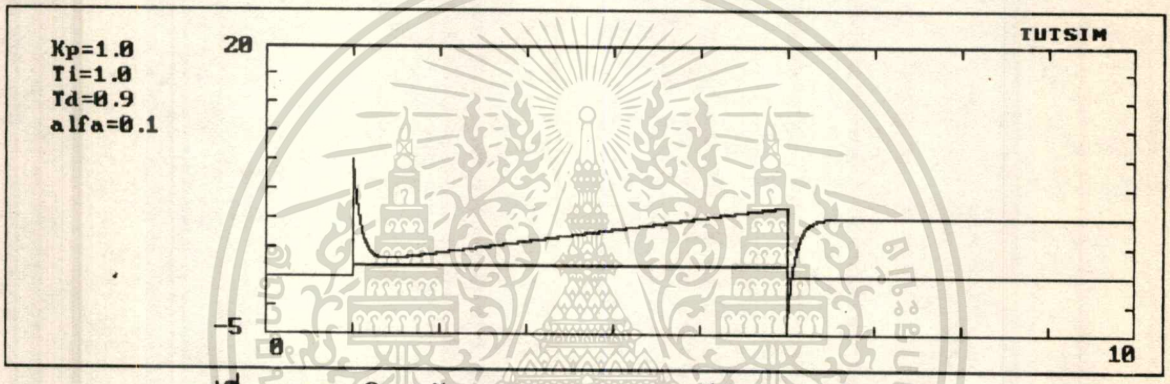
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.4 การทดสอบการควบคุมโหมดพรอปพรอชันแนล-อินทีกรัล-ดีริเวทีฟ

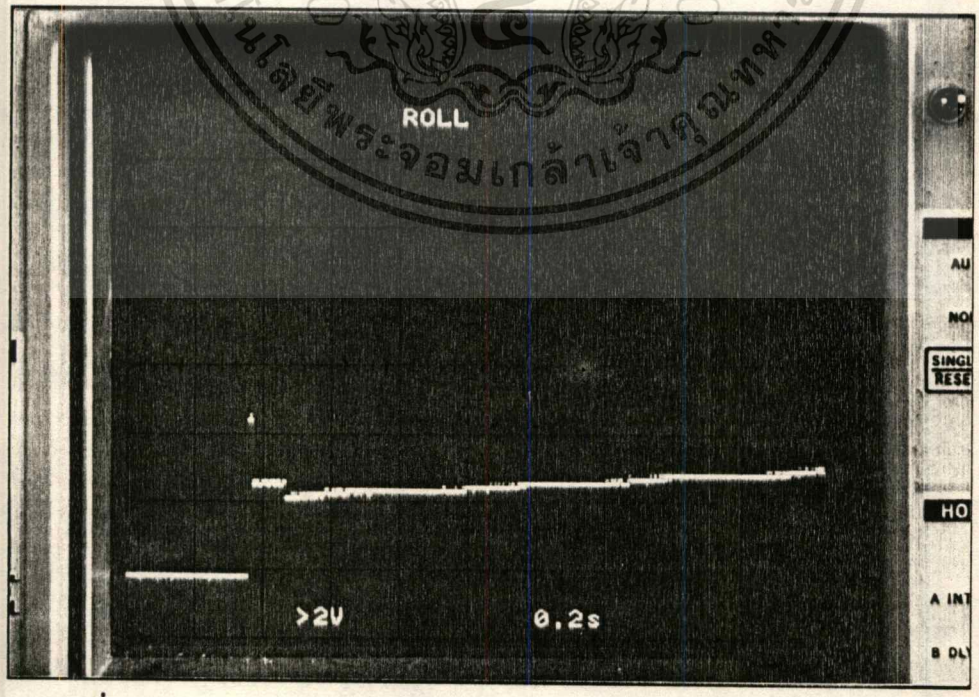
หัวข้อนี้เป็นการทดสอบโหมดการทำงานแบบ PID ของดิจิทัลคอนโทรลเลอร์โดยใช้ซอฟต์แวร์เลียนแบบการทำงาน ซึ่งรูทีนสำคัญของอัลกอลิธึมของโหมดนี้คือ รูทีน PID_control โดยที่จะใช้เวลาคำนวณแต่ละครั้งจะใช้เวลาสิ้นเปลือง โดยวิธีการจับเวลาการทำงานดังนี้

ให้ดิจิทัลคอนโทรลเลอร์ทำงานในโหมด PID จำนวน 960 วงรอบ ใช้เวลา 20.5 วินาที ดังนั้นในแต่ละครั้งหรือ 1 วงรอบจะใช้เวลา 21 มิลลิวินาที ซึ่งสามารถสรุปได้ว่า พล้านต์จำลอง จะต้องมีค่าคงที่ของเวลา มากกว่าเป็นสองเท่า คือ 42 วินาที,

- การทดสอบคุณลักษณะของการควบคุมโหมด PID ในลูปเปิด โดยป้อนอินพุตสเต็ป และหาผลการตอบสนอง ตามรูปดังนี้



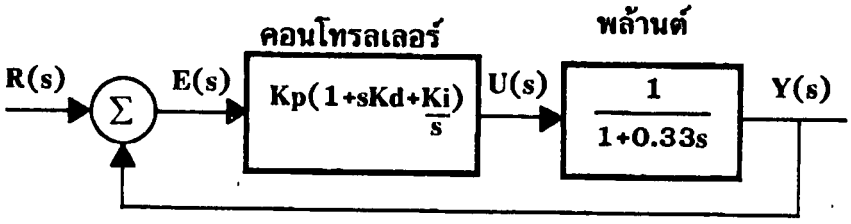
รูปที่ 6-26 *การชิมูเลชันของการตอบสนองสเต็ปของ PID ในลูปเปิด*



รูปที่ 6-27 *การตอบสนองสเต็ป 25 % ของ PID โดยวัดจากสโคป*
Kp = 1 , Ki = 0.2 ,Kd = 0.5 และ PV = 0

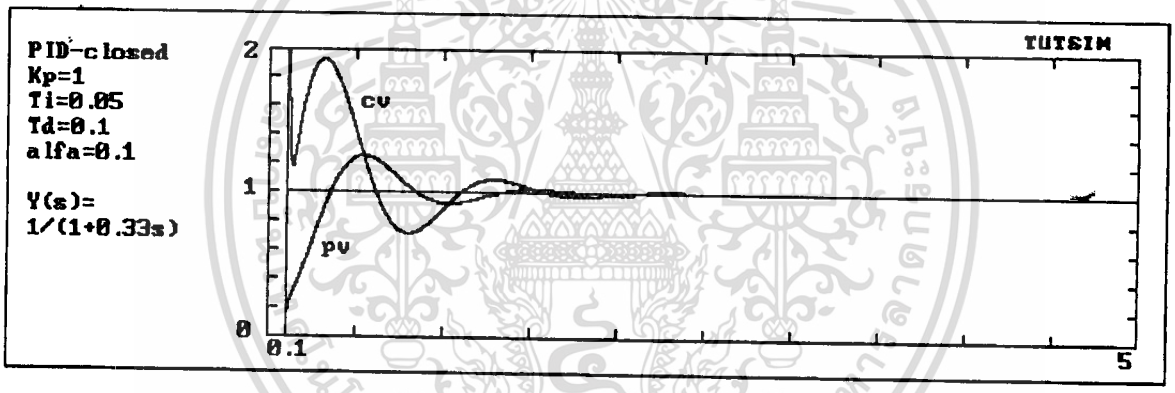
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การทดสอบการตอบสนองสเต็ปในโหมด PID โดยกำหนด พล้านต์จำลองเป็น ฟังก์ชันถ่ายโอนเป็นแบบออร์เดอร์หนึ่ง โดยมีบล็อกไดอะแกรมของระบบดังนี้

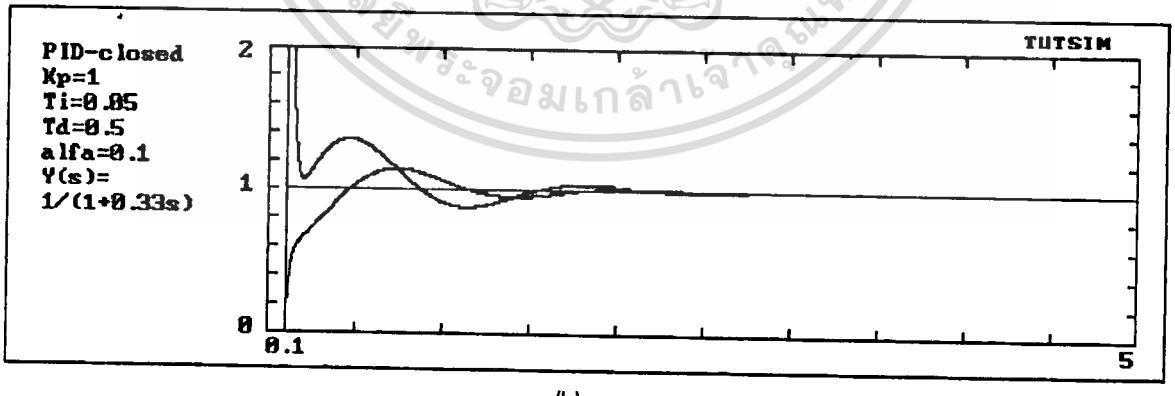


รูปที่ 6-28 *บล็อกไดอะแกรมของการควบคุมแบบ PID โดยให้พล้านต์เป็นออร์เดอร์หนึ่ง*

ตามบล็อกไดอะแกรมข้างต้นเราสามารถซิมูเลชัน ของการควบคุมแบบ PID ซึ่งใช้พล้านต์เป็นสมการออร์เดอร์หนึ่ง โดยกำหนดพารามิเตอร์แบบต่างๆ



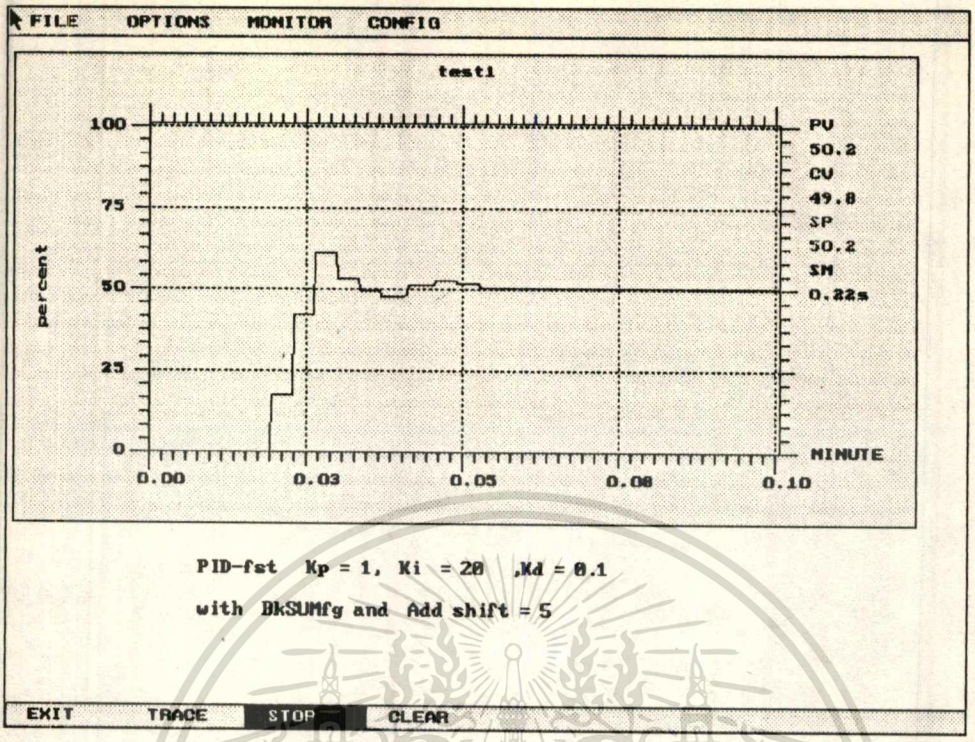
(a)



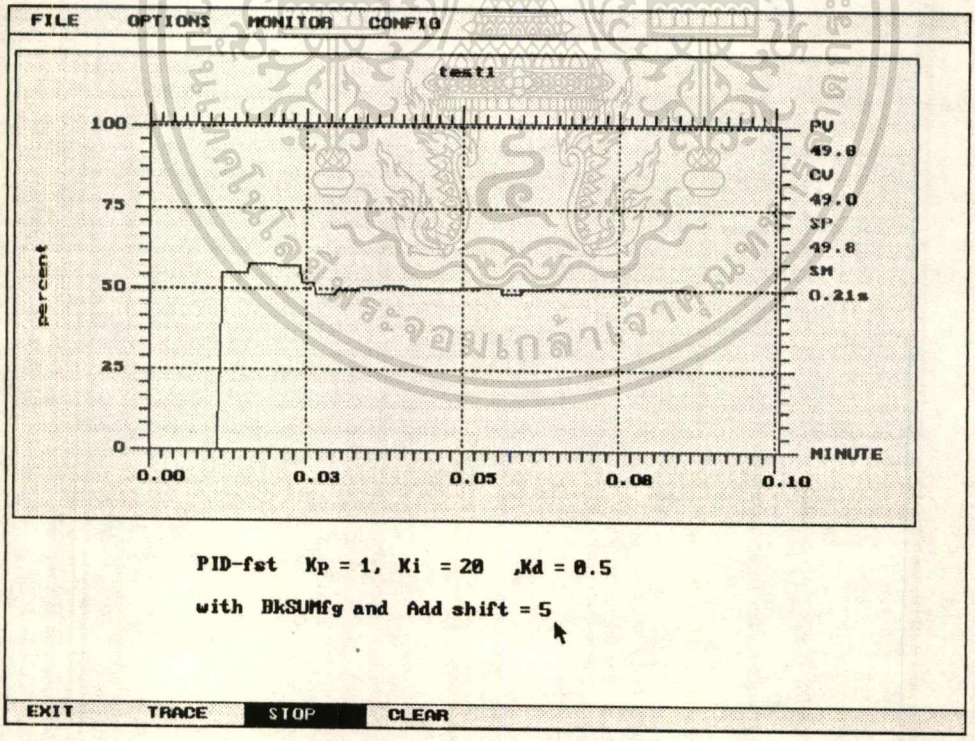
(b)

รูปที่ 6-29 (a),(b) *ผลการซิมูเลชันโดยการควบคุมแบบ PID โดยให้พล้านต์เป็นสมการออร์เดอร์หนึ่ง ในขณะค่า Kp,Ki,Kd ต่างกัน โดยโปรแกรมคณิตศาสตร์*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



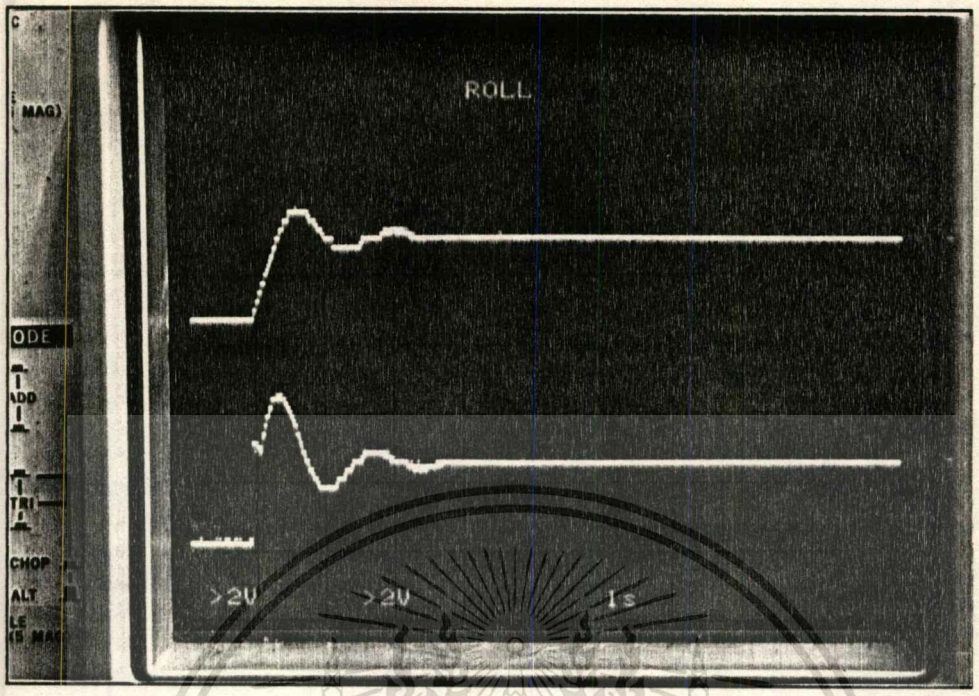
(a) Kp = 1.0 , Ki = 20 , Kd = 0.1



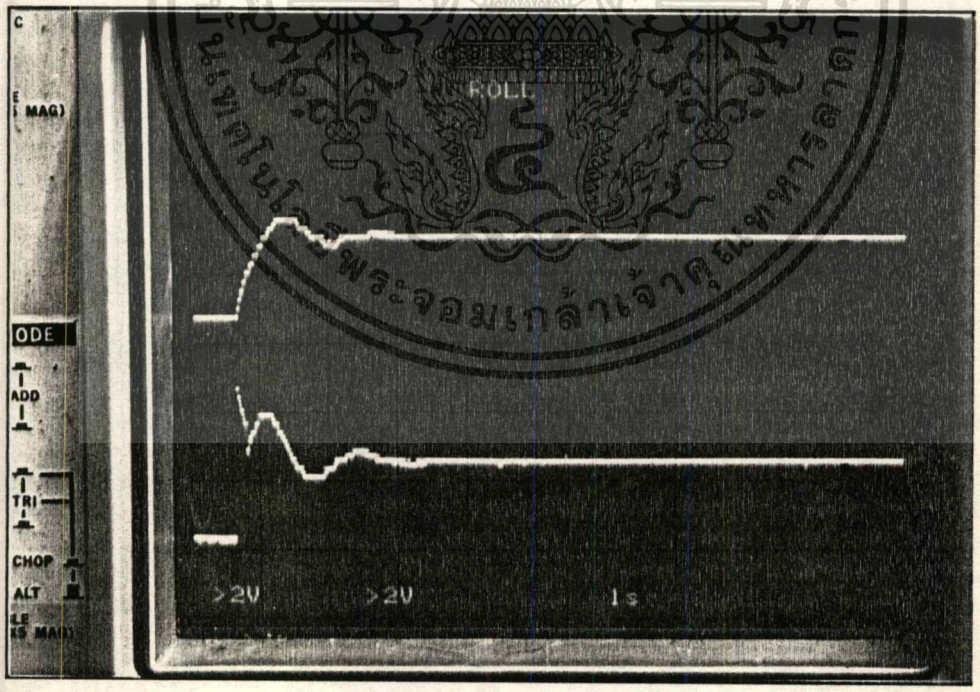
(b) Kp = 1.0 , Ki = 20 , Kd = 0.5

รูปที่ 6-30 *(a),(b) ผลการตอบสนอง 50% สเต็ปของ PID โดยสังเกตจากโฮส*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(a) Kp = 1 , Ki = 20 ,Kd = 0.1



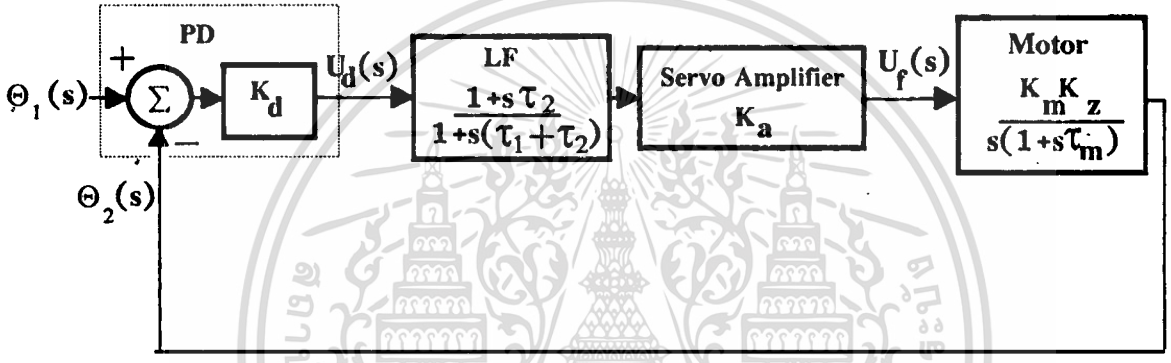
(b) Kp = 1 , Ki = 20 ,Kd = 0.5

รูปที่ 6-31 (a),(b) ผลการตอบสนอง 50 % สเต็ปของ PID โดยวัดจากสโคป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

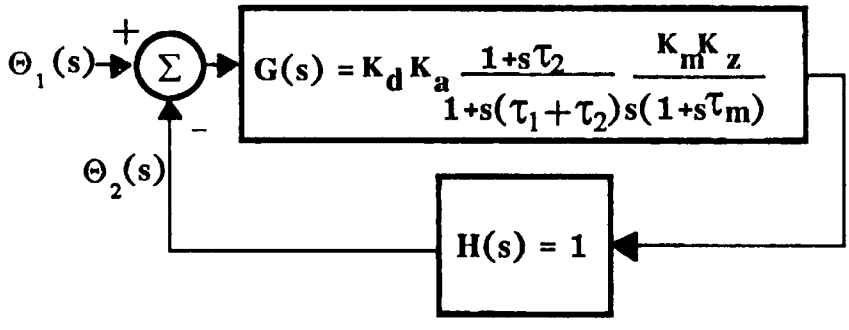
6.3 การวิเคราะห์ผลการทดลองในโหมดเฟสล็อก ในระบบควบคุมความเร็วมอเตอร์จำลอง

โมเดลทางคณิตศาสตร์ของระบบควบคุม โดยใช้โหมดเฟสล็อกของระบบควบคุมความเร็วมอเตอร์ สามารถเขียนได้ดังรูป 6-32 เซอร์โวแอมปลิไฟล์ถูกสมมุติให้มีค่าเกน K_a โดยปกติค่าโพลของแอมปลิไฟล์นี้จะถูกตัดทิ้งเพราะว่า ความถี่ของโพลตัวนี้สูงกว่าของมอเตอร์มาก ระบบจะประกอบด้วย 3 โพล และ 1 ซีโรของลูปฟิลเตอร์ซึ่งต้องระบุค่าด้วย มิฉะนั้นแล้ว เฟสของฟังก์ชันถ่ายโอนจะเกิน 180 องศา ที่ความถี่สูง และระบบจะไม่เสถียรภาพ



รูป6-32 *บล็อกไดอะแกรมของระบบควบคุมความเร็วมอเตอร์ โดยใช้เฟสล็อก*

แต่ละบล็อกตามรูป 6-33 สามารถนำมารวมกันได้เพื่อจัดให้เป็นรูปง่ายขึ้น ดังรูป 6-33 ในระบบนี้ ฟังก์ชันถ่ายโอนของส่วนแรกจะกำหนดเป็น $G(s)$ ดังนั้นฟังก์ชันถ่ายโอนของส่วนป้อนกลับ(มอเตอร์)จะเป็น $H(s)$



รูปที่ 6-33 *บล็อกไดอะแกรมลดรูปให้ง่ายขึ้น ของรูปที่ 6-32*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อระบบควบคุมความเร็วมอเตอร์ถูกออกแบบจริง พารามิเตอร์บางตัวต้องถูกกำหนดขึ้น อย่างเช่น K_m คือเกนของมอเตอร์ τ_m คือค่าคงที่เวลาของมอเตอร์ K_z คือจำนวนซี่ของเอนโค๊ด-เดอร์ในแต่ละรอบ ส่วนพารามิเตอร์ที่เหลือ(K_a และ τ_2) ต้องเลือกให้มีค่าไดนามิคดีที่สุด และระบบเสถียรภาพมากที่สุด มีหลายวิธีในการแก้ปัญหานี้ในทางวิศวกรรม ในที่นี้เราจะเลือกวิธีโบดไดอะแกรม (Bode diagram) โดยหาจากฟังก์ชันลูปเปิด $G(s)H(s)$

ที่ซึ่ง

$$G(s)H(s) = \frac{K_d K_a K_m K_z}{\tau_1} \bullet \frac{1+s\tau_2}{s[1+s(\tau_1+\tau_2)](1+s\tau_m)}$$

โดยที่ $K = K_d K_a K_m K_z$

ที่ซึ่งจากเอกสารอ้างอิง[8] K_d จะเท่ากับ

$$K_d = \frac{DF\ High - UF\ Low}{2(2\pi)} = \frac{5V - 0V}{4\pi} = 0.4\ V/rad$$

เพื่อให้เป็นการง่ายต่อการสร้างโบดพล็อต จะต้องกำหนดค่าเริ่มต้นของตัวแปรดังกล่าวจากการทดลองดังต่อไปนี้

$$K_a = K_m = K_z = 1$$

ดังนั้น

$$K = 0.4$$

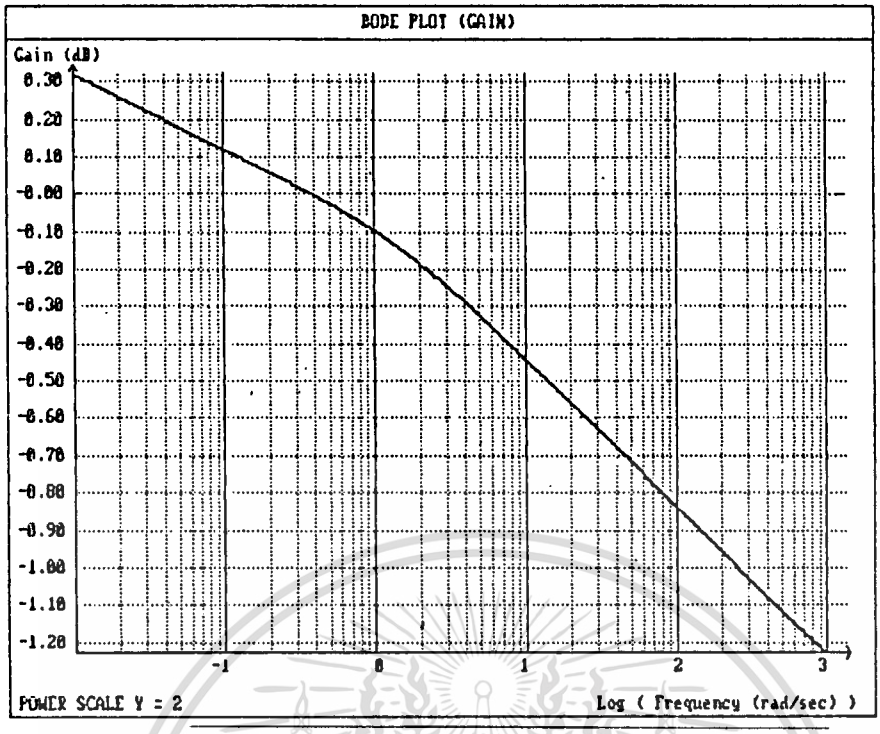
และกำหนด $\tau_1 = 0.33\text{sec}$, $\tau_2 = 0.39\text{sec}$

ทำให้

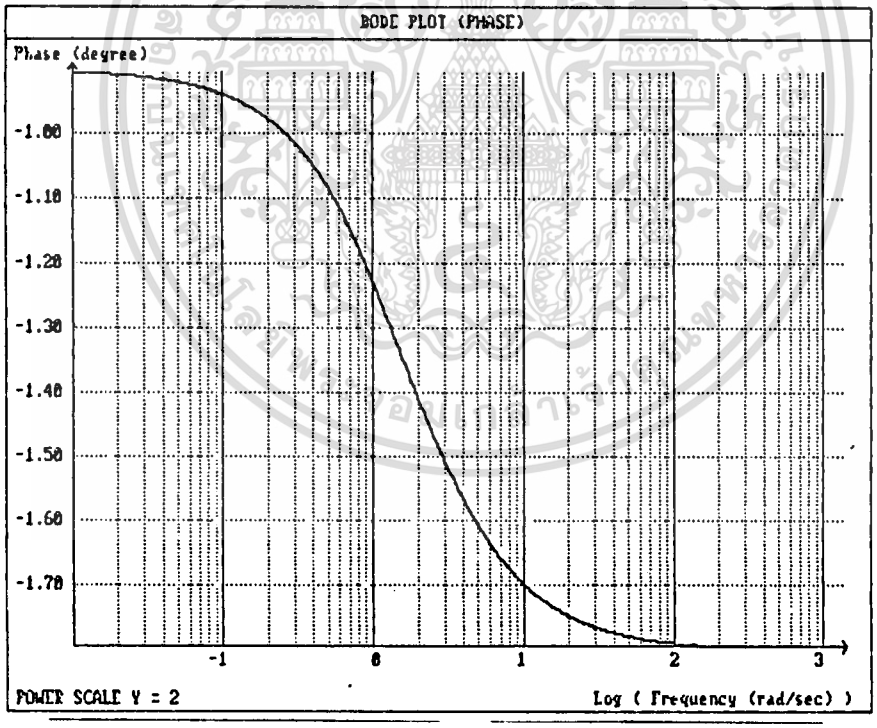
$$G(s)H(s) = 0.4 \bullet \frac{1.625s + 4.17}{s^3 + 4.38s^2 + 4.17s}$$

แทนค่าตัวแปรในสมการ $G(s)H(s)$ นำไปพล็อตกราฟ โบดไดอะแกรมดังรูปที่ 6-34 และรูปที่ 6-35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-34 *แสดงแอมปลิจูดพล็อต ของโบดไดอะแกรม*



รูปที่ 6-35 *แสดงเฟสพล็อต ของโบดไดอะแกรม*

จากรูปที่ 6-34 และ 6.35 ข้างต้นเราได้ค่าเฟสมาจิน(phase marjin) $\alpha = 75^{\circ}$ ซึ่งเรายังชี้ชัดไม่ได้ว่าระบบจะเกิดอันเดอร์แดมป์ หรือ โอเวอร์แดมป์ และความกว้างของแบนด์วิทในการใช้งานจริง ที่ซึ่งแบนด์วิท ω_G ในระบบปิด เราสามารถหาได้จากเส้นโค้งแอมปลิจูดตัดกับ แกน x ซึ่งจากรูปโบดไดอะแกรม จะได้ $\omega_G \approx 0.6\ rad \bullet \sec^{-1}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จริง ที่ซึ่งแบนด์วิท ω_G ในระบบปิด เราสามารถหาได้จากเส้นโค้งแอมปลิจูดตัดกับ แกน x ซึ่งจากรูปโบดไดอะแกรม

จะได้ $\omega_G \approx 0.6\ rad \bullet \sec^{-1}$

ค่าแดมปิงแฟกเตอร์ ζ สามารถหาค่าโดยประมาณ [9] ได้ดังนี้

$$\zeta = \frac{\alpha}{3n}$$

ที่ซึ่ง n คือค่าสโลบลาดลงของเส้นโค้งแอมปลิจูด ดังนั้น

$n \approx 40$ dB per Decade

$\alpha = 73^\circ$

เพราะฉะนั้น

$\zeta = 0.625$

ระบบจะอันเดอร์แดมป์ ถ้าจะให้ความสามารถไดนามิคดีกว่านี้ ควรจะออฟติไมซ์ค่าพารามิเตอร์เพื่อให้ค่าแดมปิงดีกว่านี้

6.4 การวิเคราะห์ผลการทดลองในโหมด PI ในระบบควบคุมความเร็วมอเตอร์จำลอง

ในระบบควบคุมความเร็วมอเตอร์ ตัวคอนโทรลเลอร์ที่ใช้ได้เหมาะสมที่สุดคือ การควบคุมแบบ PI ซึ่งฟังก์ชันถ่ายโอนของระบบปิด จะอยู่ในรูปของสมการออร์เดอร์สอง ซึ่งสามารถวิเคราะห์หาค่าพารามิเตอร์ได้ง่าย ซึ่งสามารถจัดสมการลักษณะ(characteristc equation) ให้อยู่ในรูปนอมอลไลซ์(normalized form) คือ

$$s^2 + 2\zeta\omega_n s + \omega^2{}_n \ : \text{nomalized form}$$

โดยสามารถหาสมการลักษณะ C.E. ของการควบคุม PI ในระบบควบคุมความเร็วมอเตอร์ดังนี้

$$C.E. \ = \ 1 + G(s)H(s) \ = \ 0$$

ที่ซึ่ง

$$G(s) \ = \ K_p\left[1 + \frac{K_i}{s}\right] \bullet \left[\frac{1}{1 + s\tau_m}\right]$$

$$H(s) \ = \ 1$$

ดังนั้น

$$C.E. \ = \ s^2 + \frac{1 + K_p}{\tau_m}s + \frac{K_p K_i}{\tau_m} \ = \ 0$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพราะฉะนั้น

$$\omega_n = \sqrt{\frac{K_p K_i}{\tau_m}} \quad \text{และ} \quad 2\zeta\omega_n = \frac{1+K_p}{\tau_m}$$

สังเกตว่าฟังก์ชันถ่ายโอนลูปเปิด $G(s)H(s)$ ข้างต้นจะเป็นแบบ 1(type 1)เพราะว่ามีโพลอยู่ที่จุดกำเนิด(origin)หนึ่งตัว และเป็นฟังก์ชันถ่ายโอนออร์เดอร์สอง

ดังนั้นการหาเสถียรภาพของระบบคุมคุมความเร็วของมอเตอร์ด้วย PI จะสามารถเทียบได้จาก กราฟการตอบสนองเอาต์พุตนอมอลไลซ์(nomalized output response) ของ $\omega_n t$

กำหนด $\zeta = 0.7$

$ERROR < 2\%$

จากกราฟนอมอลไลซ์จะได้ ค่า $\omega_n t$ = 7.0

<u>หมายเหตุ</u> ในทางปฏิบัติเราไม่จำเป็นกำหนดพารามิเตอร์ K_p ของดิจิทัลคอนโทลเลอร์ให้มีค่ามากได้ เพราว่าในทางปฏิบัติเอาต์พุตควบคุม CV (control variable output) จะอิ่มตัวที่ค่าโวลต์เตจสูงสุด สำพันธ์กับจำนวนบิต ของ D/A ถ้า 8 บิตคือ 0FFh ดังนั้น เมื่อเพิ่ม K_p ขึ้นก็ไม่อาจจะเพิ่มค่าไรส์ไทม์(rise time) ของพล้านต์ได้ ซึ่งในกรณีนี้สามารถเพิ่มเกนของระบบที่ส่วนอื่นได้ โดยใช้ตัวแอมปลิไฟล์ เป็นต้น

จากหมายเหตุ จึงกำหนดค่า K_p = 4.0

$$\omega_n = \frac{1+K_p}{\tau_m} \bullet \frac{1}{2\zeta}$$

ที่ซึ่ง τ_m= 0.33sec คือค่าคงที่เวลาของมอเตอร์

ดังนั้นจะได้

$$\omega_n = 15.15$$

สรุปได้ว่า เมื่อพล้านต์เข้าสู่เซตพอย์น ที่ออฟเซตเออเรอร์น้อยกว่า 2% จะต้องใช้ค่าเซตตลิง-ไทม์(seting time มากกว่า 0.46 วินาที

ดังนั้นในการหาออฟติมัม(optimum)ของเสถียรภาพของระบบตามเงื่อนไขข้างต้นจะได้

$$K_p = 4.0\,, \quad K_i = 20 \quad : \text{PI - control}$$

ผลการใช้พารามิเตอร์นี้ ดูได้ในผลการทดลอง

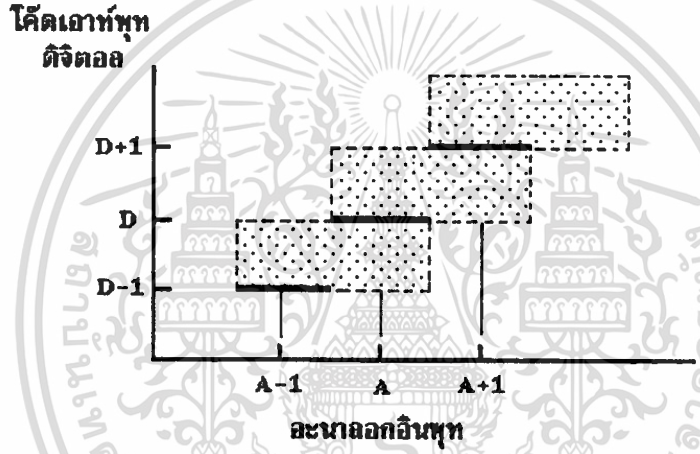
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.5 เออเรอร์ที่เกิดขึ้นจากการรับค่าอินพุตป้อนกลับ PV จากพล้านต์

-เออเรอร์เกิดขึ้นจากโหมดควบคุมปกติ

ส่วนใหญ่การควบคุมโดยใช้โหมดทั่วไปคือโหมด P,PI,PID โดยใช้ไมโครโปรเซสเซอร์ดำเนินการ ซึ่งการวัดค่าป้อนกลับ PV เข้ามาทางอินพุต ซึ่งเออเรอร์เกิดขึ้นเป็นประการแรกคือ การแปลงแอนะล็อกเป็นดิจิทัล

จากรูปที่ 6-36 แสดงถึงในกรณีที่แย่ที่สุดในการพล็อตกราฟฟังก์ชันการเปลี่ยน สำหรับ ไอซี A/D เบอร์ AD0804



รูปที่ 6-36 *เป็นกราฟฟังก์ชันการแปลงของ A/D ซึ่งมีค่าความเที่ยงตรง(accuracy) 1 บิต*

จากรูปจะเห็นได้ว่า ถ้าอินพุตเข้ามาเท่ากับค่า A จะทำให้เอาต์พุตซึ่งเป็นดิจิทัล มีเออเรอร์เท่ากับ 2 LSB ถ้าคิดเป็นเปอร์เซ็นต์ความเที่ยงตรงจะได้เท่ากับ

ถ้าสมมุติว่า A/D เป็น Ideal โดยมีค่าเออเรอร์เท่ากับ 0 บิต และ A/D มีขนาด 8 บิต

เปอร์เซ็นต์เออเรอร์เท่ากับ 1/256 x 100% = 0.4 %

ในการทดลองจริง A/D มีค่าเออเรอร์เท่ากับ $\pm$ 1 LSB เป็น A/D ขนาด 8 บิต คิดเป็นดังนี้

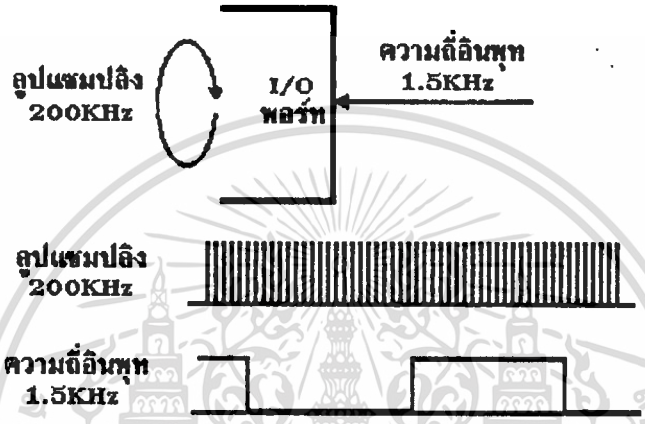
เปอร์เซ็นต์เออเรอร์เท่ากับ 1/64 x 100% = 1.6 %

ถึงอย่างไรก็ดี ถ้าเราต้องการให้มีค่าความเที่ยงตรงสูงกว่านี้ควรจะใช้ตัว A/D ที่มีขนาดจำนวนบิตมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-เออเรอร์เกิดจากโหมดควบคุมเฟสล็อก

ในรูทีนการควบคุมแบบเฟสล็อกนั้น อินพุตป้อนกลับที่รับเข้ามาจะเป็นคลื่นสี่เหลี่ยม ซึ่งมีความถี่ประมาณ 1.5 กิโลเฮิรตซ์ เทียบเป็น อินพุต 50 เปอร์เซ็นต์ ซึ่งการควบคุมแบบนี้จะตรวจจับขอบขาขึ้นและขอบขาลง ของอินพุต โดยที่กระบวนการตรวจจับนี้จะใช้ซอฟต์แวร์วนลูป แสดงตามรูปที่ 6-37 โดยคิดเวลาสิ้นเปลืองแต่ละลูป ประมาณ 5 แมชีนไซเคิล เป็นอย่างมาก ซึ่งคิดเป็นความถี่เท่ากับ 1/ 5 เมกะเฮิรตซ์ หรือเท่ากับ 200 กิโลเฮิรตซ์ ถ้าคิดเป็น เปอร์เซ็นต์เออเรอร์แล้วเท่ากับ



รูปที่ 6-37 *การใช้ซอฟต์แวร์วนลูปเพื่อแซมปลิงความถี่อินพุต ในโหมดเฟสล็อก*

เปอร์เซ็นต์เออเรอร์ = 1.5/200 x 100% = 0.75%

6.6 เหตุผลเปรียบเทียบการตอบสนองกระบวนการ ในขณะที่ใช้โหมดเฟสล็อก และโหมด PI ในการควบคุมความเร็วมอเตอร์จำลอง

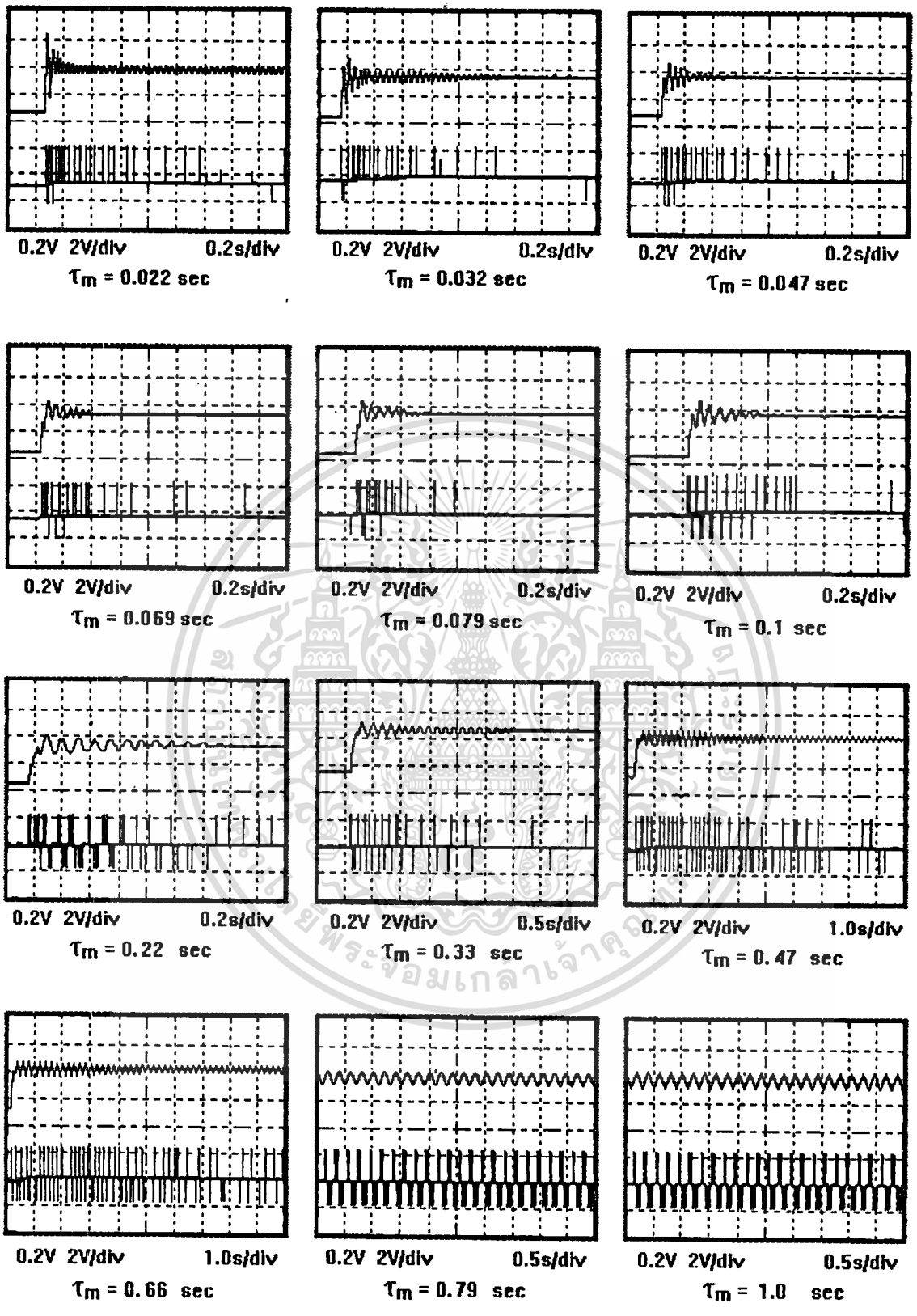
ค่าต่างๆที่ได้จากการทดลองสามารถนำมาสร้างตารางการเปรียบเทียบ ระหว่างการควบคุมแบบเฟสล็อก กับ การควบคุมแบบ PI ในเงื่อนไขที่ใช้งานบนไม่โครโปรเซสเซอร์ โดยทั้งคู่ต่างก็เป็นรูทีนทางซอฟต์แวร์ ซึ่งในการควบคุมความเร็วของมอเตอร์ ที่สเต็ปเซ็ตพอย์น 0%-50% แล้ววัดการตอบสนองของระบบ(เช่น มอเตอร์) ซึ่งสามารถจำแนกได้เป็นตารางดังนี้

ตารางที่ 6-1 จะเปรียบเทียบผลการทำงานในโหมด PLL และ PI ของพล้านต์ออร์เดอร์หนึ่ง โดย τ_m= 0.33s

โหมด	โอเวอร์ชูต	ไรส์ไทม์	เซตตลิงไทม์	ออฟเซตเออเรอร์
เฟสล็อก	18.5%	0.16sec	2.5 sec	~0.0%
PI	41.1%	0.23 sec	1.11 sec	~0.0%

ตารางที่ 6-1 *การเปรียบเทียบผลการทดลองระหว่าง PLL กับ PI ในเงื่อนไขเดียวกัน*

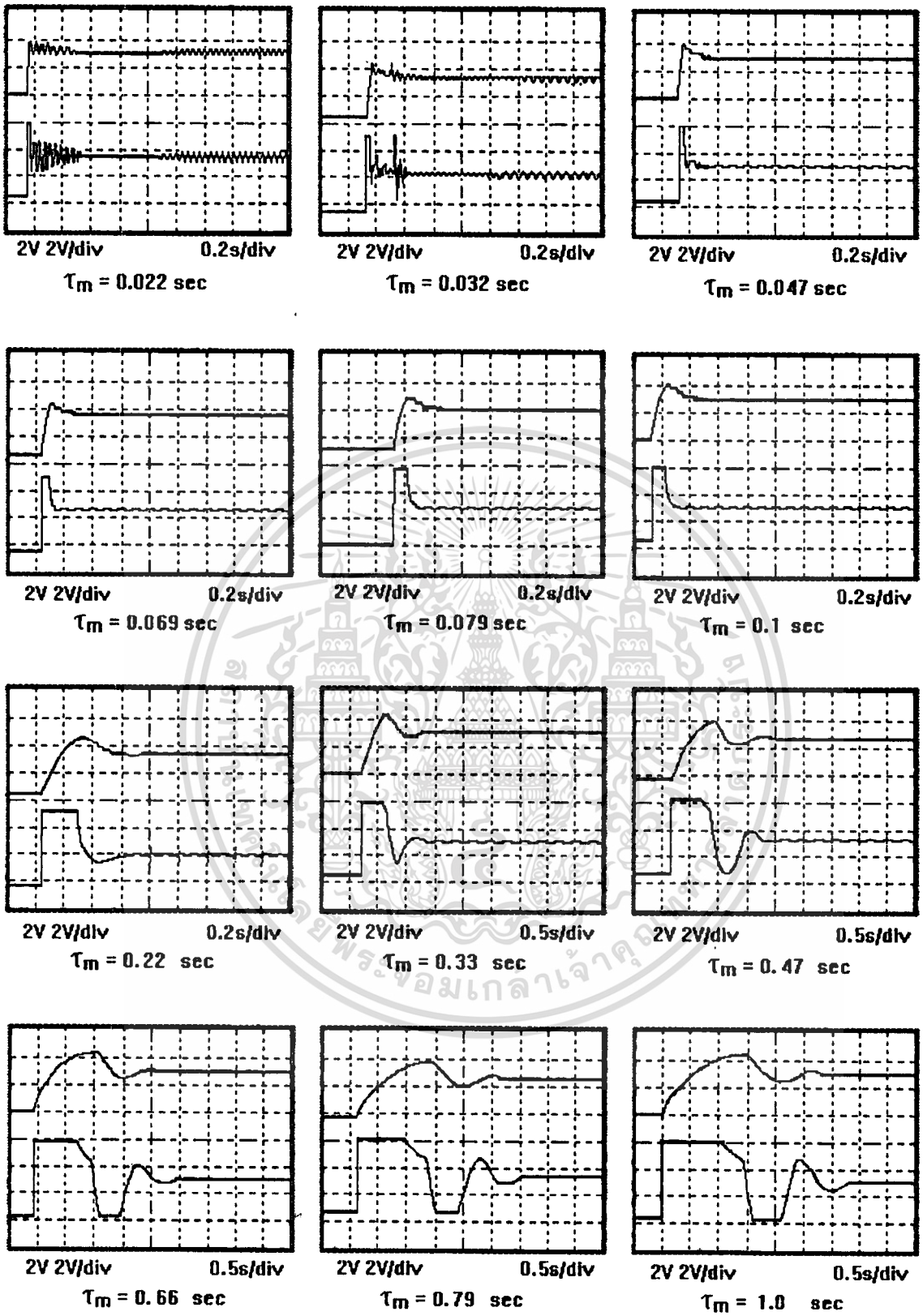
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-38 *ผลการทดสอบโหมด PLL โดยแปรค่า τ_m ของพล้านต์ออร์เดอร์หนึ่ง*

ใช้ LF=(1+0.39s)/(1+0.72s) , บน: เอาต์พุตพล้าน, ล่าง:เอาต์พุต LF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6-39 *ผลการทดสอบโหมด PI โดยแปรค่า τ_m ของพล้านต์ออร์เดอร์หนึ่ง*

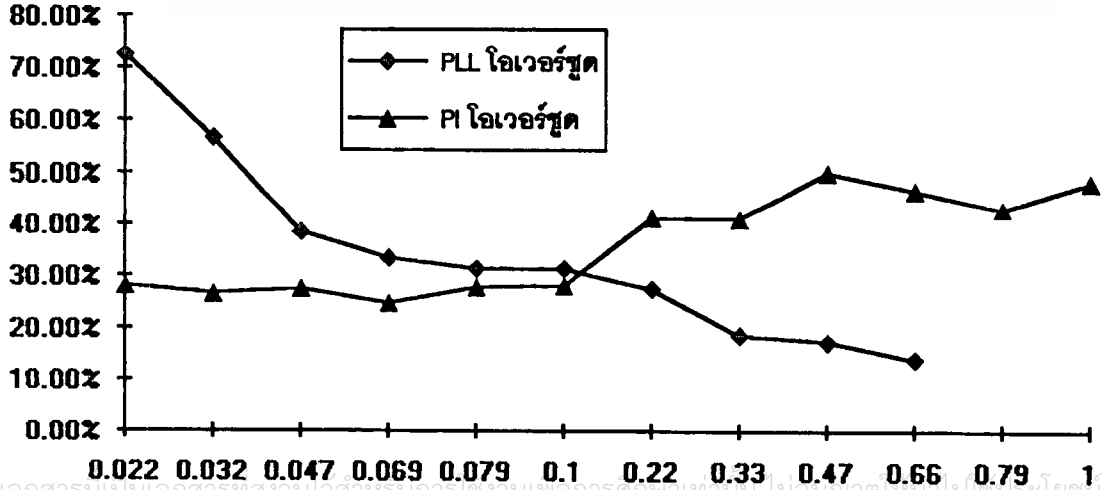
ใช้ Kp = 4 , Ki = 20 , บน: เอาต์พุตพล้าน(PV), ล่าง:เอาต์พุตควบคุม(CV)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6-2 จะเปรียบเทียบค่าโอเวอร์ชูตระหว่างโหมดการทำงานแบบ PLL กับโหมด PI โดยเปลี่ยนแปลงพลานต์ออร์เดอร์หนึ่งให้มี τ_m ค่าต่างๆ แล้วเขียนกราฟเปรียบเทียบดังรูปที่ 6-40

τ_m (Sec)	PLL โอเวอร์ชูต (%)	PI โอเวอร์ชูต (%)
0.022	72.5	28.1
0.032	56.7	26.6
0.047	38.6	27.5
0.067	33.45	24.7
0.079	31.25	27.8
0.1	31.3	28.1
0.22	27.4	41.2
0.33	18.5	41.1
0.47	17.3	50
0.66	13.9	46.6
0.79	วัดไม่ได้	43.0
1.0	วัดไม่ได้	48.1

ตารางที่ 6-2 *การเปรียบเทียบโอเวอร์ชูตของ PI และ PLL ใน τ_m ของพลานต์ต่างๆ*



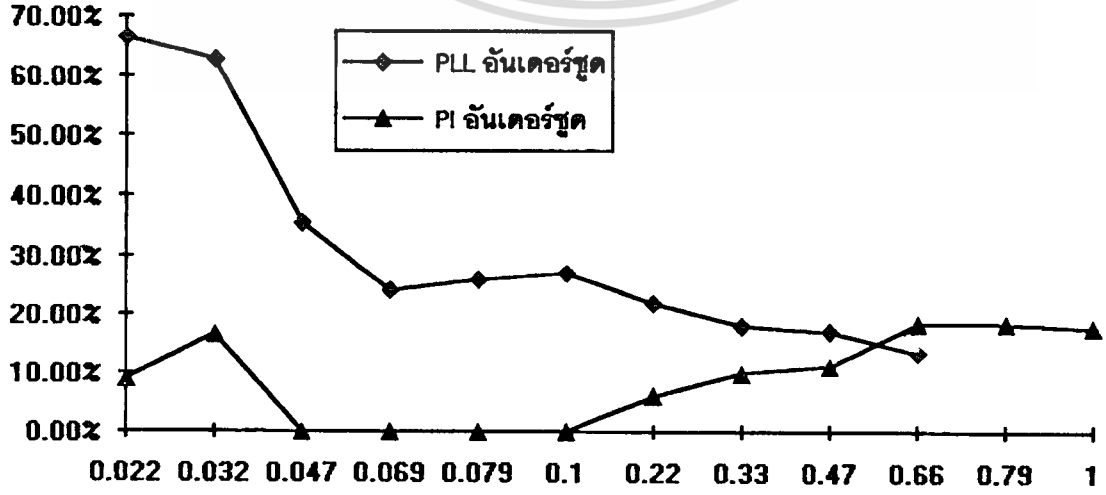
รูปที่ 6-40 *กราฟแสดงการเปรียบเทียบข้อมูลโอเวอร์ชูต จากตารางที่ 6-2*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6-3 จะเปรียบเทียบอันเดอร์ชูตระหว่างโหมดการทำงานแบบ PLL กับโหมด PI โดยเปลี่ยนแปลง พล้านต์ออร์เดอร์หนึ่งให้มี τ_m ค่าต่างๆ แล้วเขียนกราฟเปรียบเทียบดังรูปที่ 6-41

τ_m (Sec)	PLL อันเดอร์ชูต (%)	PI อันเดอร์ชูต (%)
0.022	66.5	9
0.032	62.7	16.7
0.047	35.5	0
0.067	24.0	0
0.079	25.9	0
0.1	27.0	0
0.22	21.8	6
0.33	18	10
0.47	17	11.2
0.66	13.4	18.4
0.79	วัดไม่ได้	18.4
1.0	วัดไม่ได้	17.7

ตารางที่ 6-3 *การเปรียบเทียบอันเดอร์ชูตของ PI และ PLL ใน τ_m ของพล้านต์ต่างๆ*



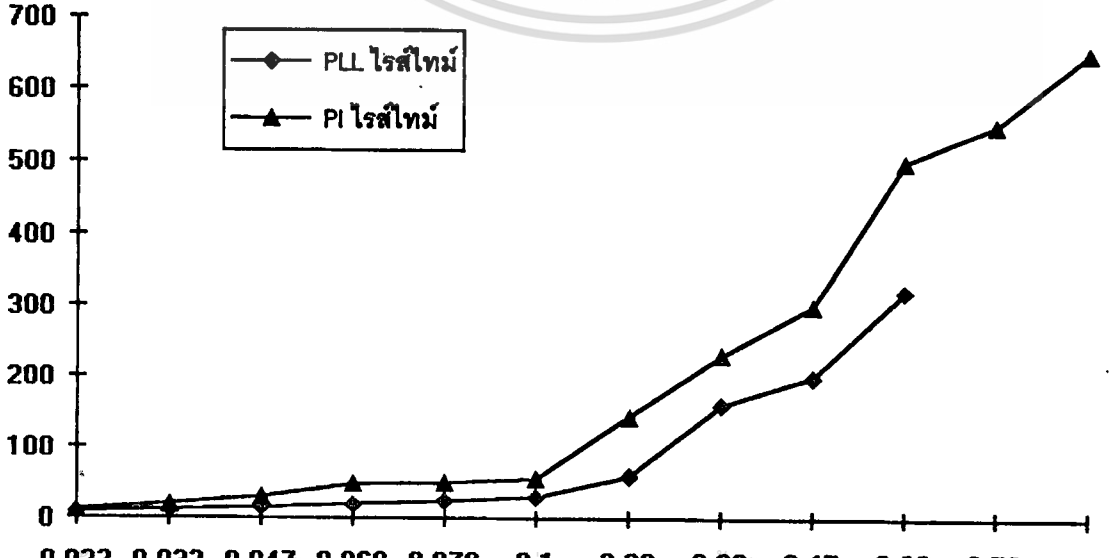
รูปที่ 6-41 *กราฟแสดงการเปรียบเทียบข้อมูลอันเดอร์ชูต จากตารางที่ 6-3*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6-4 จะเปรียบเทียบไรส์ไทม์ระหว่างโหมดการทำงานแบบ PLL กับ PI โดยเปลี่ยนแปลง พล้านต์ออร์เดอร์หนึ่งให้มี τ_m ค่าต่างๆ แล้วเขียนกราฟเปรียบเทียบดังรูปที่ 6-42

τ_m (Sec)	PLL ไรส์ไทม์ (mS)	PI ไรส์ไทม์ (mS)
0.022	8	12
0.032	12	20
0.047	15	30
0.069	20	48
0.079	24	50
0.1	30	56
0.22	60	144
0.33	160	230
0.47	200	300
0.66	320	500
0.79	วัดไม่ได้	550
1.0	วัดไม่ได้	650

ตารางที่ 6-4 *การเปรียบเทียบไรส์ไทม์ของ PI และ PLL ใน τ_m ของพล้านต์ต่างๆ*



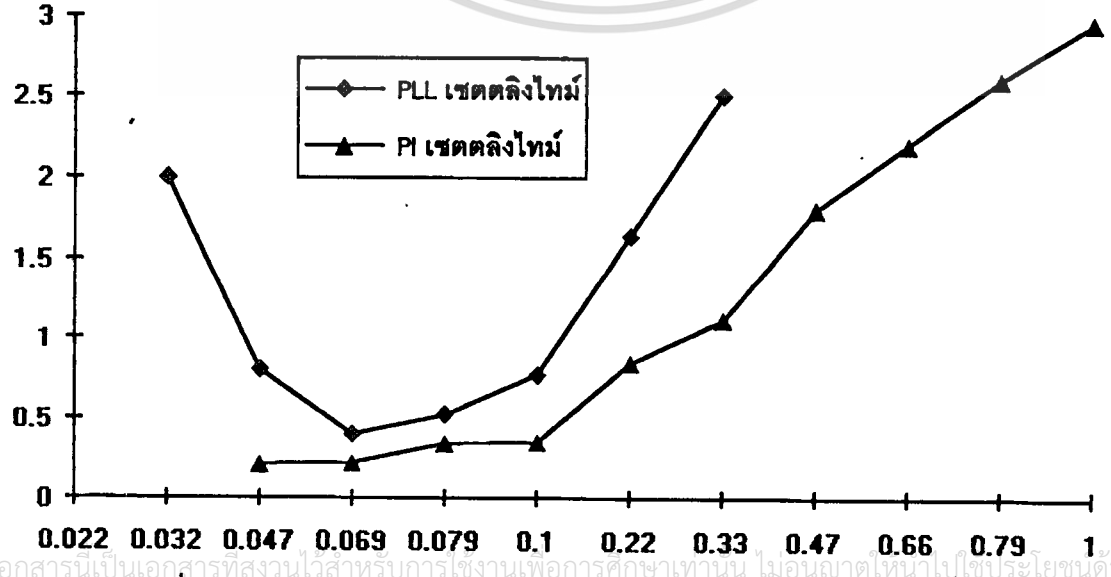
รูปที่ 6-42 *กราฟแสดงการเปรียบเทียบไรส์ไทม์ของข้อมูลในตารางที่ 6-4*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6-5 จะเปรียบเทียบเซตตลิงไทม์ระหว่างโหมดการทำงานแบบ PLL กับโหมด PI โดยเปลี่ยนแปลง พล้านต์ออร์เดอร์หนึ่งให้มี τ_m ค่าต่างๆ แล้วเขียนกราฟเปรียบเทียบดังรูปที่ 6-43

τ_m (Sec)	PLL เซตตลิงไทม์ 2% (Sec)	PI เซตตลิงไทม์ 2% (Sec)
0.022	วัดไม่ได้	วัดไม่ได้
0.032	2.0	วัดไม่ได้
0.047	0.76	0.21
0.069	0.4	0.22
0.079	0.52	0.34
0.1	0.77	0.35
0.22	1.64	0.84
0.33	2.5	1.11
0.47	วัดไม่ได้	1.8
0.66	วัดไม่ได้	2.2
0.79	วัดไม่ได้	2.6
1.0	วัดไม่ได้	2.9

ตารางที่ 6-5 *การเปรียบเทียบเซตตลิงไทม์ของ PI และ PLL ใน τ_m ของพล้านต์ต่างๆ*



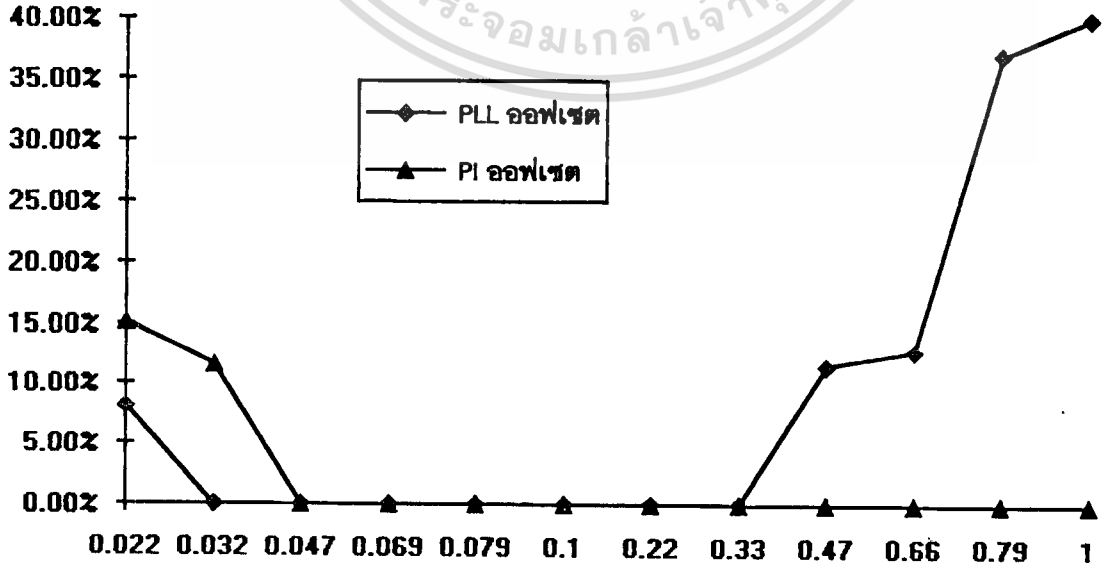
รูปที่ 6-43 *กราฟแสดงการเปรียบเทียบเซตตลิงไทม์ของข้อมูลในตารางที่ 6-5*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6-6 จะเปรียบเทียบเซตตลิงไทม์ระหว่างโหมด PLL กับโหมด PI โดยปลี่ยนแปลงพล้านต์ออร์เดอร์หนึ่งให้มี τ_m ค่าต่างๆ แล้วเขียนกราฟเปรียบเทียบดังรูปที่ 6-44

τ_m (Sec)	PLL ออฟเซตเออเรอร์ (%)	PI ออฟเซตเออเรอร์ (%)
0.022	8.1	15.2
0.032	0.0	11.6
0.047	0.0	0.0
0.069	0.0	0.0
0.079	0.0	0.0
0.1	0.0	0.0
0.22	0.0	0.0
0.33	0.0	0.0
0.47	11.5	0.0
0.66	12.8	0.0
0.79	37.0	0.0
1.0	40.0	0.0

ตารางที่ 6-6 *การเปรียบเทียบออฟเซตเออเรอร์ของ PI และ PLL ใน τ_m ของพล้านต์ต่างๆ*



รูปที่ 6-44 *กราฟแสดงการเปรียบเทียบออฟเซตเออเรอร์ของข้อมูลในตารางที่ 6-6*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

สรุปวิจารณ์ในงานวิจัยตลอดจนปัญหาและอุปสรรค์ด้านเทคนิคของงานวิจัย

7.1 ข้อสรุปผลการทดลอง

7.1.1 ผลการทดสอบตัวดิจิทัลคอนโทรลเลอร์

1) ในโหมดการควบคุมเฟสล็อก

การทดสอบระบบควบคุมโดยใช้ตัวดิจิทัลคอนโทรลเลอร์ในโหมดเฟสล็อก โดยใช้ซอฟต์แวร์ภาษาแอสเซมบลีของ 8031 เลียนแบบอัลกอลิธึมการทำงานของตัวดีเทกเตอร์เฟส/ความถี่ หรือเฟสดีเทกเตอร์แบบ 4 โดยใช้โวลต์เตจคอนโทรลออสซิลเตอร์ หรือ VCO แทนกระบวนการอย่างหนึ่งซึ่งเปรียบเสมือนกับกระบวนการควบคุมความเร็วมอเตอร์นั้น การควบคุมจะถูกกำหนดโดยเซตพอย์น ตั้งแต่ 0 เปอร์เซ็นต์ ถึง 100 เปอร์เซ็นต์ โดยเปรียบเทียบเป็นความถี่ได้ตั้งแต่ 60 เฮิรตซ์ ถึง 1.5 กิโลเฮิรตซ์ โดยที่เซตพอย์นจะสามารถตั้งได้จากผู้ใช้โดยใช้แป้นคีย์

ความคิดที่เกิดขึ้นหลังการทดลองแบ่งได้เป็นหลายประเด็นคือ

1.ในสภาวะล็อก คือความถี่เซตพอย์นเท่ากับความถี่ป้อนกลับจากระบบนั้น ในการทดลองความถี่ทั้งสองจะเท่ากันเสมอ ซึ่งในสภาวะล็อกนี้จะเกิดเฟสเลื่อนเป็นบางครั้ง เป็นเพราะว่าเรกูเลชั่นของแหล่งจ่ายไฟก็ดี หรือว่าความเที่ยงของ VCO ซึ่งแทนมอเตอร์ก็ดี โดยเฟสเลื่อนจะเกิดขึ้นเป็นบางครั้ง ซึ่งจากการสังเกตไม่เกิน 180 องศา อย่างไรก็ดีก็ยังถือว่าอยู่ในสภาวะล็อกคือความถี่เท่ากัน

เมื่อเฟสเลื่อนไปการควบคุมก็ยังสามารถแก้ไขเฟสให้ถูกต้อง คือมีเฟสเดียวกันกับเฟสอ้างอิง(ยังอยู่ในสภาวะล็อก) การอินเฟสของความถี่อ้างอิงหรือความถี่เซตพอย์นกับความถี่ป้อนกลับจะขึ้นอยู่กับการตรวจสอบเฟสของกระบวนการทางโปรแกรม และฮาร์ดแวร์สนับสนุนของไมโครโปรเซสเซอร์ โดยให้ความถูกต้องสูงถึง 99.25 เปอร์เซ็นต์(มาจากการทดลองในบทที่ 6) โดยสามารถคิดเป็นความผิดพลาดในการตรวจสอบเฟสคือ 0.9 องศา

2.ข้อจำกัดในการควบคุมความเร็วของมอเตอร์สูงสุด เนื่องการกำหนดค่าเซตพอย์นสูงสุดคือ 100 เปอร์เซ็นต์หรือ เทียบเป็น 1.5 กิโลเฮิรตซ์ ซึ่งสามารถนำไปควบคุมมอเตอร์ได้ความเร็วสูงสุดเท่ากับ 1500 รอบต่อวินาที(ถ้าอัตราวัดของเอนโค๊ดเดอร์เท่ากับ 1 ต่อ 1) หรือเท่ากับ 90000 รอบต่อนาทีหรือ RPM ซึ่งถ้าพิจารณาแล้วความเร็วสูงสุดที่ได้นั้น จะสามารถครอบคลุมช่วงใช้งานของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มอเตอร์ทั่วไปได้ แต่ถ้าต้องการควบคุมความเร็วได้มากกว่านี้ สามารถทำได้โดยต่อวงจรหารดิจิทัลวางไว้ก่อนอินพุตเข้าของเฟสดีเทกเตอร์ หรือช่อง PD inp แต่ความเที่ยงตรงในการล็อกจะลดลงเป็นสัดส่วนโดยตรงกับอัตราการหารของวงจรหาร

3.ฟังชันถ่ายโอนของ VCO ซึ่งแท่นเป็นมอเตอร์ต่อด้วยเอนโค๊ดเดอร์ที่เพลา และโมเมนต์ความเฉื่อยของโรเตอร์จะเปรียบเสมือนกับ ฟังก์ชันถ่ายโอนของโลว์พาสฟิลเตอร์ ในการควบคุมความเร็วโดยปกติจะให้มีเรกูเลชั่นที่ดีแล้ว ต้องกำหนดให้มีโมเมนต์ความเฉื่อยมาก ซึ่งในกรณีควบคุมด้วยเฟสล็อกได้ทดลองโดยให้โมเมนต์ความเฉื่อยของโรเตอร์เหมาะสม โดยเทียบเป็นค่าคงที่เวลาเท่ากับ 0.3 วินาที ซึ่งผลลัพท์ที่ได้เมื่อในสภาวะล็อก มีเรกูเลชั่นความเร็วที่ดีมาก ซึ่งมีเพียงการเกิดเฟสเลื่อนโดยให้เหตุผลไว้ในข้อหนึ่ง

ดังนั้นกล่าวได้ว่า การควบคุมความเร็วของมอเตอร์ให้มีเรกูลชั่นที่ดี ไม่จำเป็นต้องใช้ไฟวีลมาช่วยในกรณีที่โมเมนต์ความเฉื่อยของโรเตอร์น้อยอีก ถ้าให้การควบคุมแบบเฟสล็อก

4.ความผิดพลาดในการควบคุมเนื่องจากการตอบรับคีย์บอร์ด มีเหตุผลดังนี้ คือตัวดิจิทัลคอนโทรลเลอร์นี้ ออกแบบมาให้ทำงานในการควบคุมลูปอย่างต่อเนื่องในโหมดการควบคุมที่เซตไว้เดิม เพราะได้คำนึงถึงในการควบคุมบางอย่างซึ่งไม่สามารถหยุดกระบวนการได้ ดังนั้นการเซตค่าพารามิเตอร์ต่างๆของดิจิทัลคอนโทรลเลอร์จากแป้นคีย์ จะไม่ทำให้หยุดการควบคุมอย่างเด็ดขาด แต่จะมีการรบกวนการทำงานในโหมดควบคุมบ้างเล็กน้อย เพราะไมโครโปรเซสเซอร์ต้องแชร์เวลาไปให้การตอบรับคีย์

5.ความผิดพลาดในการควบคุมเนื่องจากการสื่อสารอนุกรม กับโฮสคอมพิวเตอร์ ก่อนที่จะกล่าวถึงเหตุผลนี้ จอภาพของโฮสคอมพิวเตอร์มีข้อความแสดงว่า "SERIAL ATTEMPT" แสดงว่า ในขณะนี้การเซตพารามิเตอร์ของการสื่อสารอนุกรมของตัวดิจิทัลคอนโทรลเลอร์ ไม่ตรงกับของโฮสคอมพิวเตอร์ ดังนั้นเหตุผลนี้จึงไม่ใช่ข้อผิดพลาดในการควบคุมเนื่องจากการสื่อสารอนุกรม

โดยปกติระบบมอนิเตอร์ที่ใช้ในอุตสาหกรรมโดยทั่วไป ข้อมูลจะถูกแก้ไขทันกาล(update) ทุก 5 วินาที[7] เพราะว่ากระบวนการที่ใช้ในอุตสาหกรรมปัจจุบันจะเปลี่ยนแปลงช้า แต่ในกรณีที่ทดสอบนั้นเราให้ข้อมูลของโฮสคอมพิวเตอร์ถูกแก้ไขทันกาลเร็วที่สุด จะเป็นทุกๆ 20 มิลิวินาที เพื่อให้สังเกตการตอบสนองของระบบได้ดีที่สุด เพราะการทดลองเราใช้กระบวนการที่มีค่าคงที่ของเวลาน้อย อย่างไรก็ดีเวลาการแก้ไขทันกาลนั้น เราสามารถกำหนดได้ในตัวโปรแกรมโดยตรง เพื่อให้ผู้ใช้สะดวกในการสังเกตการตอบสนองของระบบ

มีข้อสังเกตบางอย่าง ถ้าโฮสต้องการข้อมูลแก้ไขทันกาลเร็ว จะต้องไม่ติดต่อกับตัวดิจิทัลคอนโทรลเลอร์ถี่ขึ้น ทำให้ไปรบกวนการทำงานได้ เท่าที่สังเกตจากการทดลอง ในโหมดควบคุมแบบเฟสล็อกจะเสียเสถียรภาพในการควบคุมมากกว่าโหมดทั่วไปคือ P,PI,และ PID

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความถี่ในการติดต่อดังกล่าวคือ ทุก 20 มิลิวินาทีจะไม่มีผลกับเสถียรภาพในโหมดควบคุมทั่วไป แต่จะมีผลกับโหมดเฟสล็อก แต่ถ้าความถี่ในการติดต่อทุก 1 วินาที จะไม่มีผลกับการควบคุมในทุกๆโหมด

2) ในโหมดการควบคุมทั่วไป

แบ่งเป็น โหมดการควบคุมต่างๆ คือ แบบ P,PI,PID สามารถจำแนกการสรุปได้ดังนี้

1.ในแต่ละโหมดควบคุมต่างก็ใช้สัปรูทีนเป็นของมันเอง ซึ่งในการควบคุมแบบ P จะจะมีจำนวนคำสั่งสั้นที่สุด และโหมดการควบคุมแบบ PID จะมีจำนวนคำสั่งในสัปรูทีนมากที่สุด ดังนั้นทำให้ การควบคุมแบบ PID ในแต่ละรอบใช้เวลานานกว่า การใช้เวลาได้จากการทดลองคือ

ในโหมดการควบคุม P ใช้เวลาในแต่ละรอบ 17 มิลลิวินาที

ในโหมดการควบคุม PI ใช้เวลาในแต่ละรอบ 20 มิลลิวินาที

ในโหมดการควบคุม PID ใช้เวลาในแต่ละรอบ 21 มิลลิวินาที

ซึ่งรอบหนึ่งก็เปรียบเหมือนการแซมปลิงครั้งหนึ่ง ดังเพื่อให้โหมดการควบคุมทั่วไป สามารถทำงานได้อย่างเสถียรภาพ ตัวกระบวนการ(process) ต้องมีค่าคงที่เวลามากกว่า 20 มิลลิวินาที หรือตามหลักการของ DSP ค่าคงที่เวลาต้องมากกว่าสองเท่า นั้นคือ 40 มิลลิวินาที

2.อัลกอลิธึมในการควบคุมแบบทั่วไปนั้น สามารถหาได้ตามหนังสือเกี่ยวกับระบบควบคุมทั่วไป ดังนั้นจึงไม่จำเป็นต้องทำวิจัยกันใหม่ เหมือนดังเช่นโหมดการควบคุมแบบเฟสล็อก ดังนั้นเกี่ยวกับโหมดควบคุมแบบทั่วไปนั้น จึงได้นำมาใส่ไว้ในตัวดิจิทัลคอนโทรลเลอร์ เพื่อให้สมบูรณ์ในการใช้งานของมันเท่านั้น

3.ผลการตอบสนองของโหมดควบคุมทั่วไปนั้นเป็นไปตามทฤษฎีทุกประการ ถึงแม้จะมีค่าคาดเคลื่อนบ้างเล็กน้อย ก็เป็นเพราะว่าตัวดิจิทัลคอนโทรลเลอร์ใช้ไมโครโปรเซสเซอร์ที่ใช้เป็น 8 บิต ดังนั้นความสามารถในทางคณิตศาสตร์จึงค่อนข้างต่ำ แต่ถ้าใช้ไมโครโปรเซสเซอร์ 16 บิตความคาดเคลื่อนเหล่านี้จะลดลง อย่างเช่นจำนวนจำกัดของเรจิสเตอร์ในรูทีนการคูณ 32 บิตเป็นต้น ดังนั้นความละเอียดของการคูณต้องลดลงมา เป็นต้น

4.ค่าต่างๆที่วัดจากการตอบสนองระบบ เช่น ไรส์ไทม์ จะแตกต่างจากทางทฤษฎี เพราะเมื่อใช้ ค่าเกนสูงเกินไป จะทำให้เอาต์พุตคอนโทรล (CV) เกิดอิ่มตัว(saturate) ณ ค่าแรงดันของแหล่งจ่ายของวงจร DAC เช่น 10 โวลต์เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.1.2 สรุปการทดสอบกับพล้านออร์เดอร์หนึ่ง ระหว่าง PLL กับ PI

จากการทดสอบเปรียบเทียบระหว่างโหมดการทำงานแบบ PLL และ แบบ PI ในพล้านต์ที่มีฟังก์ชันถ่านโอนเป็นแบบดีเลย์ออร์เดอร์หนึ่ง จากผลจะเห็นได้ว่า การตอบสนองสเต็ปของพล้านต์ในโหมดการควบคุมทั้งสองจะให้ผลแตกต่างกันไป ดังนี้

ถ้าค่าคงที่เวลาของพล้านเท่ากับ 0.33 วินาที โดยเซตพารามิเตอร์ในโหมดความคุมทั้งสองให้ออฟติมัม โดยโหมด PI มีพารามิเตอร์ Kp = 4.0 และ Ki = 20.0 และโหมด PLL จะเซตตัวโลว์พาสฟิลเตอร์เป็น LF = (1+0.39s)/(1+0.72s) ผลการทดสองจะได้ว่าทั้งคู่สามารถทำให้ออฟเซตเออเรอร์เท่ากับศูนย์ โดยที่โหมด PI มีค่าเซตตลิงไทม์ดีกว่า และโหมด PLL จะให้ค่าไรส์ไทม์ดีกว่า

ถ้าค่าคงที่ของพล้านแปรค่าตั้งแต่ 0.022 วินาที ถึง 1.0 วินาที โดยพารามิเตอร์ในการควบคุมของทั้งสองโหมดเท่าเดิม ซึ่งการทดสอบนี้มีจุดมุ่งหมายเพื่อทดสอบว่า ถ้าหากค่าคงที่ของเวลาของพล้านแปรเปลี่ยนไปจะค่าอ้างอิงคือ 0.33 วินาที จะมีผลการตอบสนองของทั้งสองโหมดอย่างไร โดยสามารถสรุปได้ คือ ในโหมด PI การตอบสนองเมื่อพล้านต์เปลี่ยนแปลง การตอบสนองของโอเวอร์ชูต อันเดอร์ชูต การแดมป์ ไรส์ไทม์ ต่างเหล่านี้ จะมีการแปรผันอย่างเห็นได้ชัด ตามรูปที่ 6-39 ทำให้ในทางปฏิบัติต้องปรับเปลี่ยนพารามิเตอร์การควบคุมอยู่เสมอเพื่อให้ระบบออฟติมัม ซึ่งต่างกับโหมด PLL จะให้การตอบสนองของพล้านต์ไม่มีการแปรผันมากนัก ในช่วงค่าคงที่ของเวลาของพล้านที่แปรเปลี่ยนค่าหนึ่ง ตามรูปที่ 6-38 สังเกตเส้นการตอบสนองจะมีลักษณะจะเปลี่ยนแปลงน้อย ซึ่งประเดนนี้ทำให้เห็นข้อได้เปรียบบางอย่างของการควบคุมโหมดเฟสล็อก

7.1.3 สรุปผลการทดสอบโปรแกรมพล้านต์มอนิเตอริง

โปรแกรมพล้านต์มอนิเตอริง เป็นโปรแกรมที่รันบนเครื่องคอมพิวเตอร์ส่วนบุคคล(PC) เพื่อใช้เป็นโฮส (host)คอมพิวเตอร์ ซึ่งมีหน้าที่ดูแล(monitor)ตัวดิจิทัลคอนโทรลเลอร์ในเครือข่ายทั้งหมด ซึ่งมีข้อสรุปปลีกย่อยดังนี้

1.การออกแบบซอฟต์แวร์โปรแกรมหลักยูสเซอร์อินเตอร์เฟส คือตัวโปรแกรมสามารถตอบสนองการใช้งานของผู้ใช้ได้สะดวก โดยใช้หลักการของ pop-up pulldown menu ซึ่งวิธีนี้เป็นที่ยอมรับกันกว้างขว่างในวงการการเขียนโปรแกรม ซึ่งโปรแกรมพล้านต์มอนิเตอริงในสมบูรณ์ในแง่นี้มาก

การออกแบบซอฟต์แวร์นี้ ซอร์สโค๊ดออกแบบมาให้พัฒนาต่อหรือขยายขีดความสามารถได้ แม้ว่าตัวซอฟต์แวร์ลิสต์ค่อนข้างจะใหญ่ แต่ผู้ออกแบบจะจัดแบ่งไว้เป็นหมวดหมู่อย่างชัดเจน และได้อธิบายไว้พอสมควรในเนื้อเรื่องบทที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.โปรแกรมพล้านต์มอนิเตอริงนี้มี ข้อความช่วยเหลือ(help) พอสมควร ซึ่งถ้าไม่เข้าใจแล้วสามารถดูในเนื้อเรื่องบทที่ 5 ได้

3.ในโปรแกรมพล้านต์มอนิเตอริง จะให้ความสำคัญกับกราฟแนวโน้มข้อมูล (trend recording) มากที่สุด เพราะสามารถจะดูการตอบสนองของกระบวนการในลูป โดยการควบคุมของดิจิทัลคอนโทรลเลอร์ นั้นเป็นส่วนสำคัญ โดยสามารถกำหนดตัวแปรหรือพารามิเตอร์ต่างของกราฟต่าง เช่น เปลี่ยนแกนเวลา ได้ตั้งแต่ 0.1 นาที ถึง 99999 นาที หรือ กำหนดค่าเวลาในการสุ่ม เช่น การอ่านค่าเข้ามาเพื่อพล็อตกราฟแต่ละจุดทุกๆ 5 วินาทีเป็นต้น หรือ การเลือก แสดงผลกราฟแนวโน้มข้อมูลของสถานีไหน หรือ เลือกจำนวนกราฟแสดงผลจอภาพในแต่ละครั้ง เป็นต้น

อย่างไรก็ดีกราฟแนวโน้มข้อมูลยังมีความสามารถเด่นชัด คือ สามารถการเก็บข้อมูลในอดีต ลงเป็นไฟล์ได้ โดยไม่จำกัดช่วงเวลา ซึ่งขนาดของไฟล์จะแปรตามระยะเวลาเก็บข้อมูล

4.กราฟบาร์ ซึ่งเป็นออปชันหนึ่งของโปรแกรมนั้น เป็นการแสดงผลของลูปได้ โดยแสดงพร้อมกัน 12 ลูปในขณะเดียวกัน บนจอภาพเดียวกัน โดยผู้ใช้สามารถกำหนดพารามิเตอร์ได้เช่นกัน

5.อย่างไรก็ดีโปรแกรมนี้เป็นไม่สามารถ กำหนดพารามิเตอร์ส่งไปให้แก่ ตัวดิจิทัลคอนโทรลเลอร์ได้ เช่น เซตพอย์น เป็นต้น ดังนั้นการเซตพารามิเตอร์ต้องไปที่ตัวดิจิทัลคอนโทลเลอร์ในแต่ละสถานีนั้น เพราะว่าเราให้ความสำคัญกับการดูแล(monitor)ของโฮสคอมพิวเตอร์เท่านั้น

6.รหัสเฟรม หรือกระบวนไบต์ข้อมูลในการส่งข้อมูลอนุกรมนั้น ไม่ได้ใช้มาตราฐานของบริษัทใดๆ ดังนั้นตัวดิจิทัลคอนโทลเลอร์จึงไม่สามารถนำไปใช้เชื่อมโยงกับมาตราฐานอื่นได้ แต่อย่างไรก็ดีสามารถดัดแปลงเฟรมข้อมูลจะซอร์สโปรแกรม ให้สามารถเข้ากับมาตราฐานอื่นได้โดยไม่ยากนัก

7.2 ความเป็นไปได้ในการนำผลงานวิจัยและพัฒนาไปใช้ในการผลิตเชิงอุตสาหกรรม

เป็นการพัฒนาตัวดิจิทัลคอนโทรลเลอร์ในโหมดต่างๆ ในตัวเดียวกัน และระบบพล้านต์มอนิเตอริงนั้น จะเป็นประโยชน์มากในระบบอุตสาหกรรม ซึ่งมีกระบวนการในการผลิตที่ซับซ้อน เช่น ในอุตสาหกรรมเคมีภัณฑ์ เป็นต้น โดยปกติแล้ว ระบบพล้านต์มอนิเตอริง นี้จะใช้ในอุตสาหกรรมขนาดใหญ่ หรืออุตสาหกรรมที่มีมูลค่าสูง เท่านั้น เพราะว่าการติดตั้งระบบพล้านต์มอนิเตอริงนั้นต้องสั่งจากต่างประเทศ ซึ่งมีค่าใช้จ่ายมาก จึงไม่นิยมใช้กันในอุตสาหกรรมขนาดกลาง หรือต่ำกว่า อย่างไรก็ตามการที่เราได้พัฒนาได้เองนั้นทำให้ประหยัดค่าใช้จ่ายอย่างมาก ต่ออุตสาหกรรมขนาดเล็ก และขนาดกลางที่ต้องการนำระบบพล้านต์มอนิเตอริงมาใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.3 ข้อเสนอแนะและข้อคิดเห็น

เนื่องจากบทความนี้ได้เน้นการใช้พล้านเป็นแบบดีเลย์ออร์เดอร์หนึ่ง ซึ่งเป็นพื้นฐานกระบวนการโดยทั่วไป อย่างเช่น การควบคุมระดับเป็นต้น โดยเน้นเนื้อหาการเปรียบเทียบระหว่างโหมดควบคุมแบบพรอปพรอชันแนล-อินทีกรัล กับโหมดเฟสล็อก โดยควมคุมขบวนการข้างต้น

อย่างไรก็ดีในพล้านมิใช่จะเป็นแบบดีเลย์ออร์เดอร์หนึ่งเท่านั้น ซึ่งถ้าหากเป็นออร์เดอร์สูงกว่านี้จะเป็นอย่างไร ในกรณีดังกล่าวถ้าเป็น ออร์เดอร์สอง จำเป็นต้องการควบคุมแบบ PID หรือ ออร์เดอร์สูงกว่าสอง ต้องมีการเพิ่มวงจรชดเชย เป็นต้น ทำให้เกินความจำเป็นสำหรับหัวข้อที่จะศึกษาเกี่ยวกับเรื่องนี้

ท้ายสุดหวังอย่างยิ่งว่า ผู้อ่านคงจะได้ความกระจ่างเกี่ยวกับการศึกษาทางด้านการควบคุม โดยเฉพาะอย่างยิ่งการควบคุมในโหมดเฟสล็อก ถ้ามีข้อสงสัย หรือติชม ยินดีน้อมรับด้วยความเต็มใจเป็นอย่างสูง สวัสดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1.Dr Roland E.Best:" PHASE-LOCKED LOOPS Theory, Design and Application.",Copyright 1984 ,McGraw Hill,1984.

2.Curtis D. Johnson:" MICROPROCESSOR-BASED PROCESS CONTROL.",Copyright 1984.,Prentice-Hall,Inc.,Englewood Cliff,1984.

3.Kenneth J. Ayala:" The 8051 Microcontroller ARCHITECTURE, PROGRAMMING" and APPLICATIONS.",Copyright 1991., WEST BUBLISHING COMPANY.,1991.

4.Morris,Noel Malcolm. "Control engineering.",Copyright 1968,1974 and" 1983. McGraw-Hill Book Company(UK).

5.National Semiconductor "Linear databook".

6.Howard A, Raphael." Motor control by PLL can be achieved with a microprocessor. Software replaces the phase detector." Electronic Design 9. April 26,,1973.

7.R.C. Holland. " Microcomputers for process control ",Pergamon Press.

8. Garth Nash " Phased-Locked loop Design Fundamentals" Application Note AN-535 ,MOTOROLA Semiconductor Products Inc.

9. Izawa ,K. : *Introduction to Automatic Control* , Elsevier, New York, 1663.

10. โยธิน เปรมปราณีรัชต์ , เจตนา สิงห์สินธุ์ " ระบบเซอร์โวควบคุมตำแหน่งด้วยเทคนิคของเฟสล็อกลูป " วารสารการประชุมวิชาการวิศวกรรมไฟฟ้าครั้งที่ 12 , พ.ศ. 2532

11. โยธิน เปรมปราณีรัชต์ , มนตรี พรรณรัตน์ "ระบบควบคุมความเร็วแบบสี่ควอดแรนท์ด้วยเฟสล็อกลูป" วารสารการประชุมวิชาการวิศวกรรมไฟฟ้าครั้งที่ 12 , พ.ศ. 2532

12. โยธิน เปรมปราณีรัชต์ , เจตนา สิงห์สินธุ์ " การพัฒนาตัวควบคุม PFD เพื่อปรับปรุงการล็อคในระบบเซอร์โวควบคุมตำแหน่งให้เร็วขึ้น " วารสารการประชุมวิชาการวิศวกรรมไฟฟ้าครั้งที่ 13 ,พ.ศ. 2533

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

กระบวนการดึงเข้า(pull-in)ของดิจิทัล PLL โดยใช้เฟสดีเทกเตอร์แบบ 4

ในขบวนการล็อกของ PLL โดยใช้เฟสดีเทกเตอร์แบบ 4 ซึ่งได้บรรยายไว้ในหัวข้อที่ 4-1-2 อ้างถึงสมการที่ 4-4 ขณะอยู่ในสภาวะไม่ล็อกของ PLL เฟสดีเทกเตอร์แบบ 4 จะสร้างสัญญาณเอาต์พุตเป็นสัดส่วนกับออฟเซตของความถี่ $\Delta\omega = \omega_1 - \omega_2$ ที่ซึ่ง ω_1 คือความถี่อ้างอิงและ ω_2 คือความถี่ป้อนกลับจาก VCO

ในภาคผนวกนี้เราจะวิเคราะห์ในรายละเอียดของ PD ในการปฏิบัติในสภาวะไม่ล็อก โดยเฉพาะเราจะเสนอรูปแบบทางคณิตศาสตร์ที่ได้รับมาจากสมการที่ 4-4 และ 4-5 ซึ่ง DPLL ได้บรรยายตามรูปที่ 4-4b เพื่อให้ง่ายขึ้นเราสมมุติว่า N = 1 และยิ่งไปกว่านั้น PLL อยู่ในเงื่อนไขเริ่มต้นในสภาวะไม่ล็อกซึ่ง $\omega_1 > \omega_2$ ดังรูปคลื่นที่แสดงในรูปที่ 4-3c เฉพาะพัลส์เอาต์พุต UP ของ PD เท่านั้น ในขณะที่เอาต์พุต DOWN ยังคงเป็นเป็นสภาวะ HIGH ตลอดเวลา(ไม่ทำงาน) เราควรจำไว้ว่าเอาต์พุต UP เป็นพัลส์ LOW (แอกทีฟ) โดยขอบขาลงของสัญญาณ u_1 และเอาต์พุตพัลส์ UP เป็นพัลส์ HIGH โดยขอบขาลงของสัญญาณ u_2

ขณะนี้สมมุติว่าสัญญาณ u_1 กระทำในขณะขอบขาลงของสัญญาณ ณ เวลา t = 0 ซึ่งแสดงในรูป A-1 ซึ่งการเปลี่ยนแปลงพัลส์ของสัญญาณเอาต์พุตของ PD อยู่ในสภาวะ LOW มีคำถามที่เกิดขึ้นตอนนี้คือ "เมื่อไรที่ขอบขาลงของสัญญาณ u_2 จะเกิดขึ้น" เพราะว่า PLL ในสภาวะไม่ล็อกสัญญาณ u_2 จะยังไม่ซิงโครไนซ์กับสัญญาณ u_1 ทั้งหมดนี้เรารู้ว่าความถี่ของ u_2 คือ $f = \omega_2 / 2\pi$ หรือคาบเวลาของ u_2 คือ $T_2 = 1/f_2$ ดังนั้นเราสามารถกล่าวได้ว่าขอบขาลงของสัญญาณ u_2 ครั้งต่อไปจะเกิดขึ้นภายในช่วงเวลา $0 \le t \le T_2$

ความน่าจะเป็น $w(t)$ ของขอบขาลงของสัญญาณ u_2 ครั้งต่อไปจะเกิดขึ้นภายในช่วงเวลา $0 \le t \le T_2$ เพราะฉะนั้น

$$w(t) = 1/T_2 \qquad 0 \le t \le T_2 \tag{A-1a}$$

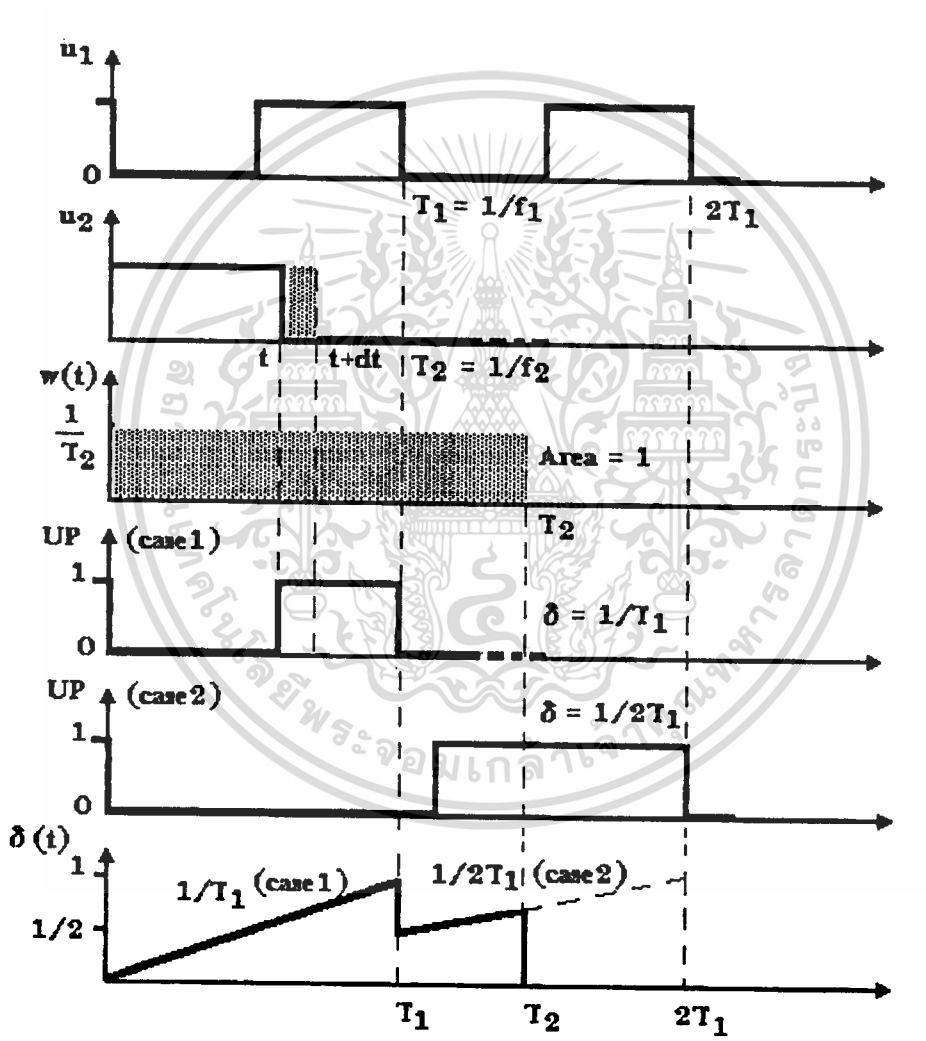
และแน่นอน

$$w(t) = 0 \qquad t \ge T_2$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชันความน่าจะเป็นของความหนาแน่น $w(t)$ แสดงเป็นเส้นโค้ง C ในรูปที่ A-1 ในตัวอย่างตามรูป A-1 คาบเวลา T_2 ถูกเลือกให้มากกว่า T_1 = 1/ T_1 แต่น้อยกว่า $2T_1$ ดังนั้นเราสามารถแยกแยะได้ 2 กรณีด้วยกัน

1. ขอบขาลงของสัญญาณ u_2 ครั้งต่อไปเกิดขึ้นภายในช่วงเวลา $0 \le t \le T_1$ (ทำนายจากเส้นโค้ง d ของรูป A-1)



รูปที่ A-1 คำนวณหาค่าเฉลี่ยของอัตราส่วนดิวตี้ไซเคิล δ ของสัญญาณเอาต์พุตของ PD ระหว่างในขบวนการดึงเข้า(pull-in) (a) รูปคลื่นของสัญญาณอ้างอิง u_1 (b) รูปคลื่นของสัญญาณเอาต์พุต VCO คือ u_2 (c) การพล็อตกราฟค่าเป็นไปได้ w(t) ของเหตุการณ์ของขอบขาลงของ $u_2(t)$ ครั้งต่อไป (d) รูปคลื่นของสัญญาณ UP ของ PD แบบ 4 ในกรณีที่ 1 (e) เหมือนกับ d แต่เป็นกรณีที่ 2 (f) คือการพล็อตค่าอัตราส่วนดิวตี้ไซเคิล δ ของสัญญาณ UP ของ PD ต่อเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ขอบขาลงของสัญญาณ u_2 ครั้งต่อไปเกิดขึ้นภายในช่วงเวลา $T_1 \le t \le T_2$ (ทำนายจากเส้นโค้ง e ของรูป A-1)

เราจำได้ว่าสัญญาณเอาต์พุตเฉลี่ย $\overline{u}_d'$ ของ PD แบบนี้คือเท่ากับอัตราส่วนของดิวตี้ไซเคิลของรูปคลื่น u_2 ซึ่งอัตราส่วนของดิวตี้ไซเคิลต่อมาได้ถูกเรียกว่า $\delta(t)$ ถูกคำนวณเหมือนกับฟังก์ชันของเวลาในเวลาเดียวกับที่ขอบขาลบของสัญญาณ u_2 ครั้งต่อไปเกิดขึ้น ในกรณีที่ 1 ดิวตี้ไซเคิลในเป็น อย่างชัดเจน ถ้าขอบขาลงเกิดขึ้นทันทีที่หลังจากที่ $t = 0$, $\delta(t)$ เข้าใกล้ศูนย์ แต่ถ้าขอบขาลงเกิดขึ้นก่อนเวลา $t = T_1$ เล็กน้อย $\delta(t)$ จะเข้าใกล้ค่ายูนิตี้ แสดงดังเส้นโค้ง f ตามรูป A-1

อย่างไรก็ดีตามกรณี 2 คือเมื่อขอบขาลงของสัญญาณ u_2 ครั้งต่อเกิดขึ้น *หลังจาก* t = T_1 ค่า $\delta(t)$ จะมากกว่า 50 เปอร์เซ็นต์เล็กน้อยเพราะว่าสัญญาณเอาต์พุต UP จะถูกเซตเป็น LOW หลังจาก $t = 2T_1$ แต่ค่า $\delta(t)$ จะเข้าสู่ยูนิตี้อีกครั้งเมื่อเกิดขอบขาลงของสัญญาณ u_2 อีกครั้งเมื่อเวลา t น้อยกว่า $2T_1$ เล็กน้อย ซึ่งในกรณีที่ 2 นี้อัตราส่วนของดิวตี้ไซเคิลนี้จะเท่ากับ $\delta(t) = t / 2T_1$

ขณะนี้เราอยู่ในตำแหน่งที่คำนวณหาอัตราส่วนดิวตี้ไซเคิลเฉลี่ยของสัญญาณเอาต์พุต UP สำหรับสถานะที่ $T_1 < T_2 < 2T_1$ ในกรณีที่ T_2 นานกว่า T_1 แต่สั้นกว่า $2T$ บนพื้นฐานทางสถิติได้ดังนี้

$$\overline{\delta} = \int_0^{T_2} \omega(t)\delta(t)dt \qquad \text{(A-2)}$$

เหมือนกับเส้นโค้ง f ในรูปที่ A-1 การอินทีกรัลแบ่งเป็น 2 ส่วนด้วยกัน

$$\overline{\delta} = \int_0^{T_1} \omega(t)\frac{t}{T_1}dt + \int_{T_1}^{T_2} \omega(t)\frac{t}{2T_1}dt$$

เราพิจารณาในขณะนี้ว่าในกรณีทั่วไป T_2 ไม่ได้ถูกจำกัดอยู่ในช่วง $T_1 < T_2 < 2T_2$ แต่ค่อนข้างน่าจะมีค่าอยู่ระหว่าง T_1 และ nT_1 ที่ซึ่ง n คือค่าตัวเลขจำนวนเต็ม แล้วฟังก์ชัน $\delta(t)$ จะประกอบด้วยเป็นเหมือนรูปคลื่นฟันเลื่อย แต่มีถึง n เซกเมนต์ด้วยกัน ตามรูปที่ A-2 เราจะได้สมการเป็น n เทอมดังนี้

ห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\bar{\delta} = \int_0^{T_1} \omega(t)\frac{t}{T_1}dt + \int_{T_1}^{2T_1} \omega(t)\frac{t}{2T_1}dt + \ldots + \int_{(n-1)T_1}^{T_2} \omega(t)\frac{t}{nT_1}dt \qquad \text{(A-3)}$$

ซึ่งจากสมการที่ (A-3) เราได้สมการใหม่เป็น

$$\bar{\delta} = \frac{f_2}{2f_1}\left(n - \sum_{i=1}^{n}\frac{1}{i}\right) + \frac{f_1}{2nf_2} \qquad f_1 > f_2 \qquad \text{(A-4)}$$

ในสมการ(A-4) n ถูกกำหนดโดย $n = \text{int}(f_1/f_2) + 1$ นั่นคือ n คือจำนวนเต็ม(integer)ที่ซึ่งมีค่ามากกว่า f_1/f_2 (เช่น f_1/f_2 = 8.36) ซึ่งสมการ A-4 จะพล็อตเป็นกราฟตามรูป A-3 โดยสมมุติว่า $f_2 < f_1$ เหตุนี้สมการจึงยอมให้เส้นโค้งอยู่ในช่วง $0 \le f_2/f_1 \le 1$ โดยยังคงเหลือส่วนของเส้นโค้งถูกทำให้การรวมกันของบางอากิวเมนต์ มันเป็นการง่ายที่จะแสดงในกรณีว่า f_2 ยังคงโปัฉจรึงเสมิย

$$\bar{\delta}\left(\frac{f_2}{f_1}\right)\Big|_{\text{upper tree}} = -\bar{\delta}\left(\frac{f_1}{f_2}\right)\Big|_{\text{lower tree}} \qquad \text{(A-5)}$$

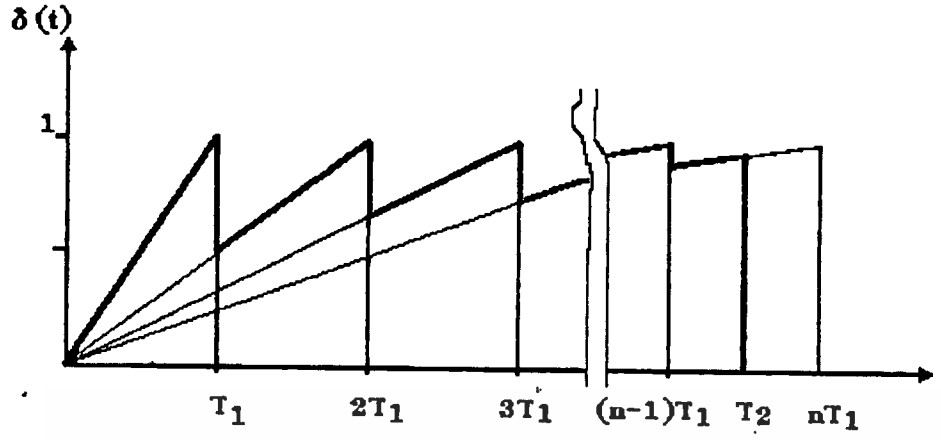
ถ้า f_2/f_1 ให้เข้าสู่ค่า 1 จากทางซ้ายมือทำให้ $\delta(t)$ กลายเป็น 0.5 และถ้า f_2/f_1 เข้าใกล้ 1 จากทางขวามือทำให้ $\delta(t)$ กลายเป็น -0.5

ฟังก์ชัน $\delta(f_2/f_1)$ ค่อนข้างจะยุ่งยากดังแสดงดังรูป A-3 ซึ่งไม่ได้ปรับปรุงให้เหมาะสมสำหรับการคำนวณของขบวนการดังเข้า(pull-in) การประมาณค่าอย่างหยาบๆจึงได้เส้นโค้งซึ่งเป็นไปได้ตามรูปที่ A-3 ถูกแทนโดยเส้นตรง(ตามเส้นประดังรูปที่ A-3) สำหรับการประมาณค่าแบบนี้เราได้

$$\bar{\delta} = \frac{f_1 - f_2}{f_1} = \frac{\omega_1 - \omega_2}{\omega_1} \qquad \text{(A-6)}$$

ค่าสัญญาณเอาต์พุตเฉลี่ยของ PD สามารถคำนวณได้ ถ้าแหล่งจ่ายไฟแรงดันโวลต์ของ PD คือ U_B เพราะฉะนั้นสัญญาณเอาต์พุตเฉลี่ยคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ A-2 *เป็นการพล็อตค่าอัตราส่วนของดิวตี้ไซเคิล* $\delta(t)$ *ต่อเวลาสำหรับในกรณีทั่วไป*

$$\overline{u}_d = \overline{\delta} U_B = \frac{U_B}{\omega_1}(\omega_1 - \omega_2) \qquad \text{(A-7)}$$

ดังนั้นค่า $\overline{u}_d$ จึงเป็นสัดส่วนโดยตรงกับออฟเซตของความถี่ $\Delta\omega = \omega_1 - \omega_2$ และ U_B / ω_1 สามารถกำหนดเป็นค่าเกนของเฟสดีเทกเตอร์ในสภาวะไม่ล็อก(unlocked) K'_d

$$K'_d = U_B / \omega_1 \qquad \text{(A-8a)}$$

ถ้า PLL ทำงานใกล้ค่าความถี่กลาง $\omega_1 \approx \omega_0$ และสมการ(A-8a)กลายเป็น

$$K'_d \approx U_B / \omega_1 \qquad \text{(A-8b)}$$

ดังนั้นค่าเกนแฟกเตอร์ K'_d ไม่ได้ปนกันกับค่าเกนของเฟสดีเทกเตอร์ K_d ที่กล่าวไว้ในเนื้อหาบทตอนต้น(ในสภาวะล็อกของPLL)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

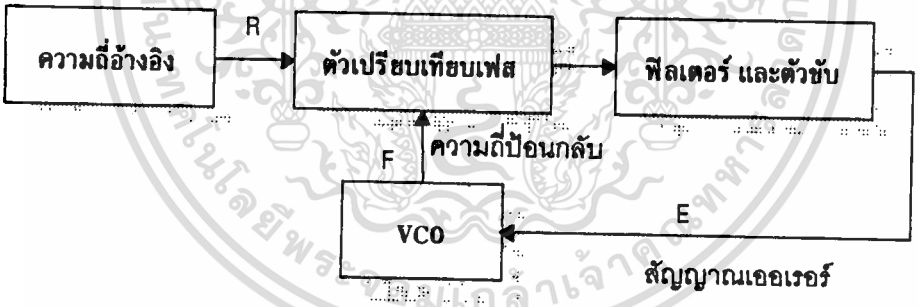
มอเตอร์ควบคุมโดยเฟสล็อกลูปบนไมโครโปรเซสเซอร์ โดยใช้ซอฟต์แวร์เลียนแบบเฟสดิเทกเตอร์

Howard A. Raphael [6]

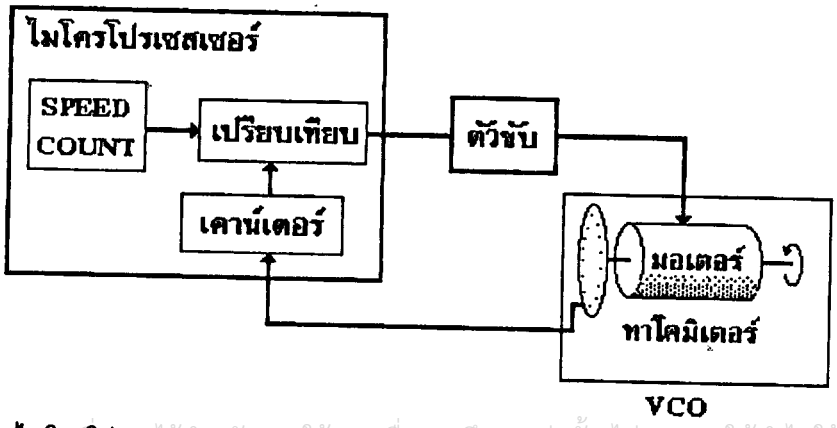
กล่าวนำ

ไมโครโปรเซสเซอร์สามารถเสนอความได้เปรียบที่สำคัญ ในการใช้เฟสล็อกมาใช้เป็นตัวคอนโทรลเลอร์ในการควบคุม เช่น ใช้ควบคุมความเร็วมอเตอร์ เป็นต้น แม้ว่าจะผิดแผกจากการคอนโทรลแบบอื่น แต่ประโยชน์ที่ได้รับจะการปรับปรุงทางดิจิทัลตามแนวทางหลักดังนี้

เฟสล็อกลูปแบบดิจิทัล จะมีการปฏิบัติคล้ายกับลักษณะพัลส์วิทด์มอดูเลชั่น(PWM) โดยใช้ฟังก์ชันการทำงานแบบนับและเปรียบเทียบ(ไม่ใช้คำนวณทางคณิตศาสตร์ แต่ใช้ความสามารถทางการเชื่อมโยงอินพุตและเอาต์พุตนั้น เป็นลักษณะที่น่าจะเหมาะสมกับลักษณะงานต่าง เกี่ยวข้องกับเฟสล็อก



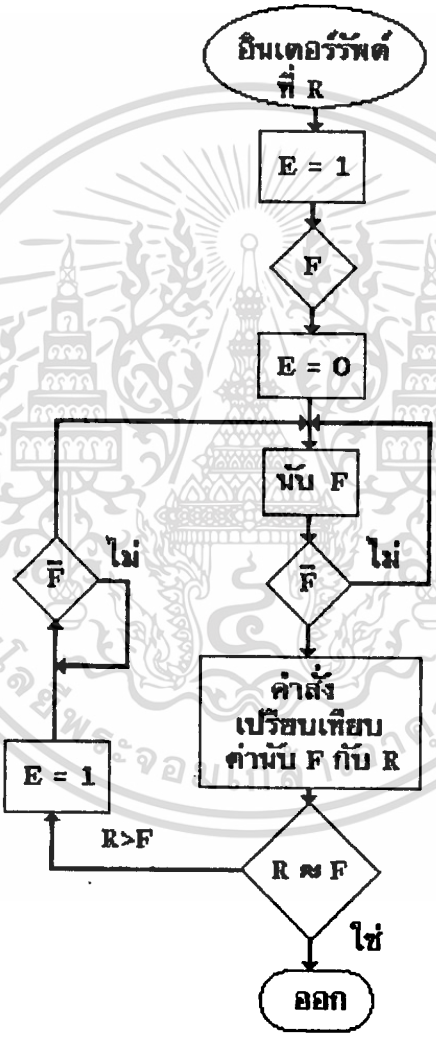
รูปที่ B-1 *เฟสล็อกลูปตามแบบฉบับ ความถี่ VCO จะแทรคกิ้งตามความถี่อ้างอิง สำหรับการควบคุมมอเตอร์ VCO จะแทนมอเตอร์ และ ทาโคมิเตอร์*



รูปที่ B-2 *ไมโครโปรเซสเซอร์จะดำเนินการเปรียบเทียบเฟสและความถี่ภายใต้คำสั่งโปรแกรม*

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เฟสล็อกลูปประกอบด้วยตัวกำเหนิดความถี่อ้างอิง ตัวเฟสคอมพาราเตอร์ ลูปฟิลเตอร์ และโวลต์เตจคอนโทรลออสซิลเลเตอร์ ตามรูปที่ B-1 วงจร VCO จะกำเนิดสัญญาณความถี่ตามสเต็ปของสัญญาณอ้างอิง ลูปจะเริ่มต้นแก้ไขให้ถูกต้องเมื่อเฟสเอาต์พุตเลื่อนไปเสมอจากความถี่อ้างอิง โดยที่โวลต์เตจแก้เออเรอร์เฟสคำนวณโดยตัวเปรียบเทียบความถี่-เฟส(phase-frequentcy comparator) ซึ่งเอาต์พุตโวลต์เตจทำให้ VCO จัดเอาต์พุตใหม่ตามความถี่อ้างอิง

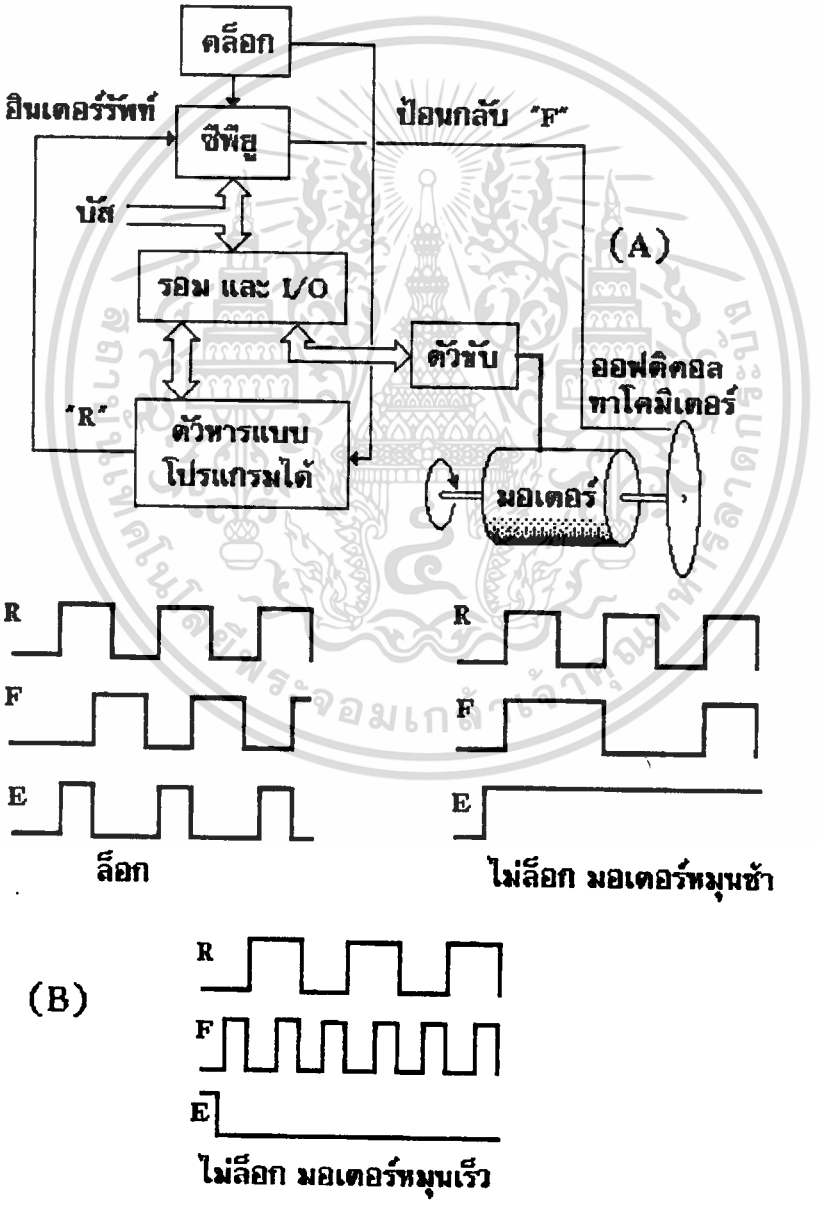


รูปที่ B-3 *สัญญาณควบคุมการทำงานดิวตี้ไซเคิล ถูกกำเหนิดเมื่อ R = F เป็นการควบคุมเฟส ส่วนล่างของโฟรว์ชาร์ตเป็นการเป็นเออเรอร์แบบหยาบโดยการนับสัญญาณ F และนำมาเปรียบเทียบ*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VCO เลียนแบบ มอเตอร์

ในการใช้งานควบคุมมอเตอร์ ตัวมอเตอร์และทาโคมิเตอร์(motor-tachometer)ทำหน้าที่แทน VCO การออกแบบต้องการเป็นสองโหมดเป็นลักษณะอินล็อก(in-lock) และเอาต์ล็อก (out-lock) ในลักษณะแรก VCO มีเอาต์พุตเพียงแต่ต่างเฟสกันซึ่งความถี่หรือความเร็วถูกต้อง จะถูกแก้ไขโดยเฟสดีเทกเตอร์โดยให้เอาต์พุตซึ่งเป็นพัลส์มีดิวตี้ไซเคิลน้อยออกไป แต่ถ้าเอาต์พุต VCO มีความถี่ต่างกันมากหรือโหลดของมอเตอร์เปลี่ยนไปทันทีทันใด ก็เฟสดีเทกเตอร์จะแก้ไขแบบอยาบเพื่อให้ความเปลี่ยนเร็วที่สุดเท่าที่เป็นไปได้



รูปที่ B-4 *ไมโครโปรเซสเซอร์ ถูกอินเตอร์รัพต์ (a) ที่ขอบนำของ R คำนวณ PWM คอนโทรล (b) คลิสตอลภายในเตรียมคาบเวลา มุมเฟสวัดระหว่างขอบขาขึ้นของ R ,F*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้วยความสามารถของไมโครโปรเซสเซอร์ในขบวนการคำนวณ อ้างได้ว่ามีค่าเออเรอร์น้อยกว่า 0.1 เปอร์เซ็นต์รวมทั้งจังหวะเวลาและคำสั่งด้านซอฟต์แวร์

มอเตอร์ ทาโคมิเตอร์ และตัวขับแทนเป็น VCO และ โลว์พาสฟิลเตอร์ของ PLL ตามแบบดังเดิม มอเตอร์ DC กับแม่เหล็กถาวร หรือมอเตอร์แบบฮอลเอฟเฟค(Hall-effect)มีลักษณะคุณสมบัติเป็น VCO ที่มีคุณสมบัติเชิงเส้นสูงมาก ตัวอย่างเช่น ความสำพันธ์เชิงเส้นระหว่าง ความเร็ว ทอร์ก และ กระแสของมอเตอร์ที่นำมาประยุกต์ทดสอบมีอัตรากำลังน้อยกว่า 1/4 แรงม้า ซึ่งหามาใช้งานง่าย ถ้ามอเตอร์ต้องการใช้ความเร็วของมอเตอร์ค่าเดียว แล้วมอเตอร์แบบความเฉื่อย(inertia)สูงจะเป็นที่พึงประสงค์มากกว่า ซึ่งความเฉื่อยโรเตอร์จะลดความแปรผันของความเร็วได้มากกว่า แม้ว่าจะมีฟลายวีล(flywheel) ซึ่งถ้าเป็นมอเตอร์แบบแพนเคก(pancake)มาใช้ได้จะดีเยี่ยม

ในกรณีเซอร์โวแบบออร์เดอร์หนึ่ง นั้นต้องการส่วนป้อนกลับเป็นดิจิทัลทาโคมิเตอร์ต่อกับเพลาของมอเตอร์โดยตรง ความละเอียดของทาโคมิเตอร์คือจำนวนพลัลส์ต่อรอบในคาบเวลานั้นต้องพอเพียงกับ การใช้งานสำหรับความเร็วสูงจะใช้ค่อนข้างหยาบประมาณ 100 ถึง 1000 พัลส์ต่อรอบ ซึ่งถ้าเป็นความเร็วต่ำต้องใช้ประมาณ 100 พัลส์ต่อรอบ

ในทาโคมิเตอร์แบบออปติคอลในตัวหนึ่งทำด้วยแผ่นจานดิสก์ ซึ่งมีส่วนแสดงทะลุผ่านขนาดเท่ากับตัวกำเหนิดแสงและโฟโตทรานซิสเตอร์ แต่เพื่อความแน่ใจควรเลือกขบวนพัลส์ให้มีขนาดเท่ากันตลอด

ฟิลเตอร์ของ PLL แบบเดิมสามารถตัดทิ้งได้เพราะแทนด้วยความเฉื่อยของมอเตอร์สนับสนุนอยู่ ซึ่งจะลดความแปรเปลี่ยนอย่างกระทันหันของขบวนพัลส์จากสัญญาณควบคุมได้เหมือนกันกับโลว์พาสฟิลเตอร์

การดำเนินการของฟังก์ชันคำสั่งเปรียบเทียบระหว่างสัญญาณป้อนกลับและสัญญาณอ้างอิง ไมโครโปร-เซสเซอร์ต้องคำนวนความเปลี่ยนแปลงสัญญาณป้อนกลับจากทาโคมิเตอร์ ซึ่งจะเป็นสัดส่วนกับความเร็วของมอเตอร์เปรียบเทียบกับสัญญาณอ้างอิง โดยทำการนับคาบสัญญาณของทาโคมิเตอร์ ซึ่งทาโคมิเตอร์ต้องมีดิวตี้ไซเคิลเป็น 50เปอร์เซ็นต์ ตลอดช่วงความกว้างของความเร็วที่น่าสนใจ การนับนั้นนับเฉพาะครึ่งของคาบก็เพียงพอแล้ว

ผลลัพท์ของการนับสามารถนำมาเปรียบเทียบกับสัญญาณอ้างอิง ที่ซึ่งเป็นฟังก์ชันของความเร็วตามต้องการ ถ้าการนับนั้นมากกว่าความเร็วที่ต้องการในขณะที่มอเตอร์มอเตอร์หมุนช้าแล้ววงจรจะจ่ายกำลังให้กับมอเตอร์ทันที ถ้าการนับนั้นมากกว่าก็การจ่ายกำลังก็จะลดลง

หลักของการดำเนินการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความคาดเคลื่อนซึ่งยอมรับในการนับเกิดขึ้นในระหว่างขอบเขตของการล็อก(lock range) และเออ-เรอร์แบบหยาบสำหรับสภาวะไม่ล็อก(out off lock) ซึ่งเอาต์พุตจะให้เต็มอัตรากำลังเพื่อให้มอเตอร์รักษาสเต็ปตามสัญญาณอ้างอิง คาบเวลาที่แตกต่างกันมากระหว่างสัญญาณอ้างอิงกับคาบสัญญาณทาโคมิเตอร์จะส่งผลให้มอเตอร์ถูกจ่ายเต็มกำลัง หรือไม่จ่ายกำลังจนกระทั่งความแตกต่างจะเกิดขึ้น แล้วตกลงอยู่ในช่วงความคาดเคลื่อนที่ยอมรับ(tolerance band) ตามรูปที่ B-3 โดยที่หมายความว่ามอเตอร์จะถูกนำเข้าสู่สภาวะล็อกให้เร็วที่สุดเท่าที่เป็นไปได้ ลูปตอบสนองการเปลี่ยนในความถี่ที่อัตราพิกัด ซึ่งตอบสนองตามแบนวิทด์สัญญาณของมอเตอร์ โฟรว์ชาร์ตของคำสั่งของไมโครโปรเซสเซอร์ตามรูปที่ B-3 จะรับสนองการกระทำดังกล่าว

ไซเคิลเวลาในการคำนวณ และจำนวนคำสั่งเป็นข้อจำกัดในความเร็วสูงสุดที่ซึ่งโปรเซสเซอร์สามารถรักษาการซิงโครไนซ์ ความเร็วที่จำกัดของไมโครโปรเซสเซอร์นั้นขึ้นอยู่กับไซเคิลของเวลาด้วย ตามรูปแบบ ถ้าชิปประกอบด้วยทรานซิสเตอร์ประเภทไบโพล่าจะมีไซเคิลเวลาประมาณ 200 นาโนวินาที ถ้าเป็น NMOS ประมาณ 2 uS และ PMOS จะประมาณ 5-10 ไมโครวินาที ถ้าคุณกำหนดการทำงานมอเตอร์ที่ความเร็ว 600 rpm หรือ 10 rps และทาโคมิเตอร์มี 100 พัลส์ต่อรอบคิดเป็นความถี่ 1 กิโลเฮิรตซ์ หรือ คาบเวลา(period)เป็น 1 มิลลิวินาที ฉะนั้นเวลาโปรเซสเซอร์คำนวณคือครึ่งคาบเป็น 500 ไมโครวินาที ถ้าไมโครโปรเซสเซอร์มีไซเคิลเวลา(cycle time)เท่ากับ 200 nS จะสามารถปฏิบัติได้ 2500 คำสั่ง หรือถ้าไซเคิลเวลาเป็น 10 ไมโครวินาที จะทำได้ 50 คำสั่งในช่วงเวลา 500 ไมโครวินาที เมื่อ มอเตอร์หมุนความเร็วตามข้างต้น ส่วนคาบเวลาที่เหลือไมโครโปรเซสเซอร์สามารถทำงานอื่นได้อีก การวิเคราะห์ดังกล่าวสมมุติว่ามีความต่างเฟสค่าใดค่าหนึ่ง

สัญญาณอ้างอิงจากภายใน และภายนอก(Implicit or Explicit Reference)

เทคนิค explicit-reference คือหนึ่งในสองเทคนิคเหมาะสมสำหรับเป็นเครื่องมือสำหรับไมโครโปร-เซสเซอร์ แหล่งกำเนิดสัญญาณอ้างอิงจากภายนอก เช่นเคาท์เตอร์แบบโปรแกรมได้ซึ่งเป็นตัวจัดการขบวนพลัสส์เพื่อให้มอเตอร์ล็อกเข้าสู่ ส่วนรูปแบบของ implicit คือโปรเซสเซอร์ต้องคำนวนคาบเวลาของการหมุนที่ต้องการแล้วบังคับให้มอเตอร์ทำตาม ซึ่งจะไม่มีการใช้มุมเฟส

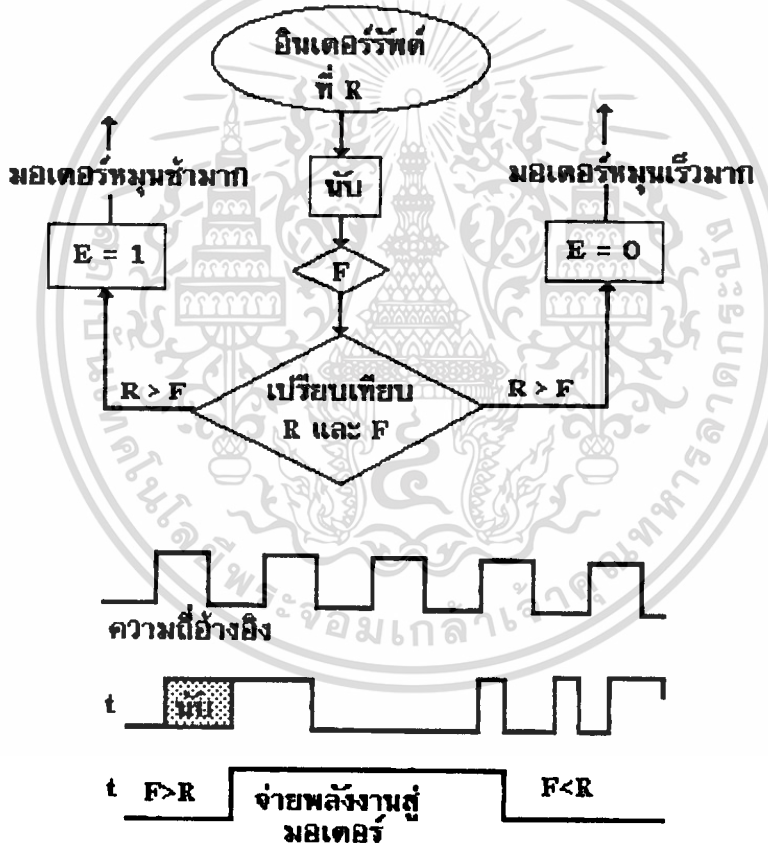
กรณีแรกที่พิจารณาเป็นเทคนิค explicit (แสดงตามรูป B-4a) สัญญาณคล็อกอ้างอิงให้สัญญา-ลักษณ์เป็น R และสัญญาณคล็อกป้อนกลับให้เป็น F ความเร็วซิงโครไนซ์จะเกิดขึ้นเมื่อ $f_R = f_F$ แม้ว่าเฟสที่มีอยู่ต่างกันก็ตาม เฟสต่างกันถูกวัดได้จากถูกวัดได้จากขอบขาขึ้น(leading egde) ของสัญญาณ R และขอบขาขึ้นของสัญญาณ F โปรเซสเซอร์จะเปิดสัญญาณเอาต์พุต E ที่ขอบขาขึ้นของสัญญาณ R และจะปิดสัญญาณ E เมื่อถึงขอบขาขึ้นของ F ซึ่งการปฏิบัติดังกล่าวสอดคล้องกับส่วนบนของโฟรว์ชาร์ตตามรูปที่ B-2 ถ้ามอเตอร์มีโหลดเพิ่มดังนั้นมอเตอร์ช้าลงแล้วความกว้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของเออเรอร์พัลส์มากขึ้นด้วย ในขณะนี้จะเพิ่มกำลังเฉลี่ยให้กับมอเตอร์มากขึ้น หรือถ้าโหลดลดลง แล้วความเร็วเพิ่มก็จะลดกำลังเฉลี่ยลงเช่นกัน

สำหรับสภาวะไม่ล็อก(out-of-lock) โปรเซสเซอร์จะนับเพื่อวัดค่าคาบเวลาของสัญญาณ F และเปรียบเทียบกับค่าสัญญาณ R ที่เป็นอยู่ ผลลัพท์จะมีความแตกต่างกับมากซึ่งสัญญาณเอาต์พุต E จะอยู่ในสถานะเปิดหรือปิดอย่างเดียวเพื่อจะนำมอเตอร์เข้าสู่การล็อก(ส่วนล่างของรูปที่ B-2)

สำหรับการคำนวนการนับโปรเซสเซอร์จะทำคำสั่งลูปของซอฟแวร์ กับจำนวนคำสั่งที่คงตัวในแต่ละลูปคำสั่งนั้นเล็กและคงที่ ซึ่งจำนวนลูปที่นับได้โดยเพิ่มขึ้นทีละหนึ่งนั้นก็จะเท่ากับความกว้างของคาบเวลาของสัญญาณ F



รูปที่ B-5 *ถ้าเกี่ยวข้องเฉพาะความเร็วมอเตอร์ ค่าอ้างอิงภายในสามารถเปรียบเทียบกับคาบของทาโคมิเตอร์ คล้ายกับการทำงาน PLL เมื่อ out-of-lock*

มันไม่น่าเป็นไปได้ซึ่งการนับสำหรับ F และ R จะมีโอกาสตรงกันอย่างแน่นอน เพราะฉะนั้นค่าความคลาดเคลื่อน(tolerance)ถูกใช้เพื่อว่า F เท่ากับ R บวกหรือลบจำนวนนับ n ที่ซึ่ง n คือจำน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วนสำพันธ์กับการนับ 1/2 ของ F และ R ความละเอียดของจำนวนนับของทาโคมิเตอร์และการตอบสนองการเปลี่ยนโหลดของมอเตอร์กำหนดตามค่าคลาดเคลื่อนที่ต้องการใช้

สำหรับตัวอย่าง ถ้ามอเตอร์มีความเฉื่อย(inertia)โรเตอร์สูงมาก โหลดจะกำหนดเป็นผลของฟลายวีล(fly wheel) เพื่อลดการสายของความเร็ว ดังนั้นความเร็วของมอเตอร์จะเข้าใกล้ค่า R มากขึ้นในกรณีที่โหลดคงที่ ถ้ามอเตอร์มีความเฉื่อย(inertia)น้อยจะทำให้การรักษาซิงโครไนซ์นั้นยากมาก ดังนั้นการกำหนดค่าคลาดเคลื่อน(toerlance)กว้างขึ้น ถ้า $T_R > T_F$ แสดงว่ามอเตอร์เร็วมากต้องกำลังจ่ายให้ลดลงเพื่อให้มอเตอร์มีความเร็วอยู่ในช่วงค่าความคลาดเคลื่อน เช่นเดียวกันถ้า $T_R < T_F$ แสดงว่ามอเตอร์ช้าต้องเพื่อกำลังจ่ายเพื่อมอเตอร์เพิ่มความเร็วให้อยู่ในช่วง ความคลาดเคลื่อนที่กำหนดได้

สัญญาณการแก้ไขค่าเออเรอร์แบบต่อเนื่อง ไม่จำเป็นต้องใช้กับความเร็วซิงโคนัสสำหรับมอเตอร์ที่มีความเฉื่อยภายในสูง อีกอย่างหนึ่งผลของฟลายวีลมีผลต่อการลดการสายของความเร็วก็ทำให้ความเร็วคงที่ได้ ปราศจากความจำเป็นในการแก้ไขเฟสทำให้คล็อกเปรียบเทียบเฟสของสัญญาณอ้างอิงในไม่มีความจำเป็น ซึ่งใช้เฉพาะค่านับของความถี่อ้างอิงเท่านั้นที่ต้องการเพื่อรักษาความเร็ว

สัญญาณ F มีค่าเท่ากับการนับครึ่งคาบเปรียบเทียบกับสัญญาณอ้างอิง R ถ้า $R < F$ สัญญาณเอาต์พุตเออเรอร์จะถูกจ่าย ถ้า $R > F$ สัญญาณเอาต์พุตเออเรอร์จะไม่จ่ายออก (ตามรูปที่ 5) การจ่ายเอาต์พุตสลับกันไปมานั้นไม่เป็นสาเหตุให้ความเร็วมอเตอร์สายได้ การเปลี่ยนความเร็วในรูปแบบนี้ทำได้ง่าย โดยเปลี่ยนจำนวนนับอ้างอิง และเปรียบเทียบความแตกต่างนั้นเป็นผลได้ การเปลี่ยนทิศทางก็ทำได้ง่ายเช่นกันโดยสลับศักดาไฟฟ้าให้เป็นตรงกับข้าม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค.

ซอฟต์แวร์ลิสต์ภาษาแอมเซมบลี 8031 ของโปรแกรมของตัวดิจิทัลคอนโทรลเลอร์

โปรแกรมของตัวดิจิทัลคอนโทรลเลอร์ จะเขียนเป็นแอสเซมบลีของไมโครโปรเซสเซอร์เบอร์ 8031 ซึ่งอยู่ในตระกูล MCS-51 โดยการเขียนแอสเซมบลี นั้นตัวคอมไพล์เลอร์หรือ ตัวแอสเซมเบลอร์ จะใช้โปรแกรม CROSS-16 เพื่อแปลงไฟล์แอสเซมบลี เป็น เฮกซ์ไฟล์ ซึ่งเป็นรหัสตัวเลขฐานสิบหก ของ 8031

ซอฟต์แวร์ลิสต์จะแสดงรายละเอียดจำแนกออกเป็นส่วนดังนี้ เช่น

```
01F9 EE        MUL8C:      MOV     A,R6        ;Shift left of R7
```

จำแนกได้ดังนี้

01F9 เป็นแอดเดรสหน่วยความจำโปรแกรม (program memory)

EE เป็นรหัสตัวเลขฐานสิบหกของคำสั่งในบรรทัดนั้น

MUL8C: การกำหนดอ้างแอดเดรสในคำสั่ง

MOV A,R6 เป็นคำสั่งของแอมเซมบลี 8031

:Shift left of R7 เป็นคำอธิบาย(description)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ;-------------------------
                ; main program of digital controller
                ;-------------------------
0000                INCL  "header.asm"
                ;This is header routine. It used program memory area
                ;address between 100H to 2CEH (462 byte)
                ;*********************************************************
0000                CPU     "8051.TBL"
0000                HOF     "INT8"
                ;*********************************************************
                ;MCS-51 INTERNAL REGISTERS
00E0 =      A:      EQU     0E0H    ;ACCUMULATOR
00F0 =      B:      EQU     0F0H    ;B REGISTER
0083 =      DPH:    EQU     83H     ;DATA POINTER HIGH
0082 =      DPL:    EQU     82H     ;DATA POINTER LOW
00A8 =      IE:     EQU     0A8H    ;INTERRUPT ENABLE
00B8 =      IP:     EQU     0B8H    ;INTERRUPT PRIORITY
0080 =      P0:     EQU     80H     ;PORT 0
0090 =      P1:     EQU     90H     ;PORT 1
00A0 =      P2:     EQU     0A0H    ;PORT 2
00B0 =      P3:     EQU     0B0H    ;PORT 3
0087 =      PCON:   EQU     87H     ;POWER CONTROL REGISTER
00D0 =      PSW:    EQU     0D0H    ;PROGRAM STATUS WORD
0099 =      SBUF:   EQU     99H     ;SEND BUFFER
0098 =      SCON:   EQU     98H     ;SERIAL CONTROL
0081 =      SP:     EQU     81H     ;STACK POINTER
0088 =      TCON:   EQU     88H     ;TIMER CONTROL
008C =      TH0:    EQU     8CH     ;TIMER 0 HIGH
008D =      TH1:    EQU     8DH     ;TIMER 1 HIGH
008A =      TL0:    EQU     8AH     ;TIMER 0 LOW
008B =      TL1:    EQU     8BH     ;TIMER 1 LOW
0089 =      TMOD:   EQU     89H     ;TIMER MODE
0000 =      R0:     EQU     00H     ;
0001 =      R1:     EQU     01H     ;
0002 =      R2:     EQU     02H     ;
0003 =      R3:     EQU     03H     ;
0004 =      R4:     EQU     04H     ;
0005 =      R5:     EQU     05H     ;
0006 =      R6:     EQU     06H     ;
0007 =      R7:     EQU     07H     ;
00B2 =      INT0:   EQU     0B2H    ;
00B3 =      INT1:   EQU     0B3H    ;
                ;*********************************************************
0080 =      P0_0:   EQU     80H     ;PORT 0
0081 =      P0_1:   EQU     81H
0082 =      P0_2:   EQU     82H
0083 =      P0_3:   EQU     83H
0084 =      P0_4:   EQU     84H
0085 =      P0_5:   EQU     85H
0086 =      P0_6:   EQU     86H
0087 =      P0_7:   EQU     87H
                ;-------
0090 =      P1_0:   EQU     90H     ;PORT 1
0091 =      P1_1:   EQU     91H
0092 =      P1_2:   EQU     92H
0093 =      P1_3:   EQU     93H
0094 =      P1_4:   EQU     94H
0095 =      P1_5:   EQU     95H
0096 =      P1_6:   EQU     96H
0097 =      P1_7:   EQU     97H
                ;-------
00A0 =      P2_0:   EQU     0A0H    ;PORT 2
00A1 =      P2_1:   EQU     0A1H
00A2 =      P2_2:   EQU     0A2H
00A3 =      P2_3:   EQU     0A3H
00A4 =      P2_4:   EQU     0A4H
00A5 =      P2_5:   EQU     0A5H
00A6 =      P2_6:   EQU     0A6H
00A7 =      P2_7:   EQU     0A7H
                ;-------------
00B0 =      P3_0:   EQU     0B0H    ;Port 3
00B1 =      P3_1:   EQU     0B1H
00B2 =      P3_2:   EQU     0B2H
00B3 =      P3_3:   EQU     0B3H
00B4 =      P3_4:   EQU     0B4H
00B5 =      P3_5:   EQU     0B5H
00B6 =      P3_6:   EQU     0B6H
00B7 =      P3_7:   EQU     0B7H
                ;*********************************************************
00E0 =      A_0:    EQU     0E0H
00E1 =      A_1:    EQU     0E1H
00E2 =      A_2:    EQU     0E2H
00E3 =      A_3:    EQU     0E3H
00E4 =      A_4:    EQU     0E4H    ;Bit address of ACCUMULATOR
00E5 =      A_5:    EQU     0E5H
00E6 =      A_6:    EQU     0E6H
00E7 =      A_7:    EQU     0E7H
                ;*********************************************************
00F0 =      B_0:    EQU     0F0H
00F1 =      B_1:    EQU     0F1H
00F2 =      B_2:    EQU     0F2H
00F3 =      B_3:    EQU     0F3H
00F4 =      B_4:    EQU     0F4H
00F5 =      B_5:    EQU     0F5H
00F6 =      B_6:    EQU     0F6H
00F7 =      B_7:    EQU     0F7H
                ;*********************************************************
00A8 =      IE_0:   EQU     0A8H
00A9 =      IE_1:   EQU     0A9H
00AA =      IE_2:   EQU     0AAH
00AB =      IE_3:   EQU     0ABH    ;Bit address of IE
00AC =      IE_4:   EQU     0ACH
00AD =      IE_5:   EQU     0ADH
00AE =      IE_6:   EQU     0AEH
00AF =      IE_7:   EQU     0AFH
                ;*********************************************************
00D0 =      PSW_0:  EQU     0D0H
00D1 =      PSW_1:  EQU     0D1H
00D2 =      PSW_2:  EQU     0D2H
00D3 =      PSW_3:  EQU     0D3H
00D4 =      PSW_4:  EQU     0D4H
00D5 =      PSW_5:  EQU     0D5H
00D6 =      PSW_6:  EQU     0D6H
00D7 =      PSW_7:  EQU     0D7H
                ;*********************************************************
0088 =      TCON_0: EQU     88H
0089 =      TCON_1: EQU     89H
008A =      TCON_2: EQU     8AH
008B =      TCON_3: EQU     8BH
008C =      TCON_4: EQU     8CH
008D =      TCON_5: EQU     8DH
008E =      TCON_6: EQU     8EH
008F =      TCON_7: EQU     8FH
                ;*********************************************************
0098 =      SCON_0: EQU     98H
0099 =      SCON_1: EQU     99H
009A =      SCON_2: EQU     9AH
009B =      SCON_3: EQU     9BH
009C =      SCON_4: EQU     9CH
009D =      SCON_5: EQU     9DH
009E =      SCON_6: EQU     9EH
009F =      SCON_7: EQU     9FH
                ;*********************************************************
00B8 =      IP_0:   EQU     0B8H
00B9 =      IP_1:   EQU     0B9H
00BA =      IP_2:   EQU     0BAH
00BB =      IP_3:   EQU     0BBH
00BC =      IP_4:   EQU     0BCH
00BD =      IP_5:   EQU     0BDH
                ;*********************************************************
8000 =      P8255R0:    EQU     8000H   ; address of port
8001 =      P8255R1:    EQU     8001H
8002 =      P8255R2:    EQU     8002H
8003 =      P8255R3:    EQU     8003H
0080 =      P8255M0:    EQU     80H
0000 =      STARTRAM:   EQU     0000H
                ;*********************************************************
9000 =      KEYB:       EQU     9000H
A000 =      DISP1A:     EQU     0A000H
A001 =      DISP1B:     EQU     0A001H
A002 =      DISP1C:     EQU     0A002H
A003 =      DISP1D:     EQU     0A003H

B000 =      DISP2A:     EQU     0B000H
B001 =      DISP2B:     EQU     0B001H
B002 =      DISP2C:     EQU     0B002H
B003 =      DISP2D:     EQU     0B003H
                ;*********************
                ;* General Routine *
                ;*********************
0100                ORG     100H

0100        DELAY2:
                ;--------
                ; R2 : delay duration
                ;--------
0100 C0F0           PUSH B
0102 75F0FF  De_loop1: MOV B , #0FFH
0105 D5F0FD  De_loop2: DJNZ B ,De_loop2
0108 DAF8           DJNZ R2 ,De_loop1
010A D0F0           POP B
010C 22             RET

010D C0D0   ADD16_8: PUSH PSW
010F C3             CLR  C
0110 28             ADD  A,R0           ; (In R0,R1,A,out R0,R1)
0111 F8             MOV  R0,A           ; R0 = Low digit
0112 7400           MOV  A,#00H           ; R1 = High digit
0114 39             ADDC A,R1           ; A  = adder
0115 F9             MOV  R1,A
0116 D0D0           POP  PSW
0118 22             RET

0119 C0D0  ADD16_16: PUSH PSW ;(In R0x1,R1x10,R7x10,Ax1 - Out
R0,R1)
011B C002           PUSH R2
011D C003           PUSH R3
011F AB07           MOV  R3,R7 ;keep
0121 FA             MOV  R2,A  ;keep
0122 C3             CLR  C
0123 28             ADD  A,R0
0124 F8             MOV  R0,A
0125 EF             MOV  A,R7
0126 39             ADDC A,R1
0127 F9             MOV  R1,A
0128 AF03           MOV  R7,R3  ;get
```

สารนิพนธ์นี้เป็นที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของสารนิพนธ์นี้ทุกครั้งที่มีการนำไปใช้

```
012A EA          MOV A,R2   ;get
012B D003        POP R3
012D D002        POP R2
012F D0D0        POP PSW
0131 22          RET

0132      ADD16_8C:
          ;-------
          ; R1,R0 + R6 = R1,R0 -> C
          ;-------
0132 C3          CLR C
0133 EE          MOV A,R6
0134 28          ADD A,R0          ; (In R0,R1,R6,out R0,R1)
0135 F8          MOV R0,A          ; R0 = Low digit
0136 7400        MOV A,#00H        ; R1 = High digit
0138 39          ADDC A,R1         ; A = adder
0139 F9          MOV R1,A
013A 22          RET

013B      ADD16_16C:
          ;-----------------------
          ;(In R0x1,R1x10,R7x10,Ax1 - Out R0,R1)
          ;-----------------------
013B C002        PUSH R2
013D C003        PUSH R3
013F AB07        MOV R3,R7 ;keep
0141 FA          MOV R2,A  ;keep
0142 C3          CLR C
0143 28          ADD A,R0
0144 F8          MOV R0,A
0145 EF          MOV A,R7
0146 39          ADDC A,R1
0147 F9          MOV R1,A
0148 AF03        MOV R7,R3 ;get
014A EA          MOV A,R2  ;get
014B D003        POP R3
014D D002        POP R2
014F 22          RET

0150      ADD24_16C:
          ;-----------
          ;R2,R1,R0+R7,A=R2,R1,R0
          ;-----------
0150 C003        PUSH R3
0152 C004        PUSH R4
0154 FB          MOV R3,A          ;keep
0155 AC07        MOV R4,R7         ;keep
0157 C3          CLR C
0158 38          ADDC A,R0         ;R0+A+C=R0
0159 F8          MOV R0,A
015A EF          MOV A,R7
015B 39          ADDC A,R1
015C F9          MOV R1,A          ;R1+R7+C=R1
015D E4          CLR A
015E 3A          ADDC A,R2
015F FA          MOV R2,A          ;R2+C=R2
0160 EB          MOV A,R3          ;get
0161 AF04        MOV R7,R4         ;get
0163 D004        POP R4
0165 D003        POP R3
0167 22          RET

0168      ADD24_24C:
          ;-----------
          ;R2,R1,R0+R7,R6,R5=R2,R1,R0
          ;-----------
0168 C003        PUSH R3
016A C004        PUSH R4
016C C0F0        PUSH B
016E AB05        MOV R3,R5         ;keep
0170 AC06        MOV R4,R6         ; "
0172 8FF0        MOV B,R7          ; "
0174 C3          CLR C
0175 ED          MOV A,R5
0176 38          ADDC A,R0         ;R0+R5+C=R0
0177 F8          MOV R0,A
0178 EE          MOV A,R6
0179 39          ADDC A,R1
017A F9          MOV R1,A          ;R1+R6+C=R1
017B EF          MOV A,R7
017C 3A          ADDC A,R2
017D FA          MOV R2,A          ;R2+R7+C=R2
017E AD03        MOV R5,R3         ;get
0180 AE04        MOV R6,R4         ; "
0182 AFF0        MOV R7,B          ; "
0184 D0F0        POP B
0186 D004        POP R4
0188 D003        POP R3
018A 22          RET

018B      SUB16_8:
          ;-------
          ; subtraction
          ;-------
018B C3          CLR C     ;(In R0x1,R1x10 - A ,Out R0,R1)
018C C8          XCH A,R0
018D 98          SUBB A,R0
018E 5008        JNC LastS
0190 B90003      CJNE R1,#00,Dec
0193 D3          SETB C
0194 8002        SJMP LastS
0196 19     Dec: DEC R1
0197 C3          CLR C
0198 C8     LastS: XCH A,R0
0199 22          RET

019A      SUB16_8C:
          ;-------
          ; R1,R0 - R6 = R1,R0    R6:not distroys
          ;-------
019A C3          CLR C
019B E8          MOV A,R0
019C 9E          SUBB A,R6
019D F8          MOV R0,A
019E E9          MOV A,R1
019F 9400        SUBB A,#00H
01A1 F9          MOV R1,A
01A2 22          RET

01A3      SUB16_16C:
          ;-----------
          ; R1,R0 - R7,R6 = R1,R0      R7,R6:not distroys
          ;-----------
01A3 C3          CLR C
01A4 E8          MOV A,R0
01A5 9E          SUBB A,R6
01A6 F8          MOV R0,A
01A7 E9          MOV A,R1
01A8 9F          SUBB A,R7
01A9 F9          MOV R1,A
01AA 22          RET

01AB C0F0   MUL16_8: PUSH B   ;(In R0x1,R1x10*A - Out R0,R1)
01AD C002        PUSH R2
01AF C003        PUSH R3
01B1 C007        PUSH R7
01B3 7006        JNZ Next          ;A>0
01B5 7800        MOV R0,#00H
01B7 7900        MOV R1,#00H
01B9 8014        SJMP LastM
01BB 14     Next: DEC A
01BC B40002      CJNE A,#00H,ElseM      ;A=A-1
01BF 8008        SJMP LastM
01C1 F5F0   ElseM: MOV B,A        ;B=Counter
01C3 AA00        MOV R2,R0
01C5 AB01        MOV R3,R1
01C7 EA     LoopM: MOV A,R2
01C8 AF03        MOV R7,R3
01CA 3119        ACALL ADD16_16
01CC D5F0F8      DJNZ B,LoopM
01CF D007   LastM: POP R7
01D1 D003        POP R3
01D3 D002        POP R2
01D5 D0F0        POP B
01D7 22          RET

0002 =     MULCY:  EQU  02H       ;MULTIPLY Carry Frag

01D8       MULT8C:
           ;-----------
           ;  A=Multiplier
           ;  R6=Multiplicand    ;R7 = reserver
           ;  R1,R0=Product
           ;-----------
01D8 C002        PUSH R2
01DA C003        PUSH R3
01DC C004        PUSH R4
01DE C007        PUSH R7
01E0 FB          MOV R3,A
01E1 AC06        MOV R4,R6
01E3 FA          MOV R2,A
01E4 7800        MOV R0,#00H
01E6 7900        MOV R1,#00H
01E8 7F00        MOV R7,#00H
01EA C202        CLR MULCY
01EC EA     MUL8C2: MOV A,R2
01ED C3          CLR C
01EE 13          RRC A
01EF FA          MOV R2,A
01F0 5007        JNC MUL8C
01F2 EE          MOV A,R6          ;R1,R0+R7,A=R1,R0
01F3 313B        ACALL ADD16_16C   ;result in R1,R0
01F5 5002        JNC MUL8C
01F7 D202        SETB MULCY
01F9 EE     MUL8C: MOV A,R6   ;Shift Left of R7
01FA C3          CLR C
01FB 33          RLC A
01FC FE          MOV R6,A
01FD EF          MOV A,R7   ;Shift Left of R6
01FE 33          RLC A
01FF FF          MOV R7,A
0200 BA00E9      CJNE R2,#00H,MUL8C2
0203 A202        MOV C,MULCY
0205 AE04        MOV R6,R4
0207 EB          MOV A,R3
0208 D007        POP R7
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
020A D004          POP  R4
020C D003          POP  R3
020E D002          POP  R2
0210 22            RET

0211       MULT16C:
        ;--------------
        ;  A=Multiplier
        ;  R6,R5=Multiplicand    R7-reserve
        ;  R2,R1,R0=Product
        ;--------------
0211 C003          PUSH R3    ;reserve Mul
0213 C004          PUSH R4
0215 C007          PUSH R7
0217 C0F0          PUSH B
0219 AC05          MOV  R4,R5    ;keep
021B 8EF0          MOV  B,R6     ;keep
021D 7800          MOV  R0,#00H
021F 7900          MOV  R1,#00H
0221 7A00          MOV  R2,#00H
0223 7F00          MOV  R7,#00H
0225 C202          CLR  MULCY
0227 FB            MOV  R3,A
0228 C3      MUL16C2:  CLR  C
0229 EB            MOV  A,R3
022A 13            RRC  A
022B FB            MOV  R3,A
022C 500E          JNC  MUL16C
022E 3168          ACALL  ADD24_24C
;R2,R1,R0+R7,R6,R5=R2,R1,R0
0230 500A          JNC   MUL16C
0232 D202          SETB  MULCY
0234 7AFF          MOV R2,#0FFH
0236 79FF          MOV R1,#0FFH
0238 78FF          MOV R0,#0FFH
023A 8011          SJMP  endMul
023C C3     MUL16C:   CLR  C
023D ED            MOV  A,R5    ;Shift Left of R5
023E 33            RLC  A
023F FD            MOV  R5,A
0240 EE            MOV  A,R6    ;Shift Left of R6
0241 33            RLC  A
0242 FE            MOV  R6,A
0243 EF            MOV  A,R7    ;Shift  Left of R7
0244 33            RLC  A
0245 FF            MOV  R7,A
0246 BB00DF          CJNE  R3,#00H,MUL16C2
0249 AD04          MOV  R5,R4    ;get
024B AEF0          MOV  R6,B     ;get
024D A202    endMul:   MOV  C,MULCY
024F D0F0          POP  B
0251 D007          POP  R7
0253 D004          POP  R4
0255 D003          POP  R3
0257 22            RET

0258       MULT16C2:
        ;--------------
        ;  A=Multiplier
        ;  R6,R5=Multiplicand    R7-reserve
        ;  R3,R2,R1,R0=Product
        ;--------------
0258 C004          PUSH R4
025A C007          PUSH R7
025C C0F0          PUSH B
025E AC05          MOV  R4,R5    ;keep
0260 8EF0          MOV  B,R6     ;keep
0262 7800          MOV  R0,#00H
0264 7900          MOV  R1,#00H
0266 7A00          MOV  R2,#00H
0268 7F00          MOV  R7,#00H
026A D2D3          SETB PSW_3
026C 7800          MOV  R0,#00H    ;use R0 in bank1
026E C2D3          CLR  PSW_3
0270 C202          CLR  MULCY
0272 FB            MOV  R3,A
0273 C3    MUL16C3:  CLR  C
0274 EB            MOV  A,R3
0275 13            RRC  A
0276 FB            MOV  R3,A
0277 5017          JNC  MUL16_3C
0279 3168          ACALL  ADD24_24C
;R2,R1,R0+R7,R6,R5=R2,R1,R0
        ;-
027B D2D3            SETB PSW_3
027D E9            MOV A,R1          ;to bank1
027E 38            ADDC A,R0
027F F9            MOV R1,A
0280 C2D3            CLR  PSW_3      ;to bank0
        ;-
0282 500C          JNC   MUL16_3C
0284 D202          SETB  MULCY
0286 7BFF          MOV R3,#0FFH
0288 7AFF          MOV R2,#0FFH
028A 79FF          MOV R1,#0FFH
028C 78FF          MOV R0,#0FFH
028E 8017          SJMP  endMul1
0290 C3     MUL16_3C: CLR  C
0291 ED            MOV  A,R5    ;Shift Left of R5
0292 33            RLC  A
0293 FD            MOV  R5,A
0294 EE            MOV  A,R6    ;Shift Left of R6
0295 33            RLC  A
0296 FE            MOV  R6,A
0297 EF            MOV  A,R7    ;Shift  Left of R7
0298 33            RLC  A
0299 FF            MOV  R7,A
029A BB00D6          CJNE  R3,#00H,MUL16C3
029D AD04          MOV  R5,R4    ;get
029F AEF0          MOV  R6,B     ;get
        ;-
02A1 D2D3          SETB  PSW_3    ;to bank1
02A3 E9            MOV  A,R1
02A4 C2D3          CLR  PSW_3     ;to bank0
        ;-
02A6 FB            MOV  R3,A
02A7 A202    endMul1:  MOV  C,MULCY
02A9 D0F0          POP  B
02AB D007          POP  R7
02AD D004          POP  R4
02AF 22            RET


02B0       MULT16_16C:
        ;--------------
        ;  A,R4=Multiplier   R4*10,A*1
        ;  R6,R5=Multiplicand    R7-reserve    distroys
        ;  R2,R1,R0=Product
        ;--------------
02B0 C003          PUSH R3    ;reserve Mul
02B2 C007          PUSH R7
02B4 7800          MOV  R0,#00H
02B6 7900          MOV  R1,#00H
02B8 7A00          MOV  R2,#00H
02BA 7F00          MOV  R7,#00H
02BC C202          CLR  MULCY    ;clear multiply flag
02BE FD            MOV  R3,A
02BF C3     MUL2C2:  CLR  C
02C0 EC            MOV  A,R4
02C1 13            RRC  A
02C2 FC            MOV  R4,A     ;R4->A->C
02C3 EB            MOV  A,R3
02C4 13            RRC  A
02C5 FB            MOV  R3,A
02C6 500E          JNC  MUL2C
02C8 3168          ACALL  ADD24_24C  ;R2,R1,R0+R7,R6,R5=R2,R1,R0
02CA 500A          JNC  MUL2C
02CC D202          SETB  MULCY
02CE 78FF          MOV R0,#0FFH
02D0 79FF          MOV R1,#0FFH
02D2 7AFF          MOV R2,#0FFH
02D4 8010          SJMP  endMul2
02D6 C3    MUL2C:  CLR  C
02D7 ED            MOV  A,R5    ;Shift Left of R5
02D8 33            RLC  A
02D9 FD            MOV  R5,A
02DA EE            MOV  A,R6    ;Shift Left of R6
02DB 33            RLC  A
02DC FE            MOV  R6,A
02DD EF            MOV  A,R7    ;Shift  Left of R7
02DE 33            RLC  A
02DF FF            MOV  R7,A
02E0 BC00DC          CJNE  R4,#00H,MUL2C2
02E3 BB00D9          CJNE  R3,#00H,MUL2C2
02E6 A202  endMul2:  MOV  C,MULCY
02E8 D007          POP  R7
02EA D003          POP  R3
02EC 22            RET

02ED       MULT16_16C2:
        ;--------------
        ;  A,R4=Multiplier   R4*10,A*1
        ;  R6,R5=Multiplicand    R7-reserve    distroys
        ;  R3,R2,R1,R0=Product
        ;--------------
02ED C007          PUSH R7
02EF 7800          MOV  R0,#00H
02F1 7900          MOV  R1,#00H
02F3 7A00          MOV  R2,#00H
02F5 7F00          MOV  R7,#00H
02F7 D2D3          SETB PSW_3
02F9 7800          MOV  R0,#00H    ;use R0 in bank1
02FB 7900          MOV  R1,#00H    ;use R1 in bank1
02FD C2D3          CLR  PSW_3
02FF C202          CLR  MULCY    ;clear multiply flag
0301 FB            MOV  R3,A
0302 C3     MUL3C2:  CLR  C
0303 EC            MOV  A,R4
0304 13            RRC  A
0305 FC            MOV  R4,A     ;R4->A->C
0306 EB            MOV  A,R3
0307 13            RRC  A
0308 FB            MOV  R3,A
0309 5017          JNC  MUL3C
030B 3168          ACALL  ADD24_24C    ;R2,R1,R0 + (R0),R7,R6,R5 =
R2,R1,R0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ;—      JNC MUL3C
                ;-
030D D2D3       SETB PSW_3
030F E9         MOV A,R1        ;to bank1
0310 38         ADDC A,R0
0311 F9         MOV R1,A
0312 C2D3       CLR PSW_3       ;to bank0
                ;-
0314 500C       JNC MUL3C
0316 D202       SETB MULCY      ;set multiply bit
0318 78FF       MOV R0,#0FFH
031A 79FF       MOV R1,#0FFH
031C 7AFF       MOV R2,#0FFH
031E 7BFF       MOV R3,#0FFH
0320 801D       SJMP endMul3
0322 C3     MUL3C:  CLR C
0323 ED         MOV A,R5    ;Shift Left of R5
0324 33         RLC A
0325 FD         MOV R5,A
0326 EE         MOV A,R6    ;Shift Left of R6
0327 33         RLC A
0328 FE         MOV R6,A
0329 EF         MOV A,R7    ;Shift Left of R7
032A 33         RLC A
032B FF         MOV R7,A
                ;-
032C D2D3       SETB PSW_3  ;bank1
032E E8         MOV A,R0
032F 33         RLC A
0330 F8         MOV R0,A
0331 C2D3       CLR PSW_3   ;bank0
                ;-
0333 BC00CC     CJNE R4,#00H,MUL3C2
0336 BB00C9     CJNE R3,#00H,MUL3C2
                ;-
0339 D2D3       SETB PSW_3 ;to bank1
033B E9         MOV 0155 AC07       MOV R4,R7
033C C2D3       CLR PSW_3 ;to bank0
                ;-
033E FB         MOV R3,A    ;R0 bk1 -> R3
033F A202   endMul3:  MOV C,MULCY
0341 D007       POP R7
0343 22         RET

0344    DIV16:
        ;————————
        ; DIVISOR = R7
        ; DIVIDEND = R1.R0 ,QUTTIENT
        ;————————
0344 C002       PUSH R2
0346 C003       PUSH R3
0348 C006       PUSH R6
034A 7A00       MOV R2,#00H
034C 7B00       MOV R3,#00H  ;0 -> QUOTN
034E 75F010     MOV B,#16D   ;16 bit
0351 E4         CLR A
0352 FE         MOV R6,A    ;buff of ACC
                ;-
0353 C3     review:  CLR C
0354 E8         MOV A,R0
0355 33         RLC A
0356 F8         MOV R0,A
                ;-
0357 E9         MOV A,R1
0358 33         RLC A
0359 F9         MOV R1,A    ;Shift MSB of DIVND to LSB ACC
                ;
035A EE         MOV A,R6
035B 33         RLC A
035C FE         MOV R6,A
                ;-
035D C3         CLR C
035E 9F         SUBB A,R7
035F 4004       JC shift0
0361 FE         MOV R6,A
0362 D3         SETB C
0363 8001       SJMP notcry
0365 C3     shift0:  CLR C
0366 EA     notcry:  MOV A,R2
0367 33         RLC A
0368 FA         MOV R2,A
0369 EB         MOV A,R3
036A 33         RLC A
036B FB         MOV R3,A  ;Shift 0 into LSB of QUOTN
                ;-
036C D5F0E4     DJNZ B,review
036F A802       MOV R0,R2
0371 A903       MOV R1,R3
0373 D006       POP R6
0375 D003       POP R3
0377 D002       POP R2
0379 22         RET


037A    DIV24:
        ;————————
        ; DIVISOR = R7
        ; DIVIDEND = R2,R1.R0 ,QUTTIENT R2,R1,R0
        ;————————
037A C003       PUSH R3
037C C004       PUSH R4
037E C005       PUSH R5
0380 C0F0       PUSH B
                ;-
0382 AD02       MOV R5,R2
0384 7A00       MOV R2,#00H
0386 7B00       MOV R3,#00H
0388 7C00       MOV R4,#00H  ;0 -> QUOTN
038A 75F018     MOV B,#24D   ;24 bit
038D D2D3       SETB PSW_3   ;to bank1
038F 7800       MOV R0,#00H  ;buff of ACC
0391 7900       MOV R1,#00H
0393 C2D3       CLR PSW_3    ;to bank0
                ;-
0395 C3     review1:  CLR C
0396 E8         MOV A,R0
0397 33         RLC A
0398 F8         MOV R0,A
                ;-
0399 E9         MOV A,R1
039A 33         RLC A
039B F9         MOV R1,A   ;Shift MSB of DIVND to LSB ACC
                ;-
039C ED         MOV A,R5
039D 33         RLC A
039E FD         MOV R5,A
                ;-
039F D2D3       SETB PSW_3   ; to bank1
03A1 E8         MOV A,R0     ;ACC
03A2 33         RLC A
03A3 F8         MOV R0,A
                ;-
03A4 E9         MOV A,R1     ;ACC
03A5 33         RLC A
03A6 F9         MOV R1,A
                ;-
03A7 E8         MOV A,R0     ;keep for moment
03A8 FA         MOV R2,A
03A9 E9         MOV A,R1
03AA FB         MOV R3,A
                ;-
03AB C2D3       CLR PSW_3    ; to bank0
03AD EF         MOV A,R7
                ;-
03AE D2D3       SETB PSW_3   ; to bank1
03B0 FF         MOV R7,A     ;R7 bank1 = R7 bank0 for DIVISOR
03B1 C3         CLR C
03B2 E8         MOV A,R0
03B3 9F         SUBB A,R7    ;ACC < DIVISER if yes shift 0 to LSB of
QUTN
03B4 F8         MOV R0,A
03B5 E9         MOV A,R1
03B6 9400       SUBB A,#00H
03B8 F9         MOV R1,A
03B9 4003       JC shift0_1
03BB D3         SETB C
03BC 8005       SJMP notcry1
03BE C3     shift0_1: CLR C
03BF EB         MOV A,R3     ;return value
03C0 F9         MOV R1,A
03C1 EA         MOV A,R2
03C2 F8         MOV R0,A
                ;-
03C3 C2D3   notcry1: CLR PSW_3   ;to bank0
03C5 EA         MOV A,R2
03C6 33         RLC A
03C7 FA         MOV R2,A
                ;-
03C8 EB         MOV A,R3
03C9 33         RLC A
03CA FB         MOV R3,A
                ;-
03CB EC         MOV A,R4
03CC 33         RLC A
03CD FC         MOV R4,A     ;Shift 0 into LSB of QUOTN
                ;-
03CE D5F0C4     DJNZ B,review1   ;count-1 = count
03D1 A802       MOV R0,R2
03D3 A903       MOV R1,R3
03D5 AA04       MOV R2,R4
                ;-
03D7 D0F0       POP B
03D9 D005       POP R5
03DB D004       POP R4
03DD D003       POP R3
03DF 22         RET

        ;————
        ; end routine
        ;————

03E0        INCL "inramz.asm"
        ;————————
        ; variable in internalram
        ;————————
```

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0030 =      KPCount:   EQU  30H      ;30H
0031 =      KDCount:   EQU  31H      ;31H
0032 =      KICount:   EQU  32H      ;32H
0033 =      MDCount:   EQU  33H      ;33H
0034 =      MDWork:    EQU  34H      ;34H
0035 =      ADBuff:    EQU  35H      ;35H
0036 =      DABuff:    EQU  36H      ;36H
0037 =      KeyBuff:   EQU  37H      ;37H
0038 =      TogBuff:   EQU  38H      ;38H
0039 =      SetPoint:  EQU  39H      ;39H
003A =      Status:    EQU  3AH      ;3AH   ;status keybroad previus
pressed
003B =      PLLCount0: EQU  3BH      ;3BH
003B =      ADDRESS:   EQU  3BH      ;3CH   ;station address
003D =      BAUDRATE:  EQU  3DH      ;3DH   ;BAUDRATE
003E =      DCPID:     EQU  3EH      ;3EH   ; output of PID
003F =      DSP:       EQU  3FH      ;3FH   ;use  setpoint
0040 =      DCP:       EQU  40H      ;40H
0041 =      DC0:       EQU  41H      ;41H
0042 =      KP0:       EQU  42H      ;42H
0043 =      KP1:       EQU  43H      ;43H
0044 =      DCI:       EQU  44H      ;44H
0045 =      KI0:       EQU  45H      ;45H
0046 =      KI1:       EQU  46H      ;46H
0047 =      DT:        EQU  47H      ;47H    run sampling
0048 =      Sum0:      EQU  48H      ;48H
0049 =      Sum1:      EQU  49H      ;49H
004A =      DCD:       EQU  4AH      ;*4AH    decare repeat
004B =      DV0:       EQU  4BH      ;4BH    Not use in
USEPLLZ1.ASM
004C =      DE:        EQU  4CH      ;4CH
004D =      KD0:       EQU  4DH      ;4DH
004E =      KD1:       EQU  4EH      ;4EH
004F =      DCPI:      EQU  4FH      ;4FH

002F =      ADCOUNT:   EQU  2FH      ;2FH    address count
002E =      BDCOUNT:   EQU  2EH      ;2EH    baud  count
002D =      PIDdelay:  EQU  2DH      ;2DH    delay for transient
of PID
                                     ;2CH
                                     ;2BH
                                     ;2AH
                                     ;29H
                                     ;28H
                                     ;27H
                                     ;26H
                                     ;25H
            ;-----------
            ; Internal Frag
            ;-----------

            ;Intram 20H
0000 =      SGNFG:     EQU  00H      ;00H
0001 =      SGNDFG:    EQU  01H      ;01H
0002 =      MultC:     EQU  02H      ;02H
0003 =      SGNIFG:    EQU  03H      ;03H
0004 =      CURBLINKFG:  EQU  04H    ;04H
0005 =      ToggleFg:  EQU  05H      ;05H
0006 =      EditFg:    EQU  06H      ;06H
0007 =      Num1Fg:    EQU  07H      ;07H

            ;IntRam 21H
0008 =      PVFg:      EQU  08H      ;08H
0009 =      TestTF1Fg: EQU  09H      ;09H
000A =      OldSGNFg:  EQU  0AH      ;0AH  ;keep previously of
SGNFg
                                     ;0BH
000C =      HighFg:    EQU  0CH      ;0CH
000D =      DutyFg:    EQU  0DH      ;0DH
000E =      Intr2ACtFg:  EQU  0EH    ;0EH
000F =      PVToggleFg:  EQU  0FH    ;0FH

            ;IntRAM 22H
0010 =      FirstStartFg:    EQU  10H
0011 =      SendStartFg:     EQU  11H
0012 =      CompleteSMFg:  EQU  12H
0013 =      AddressSMFg:   EQU  13H
0014 =      PVSMFg:        EQU  14H
0015 =      ToHostHeadFg:  EQU  15H
0016 =      InlockFg:      EQU  16H      ;is in Phase-fre lock
0017 =      BufferFg:      EQU  17H      ;general buffer flag

            ;IntRam23H
0018 =      HeadOKfg:  EQU  18H      ;check header of frame of Serial
0019 =      BaudSetFg: EQU  19H      ;for set baudrate
001A =      SPFg:      EQU  1AH      ;for set Setpoint with clear
SUM0,SUM1,DC0
001B =      SUMFg:     EQU  1BH      ;for Over of SUM
001C =      BKSUMFg:   EQU  1CH      ;for do not SUM in PID_control
            ;-----------
            ; Constant
            ;-----------
0040 =      BlinkP:    EQU  040H
0030 =      BlinkPI:   EQU  030H
0020 =      BlinkPID:  EQU  020H
00FF =      BlinkPLL:  EQU  0FFH
0003 =      MDMAXCon:  EQU  03H      ;Maximum run mode
006C =      KMAXCon:   EQU  108D     ;Maximum of variable K...
0008 =      N0_Digit:  EQU  08H      ;Number of digit of K..
000A =      N0_DTDig:  EQU  0AH      ;Number of digit of Sampling time
            ;-----------
            ; boaud rate constant
            ;-----------
00FD =      BR9600:    EQU  253d
00FA =      BR4800:    EQU  250d
00F4 =      BR2400:    EQU  244d
00E8 =      BR1200:    EQU  232d
00D0 =      BR600:     EQU  208d
00A0 =      BR300:     EQU  160d
0040 =      BR150:     EQU  064d
            ;-
03E0        initAllVar:                                          *
            ;-----------
            ; initial all variable
            ;-----------
03E0 75301C       MOV KPCount,#28D      ;30H
03E3 753100       MOV KDCount,#00H      ;31H
03E6 753200       MOV KICount,#00H      ;32H
03E9 753300       MOV MDCount,#00H      ;33H
03EC 753400       MOV MDWork,#00H       ;34H
03EF 753500       MOV ADBuff,#00H       ;35H
03F2 753600       MOV DABuff,#00H       ;36H
03F5 753700       MOV KeyBuff,#00H      ;37H
03F8 753840       MOV TogBuff,#BlinkP   ;38H   Timer for cursor
blink
03FB 753980       MOV SetPoint,#80H     ;39H
03FE 753A00       MOV Status,#00H       ;3AH
0401 753B00       MOV PLLCount0,#00H    ;3BH
0404 753B00       MOV ADDRESS,#00H      ;3DH
0407 753D40       MOV BAUDRATE,#BR150   ;3DH
040A 753E00       MOV DCPID,#00H        ;3EH
040D 753F00       MOV DSP,#00H          ;3FH
0410 754000       MOV DCP,#00H          ;40H
0413 754100       MOV DC0,#00H          ;41H
0416 754200       MOV KP0,#00H          ;42H
0419 754300       MOV KP1,#00H          ;43H
041C 754400       MOV DCI,#00H          ;44H
041F 754500       MOV KI0,#00H          ;45H
0422 754600       MOV KI1,#00H          ;46H
0425 754705       MOV DT,#05H           ;47H   sampling time
constant
0428 754800       MOV Sum0,#00H         ;48H
042B 754900       MOV Sum1,#00H         ;49H
042E 754A00       MOV DCD,#00H          ;4AH
0431 754B00       MOV DV0,#00H          ;4BH
0434 754C00       MOV DE,#00H           ;4CH
0437 754D00       MOV KD0,#00H          ;4DH
043A 754E00       MOV KD1,#00H          ;4EH
043D 754F00       MOV DCPI,#00H         ;4FH
0440 752F00       MOV ADCount,#00H      ;2FH
0443 752E00       MOV BDCount,#00H      ;2EH
0446 752D00       MOV PIDdelay,#00H     ;2DH
                                        ;2CH
                                        ;2BH
                                        ;2AH
                                        ;29H
                                        ;28H
                                        ;27H
                                        ;26H

0449 C200         CLR SGNFG             ;00H
044B C201         CLR SGNDFG            ;01H
044D C202         CLR MultC             ;02H
044F C203         CLR SGNIFG            ;03H
0451 C204         CLR CURBLINKFG        ;04H
0453 C205         CLR ToggleFg          ;05H
0455 C206         CLR EditFg            ;06H
0457 C207         CLR Num1Fg            ;07H
0459 C208         CLR PVFg              ;08H
                                        ;0AH
                                        ;0BH
045B C20C         CLR HighFg            ;0CH
                                        ;0DH
                                        ;0EH
045D C20F         CLR PVtoggleFg        ;0FH
045F C210         CLR FirstStartFg      ;10H
0461 C211         CLR SendStartFg       ;11H
0463 C212         CLR CompleteSMFg      ;12H
0465 C213         CLR AddressSMFg       ;13H
0467 C214         CLR PVSMFg            ;14H
0469 C215         CLR ToHostHeadFg      ;15H
046B C216         CLR InlockFg          ;16H
046D C217         CLR BufferFg          ;17H
046F C218         CLR HeadOKFg          ;18H
0471 C219         CLR BaudSetFg         ;19H
0473 C21A         CLR SPFg              ;1AH
0475 C21B         CLR SUMFg             ;1BH
0477 C21C         CLR BKSUMFg           ;1CH

0479 22           RET
            ;-----------
            ; end routine
            ;-----------
```

เอกสารนี้จัดทำขึ้นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
047A              INCL  "allintz1.asm"          ;3

          ;  ADBUFF:    EQU   35H
          ;  DABUFF:    EQU   36H
          ;  KEYBUFF:   EQU   37H
          ;  TOGBUFF:   EQU   38H
          ;  SETPOINT:  EQU   39H     ;preload setpoint and real to DSP
          ;  STATUS:    EQU   3AH
          ;  T0BUFF:    EQU   3BH     ;timer buffer
          ;  T0COUNT:   EQU   3CH     ;timer count
          ;  ToggleFg:  EQU   05H

          ;  CURBLINKFg: EQU   04H

          ;  KEYB:      EQU   09000H

C000 =        AtoDadd:   EQU   0C000H
D000 =        DtoAadd0:  EQU   0D000H
D001 =        DtoAadd1:  EQU   0D001H
D002 =        DtoAadd2:  EQU   0D002H
D003 =        DtoAadd3:  EQU   0D003H
          ;-----------------
          ; boaud rate constant
          ;-----------------
          ;  BR9600:    EQU   253d
          ;  BR4800:    EQU   250d
          ;  BR2400:    EQU   244d
          ;  BR1200:    EQU   232d
          ;  BR600:     EQU   208d
          ;  BR300:     EQU   160d
          ;  BR150:     EQU   064d
          ;-----------------
          ;  initial of main routine
          ;-----------------

047A C2D1     InitMn:  CLR  PSW_1          ;reserve for fortune used
047C F17B              ACALL InitDisp
047E B1CF              ACALL InitDA
0480 9187              ACALL InitIntr
0482 91E6              ACALL InitT0
0484 B15C              ACALL InitSM
0486 22                RET

          ;**********
          ;  interupt
          ;**********
          ;---------
          ;  initial all of interupt
          ;---------
0487 D2AF     InitIntr: SETB IE_7          ;Enable All
0489 D288               SETB TCON_0        ;Edge intr0 KEY
048B D28A               SETB TCON_2        ;Edge intr1
048D 90C000             MOV DPTR ,#AtoDadd
0490 E0                 MOVX A,@DPTR       ;initial read for clear intr sig  ex0
0491 D2A8               SETB IE_0          ;enable EX0   keybroad
0493 D2AA               SETB IE_2          ;enable EX1 and use again in
"useAllpi"
0495 22                 RET

          ;Interupt IE1 is in Inkey2.asm files

0496        WriteAD:
          ;---------
          ;  Write to AtoD
          ;---------
0496 C000               PUSH  R0
0498 C001               PUSH  R1
049A C002               PUSH  R2
049C C006               PUSH  R6
049E C007               PUSH  R7
04A0 C3                 CLR  C
04A1 7800               MOV  R0,#00H
04A3 7900               MOV  R1,#00H
04A5 7F0A               MOV  R7,#0AH          ; read A/D 16 times then
Average
04A7 90D002   Loopbf:   MOV  DPTR,#DtoAadd2   ;wait of was read of
AD
04AA E0                 MOVX A,@DPTR          ; at Pin PC0 of 8255
indentical
04AB 5401               ANL  A,#00000001B     ;DA.
04AD 60F8               JZ   Loopbf
          ;
04AF 90C000             MOV DPTR ,#AtoDadd    ;act WR\ to A/D
04B2 7400               MOV A,#00H
04B4 F0                 MOVX @DPTR,A
          ;
04B5 90D002   LoopWAD:  MOV  DPTR,#DtoAadd2   ;wait of time
conversion of AD
04B8 E0                 MOVX A,@DPTR          ; at Pin PC0 of 8255
indentical
04B9 5401               ANL  A,#00000001B     ;DA.
04BB 70F8               JNZ  LoopWAD
          ;
04BD 7A01               MOV  R2,#01H
04BF 120100             LCALL  DELAY2         ;delay conversion for 1mS
          ;
04C2 90C000             MOV  DPTR,#AtoDadd    ;Read A/D
04C5 E0                 MOVX A,@DPTR

04C6 FE                 MOV  R6,A
          ;-
04C7 120132             LCALL ADD16_8C        ;R1R0+R6->R1R0
04CA DFDB               DJNZ  R7,loopbf
          ;-
04CC 7F0A               MOV R7,#0AH
04CE 120344             LCALL DIV16           ;R1R0/R7->R1R0
04D1 8835               MOV ADBuff,R0

04D3 00       OutAD:    NOP
04D4 D007               POP R7
04D6 D006               POP R6
04D8 D002               POP R2
04DA D001               POP R1
04DC D000               POP R0
04DE 22                 RET

          ;*****************
          ;  interupt IE1 routine
          ;*********************
04DF C2AA     IntrIE1:  CLR  IE_2
04E1 D20E               SETB Intr2ActFg       ;use PLL_V5
04E3 D2AA               SETB IE_2
04E5 22                 RET

          ;*****************
          ;  Timer0 routine
          ;*****************
04E6        InitT0:
          ;---------
          ;  initial timer0
          ;---------
04E6 758902             MOV  TMOD, #00000010B       ;set timer mode 2
                  ;  MOV  TL0 ,#00H               ;not used in timer mode2
04E9 758C9C             MOV  TH0 ,#9CH
04EC D28C               SETB  TCON_4          ;auto reset time
04EE C2A9               CLR  IE_1  ;disable intr timer Used again in Set_PLL
"useallpi.asm"
04F0 22                 RET

04F1        IntrTF0:
          ;---------
          ;  use timer0 intr
          ;  ? don't use in PLL_V3
          ;---------
04F1 C2A9               CLR  IE_1

04F3 90A000             MOV  DPTR,#DISP1A   ;+
04F6 7403               MOV  A,#00000011B   ;+  check Timer work
04F8 F0                 MOVX  @DPTR, A      ;+

          ;--- PLL counters and used in PLLmode ---
04F9 C3                 CLR  C
04FA E53B               MOV  A,PllCount0
04FC 2401               ADD  A,#01H
04FE F53B               MOV  PLLCount0,A
0500 5003               JNC  PllOutOut    ; No More Than  FFH
0502 753BFF             MOV  PLLCount0,#0FFH
0505 00       PLLOutOut:  NOP
0506 D2A9     OutIntrTF0: SETB  IE_1
0508 22                 RET

0509        TogglePV:
          ;---------------
          ;  Toggle display cursor blink
          ;---------------
0509 C000               PUSH  R0
050B C001               PUSH  R1
050D 300446             JNB   CurBlinkFg,togout
0510 D53843             DJNZ  togbuff,togout
0513 B205               CPL   toggleFg
0515 D20F               SETB  PVtoggleFg
0517 E533               MOV   A,MDCount
0519 B40005             CJNE  A,#00h,TogPI    ;load toggle of P
051C 753840             MOV   TogBuff,#BlinkP
051F 8016               SJMP  Light_Dark
0521 B40105   TogPI:    CJNE  A,#01h ,TogPID
0524 753830             MOV   TogBuff,#BlinkPI
0527 800E               SJMP  Light_Dark
0529 B40205   TogPID:   CJNE  A,#02h,TogPLL
052C 753820             MOV   TogBuff,#BlinkPID
052F 8006               SJMP  Light_Dark
0531 B40303   TogPLL:   CJNE  A,#03h,Light_Dark
0534 7538FF             MOV   TogBuff,#BlinkPLL
0537 300508   Light_Dark: JNB  ToggleFg,LightBlink
053A 7400               MOV   A,#00000000B  ;" "
053C 90A002             MOV   DPTR,#DISP1C
053F F0                 MOVX  @DPTR,A
0540 8006               SJMP  outtog
0542 7440     LightBlink: MOV   A,#01000000B  ;"-"
0544 90A002             MOV   DPTR,#DISP1C
0547 F0                 MOVX  @DPTR,A
0548 00       outtog:   NOP
0549 30080A             JNB   PVfg,togout
054C A835               MOV   R0,ADBUFF
054E 7900               MOV   R1,#00H
0550 12071A             LCALL HexToPer
0553 12066D             LCALL Showdisp3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0556 00     togout:   NOP
0557 D001             POP R1
0559 D000             POP R0
055B 22               RET

        ;*********************
        ; Serial port Control
        ;*********************
055C      InitSM:
        ;-----------
        ; Initial serial intr
        ;-----------
055C C2AC             CLR IE_4          ;disable serial interupt
055E C219             CLR BaudSetFg     ;flag for Keyboad set BAUD
0560 759850           MOV SCON ,#50h    ;set serial mode1 enable for
recives
0563 D29D             SETB SCON_5       ;MODE1 8-bit UART baud
variable
0565 53877F           ANL PCON,#01111111B  ;use timer 1 baudrate
0568 858D8D           MOV TH1,BAUDRATE     ;Set boudrate
056B 758920           MOV TMOD,#20h     ; Set timer1 auto reload
mode
056E D28E             SETB TCON_6       ; Generate baudrate clok
0570 D2AC             SETB IE_4         ; enable serial interupt
0572 22               RET

0573      IntrSM:
        ;-----------
        ; interupt serial Port
        ;-----------
0573 C000             PUSH R0
0575 C001             PUSH R1
0577 C002             PUSH R2
0579 C2AC             CLR IE_4
057B 209802           JB SCON_0,RCVE    ;recive interupt frag
057E 8040             SJMP GOout
        ;-
0580 201809   RCVE:   JB HeadOKfg,ChkADD
0583 E599             MOV A,SBUF        ;chk header of frame
0585 B4AA38           CJNE A,#10101010b,Goout
0588 D218             SETB HeadOKfg
058A 8034             SJMP Goout
        ;-
058C E599     ChkADD: MOV A,SBUF
058E B53B2D           CJNE A,ADDRESS,BeGoout ;ADDRESS begin
0..11 show 1..12
        ;-
0591 A835             MOV R0,ADBuff
0593 C298             CLR SCON_0        ;clear recv
0595 C299             CLR SCON_1        ;clear send
0597 7A10             MOV R2,#010H
0599 DAFE     SrLoop1: DJNZ R2,SrLoop1  ;delay
059B 8899             MOV SBUF,R0       ;send PV out
059D 209902   Send0:  JB SCON_1,CVout   ;wait empty buffer
05A0 80FB             SJMP Send0
        ;-
05A2 C299     CVout:  CLR SCON_1
05A4 7A10             MOV R2,#010H
05A6 DAFE     SrLoop2: DJNZ R2,SrLoop2
05A8 853699           MOV SBUF,DABUFF   ;send CV out
05AB 209902   Send1:  JB SCON_1,SPout   ;wait empty buffer
05AE 80FB             SJMP Send1
        ;-
05B0 C299     SPout:  CLR SCON_1
05B2 7A10             MOV R2,#010H
05B4 DAFE     SrLoop3: DJNZ R2,SrLoop3
05B6 853F99           MOV SBUF,DSP      ;send SP out
05B9 209902   Send2:  JB SCON_1,BeGoOut ;wait empty buffer
05BC 80FB             SJMP Send2
        ;-
05BE C218     BeGoOut: CLR HeadOKfg
05C0 C298     GoOut:  CLR SCON_0
05C2 C299             CLR SCON_1
05C4 C29A             CLR SCON_2
05C6 D2AC             SETB IE_4
05C8 D002             POP R2
05CA D001             POP R1
05CC D000             POP R0
05CE 22               RET

        ;**************
        ; DA routine
        ;**************

05CF      InitDA:
        ;-----------
        ; Initial DtoA  & has recive intrupt signal form AD to PC0
        ;-----------
05CF 90D003           MOV DPTR , #DtoAadd3
05D2 7481             MOV A ,#81H    ;PA-out,PB-out;PCl-in;PCh-out
05D4 F0               MOVX @DPTR ,A
05D5 22               RET

05D6      WriteDA:
        ;-----------
        ; Write Data to DtoA
        ;-----------
05D6 90D001           MOV DPTR ,#DtoAadd1
05D9 E536             MOV A,DABUFF
05DB F0               MOVX @DPTR,A
05DC 22               RET
        ;-----------
        ; End
        ;-----------

05DD              INCL "7-seg.asm"
        ;*********************
        ; 7-Segment routine
        ;*********************

        ; DISP1A:   EQU   0A000H
        ; DISP1B:   EQU   0A001H
        ; DISP1C:   EQU   0A002H
        ; DISP1D:   EQU   0A003H


        ; DISP2A:   EQU   0B000H
        ; DISP2B:   EQU   0B001H
        ; DISP2C:   EQU   0B002H
        ; DISP2D:   EQU   0B003H

05DD      ShowDinP1:
        ;-----------
        ; Back Disp Show
        ; R0->Digit0
        ; R1->Digit1
        ;-----------
05DD C002             PUSH R2
05DF E8               MOV A,R0
05E0 900C40           MOV DPTR ,#TABLE1  ;I-DIGIT0,DIGIT1
05E3 540F             ANL A,#00001111B  ; Only 4-Low,Bit
05E5 93               MOVC A,@A+DPTR    ; Look TABLE1
05E6 90A002           MOV DPTR ,#DISP1C
05E9 F0               MOVX @DPTR,A
05EA E8               MOV A,R0
05EB 54F0             ANL A,#11110000B ; Only 4-High,Bit
05ED 23               RL A
05EE 23               RL A
05EF 23               RL A
05F0 23               RL A
05F1 FA               MOV R2 ,A
05F2 E9               MOV A,R1
05F3 540F             ANL A,#00001111B
05F5 B40016           CJNE A,#00H,Look2
05F8 EA               MOV A,R2      ;buff
05F9 900C50           MOV DPTR ,#TABLE2
05FC 93               MOVC A,@A+DPTR
05FD 90A001           MOV DPTR ,#DISP1B
0600 F0               MOVX @DPTR ,A
0601 E9               MOV A,R1
0602 540F             ANL A,#00001111B  ; Only 4-Low,Bit
0604 900C50           MOV DPTR ,#TABLE2
0607 93               MOVC A,@A+DPTR    ; Look TABLE1
0608 90A000           MOV DPTR ,#DISP1A
060B F0               MOVX @DPTR,A
060C 8014             SJMP ends1
060E EA     Look2:    MOV A,R2      ;buff
060F 900C40           MOV DPTR ,#TABLE1
0612 93               MOVC A,@A+DPTR
0613 90A001           MOV DPTR ,#DISP1B
0616 F0               MOVX @DPTR ,A
0617 E9               MOV A,R1
0618 540F             ANL A,#00001111B  ; Only 4-Low,Bit
061A 900C40           MOV DPTR ,#TABLE1
061D 93               MOVC A,@A+DPTR    ; Look TABLE1
061E 90A000           MOV DPTR ,#DISP1A
0621 F0               MOVX @DPTR,A
0622 D002   ends1:    POP R2
0624 22               RET

0625      ShowDinP2:
        ;-----------
        ; R0->Digit2
        ; R1->Digit3
        ; Blank in front
        ;-----------
0625 C002             PUSH R2

0627 E8               MOV A,R0
0628 900C40           MOV DPTR ,#TABLE1  ;I-DIGIT2,DIGIT3
062B 540F             ANL A,#00001111B  ; Only 4-Low,Bit
062D 93               MOVC A,@A+DPTR    ; Look TABLE1
062E 90B002           MOV DPTR ,#DISP2C
0631 F0               MOVX @DPTR,A
0632 E8               MOV A,R0
0633 54F0             ANL A,#11110000B  ; Only 4-High,Bit
0635 23               RL A
0636 23               RL A
0637 23               RL A
0638 23               RL A
0639 FA               MOV R2 ,A
063A E9               MOV A,R1
063B 540F             ANL A,#00001111B
063D B40016           CJNE A,#00H,look22
0640 EA               MOV A,R2      ;buff
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0641 900C50          MOV DPTR,#TABLE2
0644 93              MOVC A,@A+DPTR
0645 90B001          MOV DPTR,#DISP2B
0648 F0              MOVX @DPTR,A
0649 E9              MOV A,R1
064A 540F            ANL A,#00001111B  ; Only 4-Low,Bit
064C 900C50          MOV DPTR,#TABLE2
064F 93              MOVC A,@A+DPTR    ; Look TABLE1
0650 90B000          MOV DPTR,#DISP2A
0653 F0              MOVX @DPTR,A
0654 8014            SJMP enda2
0656 EA      Lookt22: MOV A,R2         ;buff
0657 900C40          MOV DPTR,#TABLE1
065A 93              MOVC A,@A+DPTR
065B 90B001          MOV DPTR,#DISP2B
065E F0              MOVX @DPTR,A
065F E9              MOV A,R1
0660 540F            ANL A,#00001111B  ; Only 4-Low,Bit
0662 900C40          MOV DPTR,#TABLE1
0665 93              MOVC A,@A+DPTR    ; Look TABLE1
0666 90B000          MOV DPTR,#DISP2A
0669 F0              MOVX @DPTR,A

066A D002    enda2:  POP R2
066C 22              RET

066D         ShowDisP3:
             ;-----------
             ; R0->Digit2
             ; R1->Digit3
             ; Blank and dot in front
             ;-----------
066D C002            PUSH R2

066F E8              MOV A,R0
0670 900C40          MOV DPTR,#TABLE1  ;[-DIGIT2,DIGIT3
0673 540F            ANL A,#00001111B  ; Only 4-Low,Bit
0675 93              MOVC A,@A+DPTR    ; Look TABLE1
0676 90B002          MOV DPTR,#DISP2C
0679 F0              MOVX @DPTR,A
067A E8              MOV A,R0
067B 54F0            ANL A,#11110000B  ; Only 4-High,Bit
067D 23              RL A
067E 23              RL A
067F 23              RL A
0680 23              RL A
0681 FA              MOV R2,A
0682 E9              MOV A,R1
0683 540F            ANL A,#00001111B
0685 B40018          CJNE A,#00H,lookt23       ;chk front zero
0688 EA              MOV A,R2          ;buff
0689 900C50          MOV DPTR,#TABLE2
068C 93              MOVC A,@A+DPTR
068D 4480            ORL A,#10000000B          ;digit point
068F 90B001          MOV DPTR,#DISP2B
0692 F0              MOVX @DPTR,A
0693 E9              MOV A,R1
0694 540F            ANL A,#00001111B  ; Only 4-Low,Bit
0696 900C50          MOV DPTR,#TABLE2
0699 93              MOVC A,@A+DPTR    ; Look TABLE1
069A 90B000          MOV DPTR,#DISP2A
069D F0              MOVX @DPTR,A
069E 8016            SJMP enda3
06A0 EA      Lookt23: MOV A,R2         ;buff
06A1 900C40          MOV DPTR,#TABLE1
06A4 93              MOVC A,@A+DPTR
06A5 4480            ORL A,#10000000B
06A7 90B001          MOV DPTR,#DISP2B
06AA F0              MOVX @DPTR,A
06AB E9              MOV A,R1
06AC 540F            ANL A,#00001111B  ; Only 4-Low,Bit
06AE 900C40          MOV DPTR,#TABLE1
06B1 93              MOVC A,@A+DPTR    ; Look TABLE1
06B2 90B000          MOV DPTR,#DISP2A
06B5 F0              MOVX @DPTR,A

06B6 D002    enda3:  POP R2
06B8 22              RET

06B9         HEXtoDEC:
             ;-----------
             ; Code conversion HEX to DECIMAL
             ;-----------
06B9 C002            PUSH R2
06BB C003            PUSH R3
06BD C004            PUSH R4
06BF C007            PUSH R7

06C1 C3              CLR C
06C2 7402            MOV A,#02D        ;(In R0x1,R1x10,A, Out R0,R1)
06C4 1201AB          LCALL MUL16_8
06C7 AA00            MOV R2,R0         ;store R0
06C9 AB01            MOV R3,R1
06CB 74C8            MOV A,#0200D      ; A=200
06CD 12018B          LCALL SUB16_8     ;(In R0,R1,A,Out R0,R1)
06D0 4014            JC Work1
06D2 74FE            MOV A,#254D
06D4 12018B          LCALL SUB16_8
06D7 401B            JC Work2
06D9 74C8            MOV A,#200D
06DB 12018B          LCALL SUB16_8
06DE 4022            JC Work3
06E0 A802            MOV R0,R2
06E2 A903            MOV R1,R3
06E4 E110            AJMP LastH
06E6 900788  Work1:  MOV DPTR,#TableHD0
06E9 AC02            MOV R4,R2         ;R4 = buff
06EB EC              MOV A,R4
06EC 93              MOVC A,@A+DPTR
06ED F9              MOV R1,A
06EE EC              MOV A,R4
06EF 04              INC A
06F0 93              MOVC A,@A+DPTR
06F1 F8              MOV R0,A
06F2 801C            SJMP LastH
06F4 900850  Work2:  MOV DPTR,#TableHD1
06F7 AC00            MOV R4,R0
06F9 EC              MOV A,R4
06FA 93              MOVC A,@A+DPTR
06FB F9              MOV R1,A
06FC EC              MOV A,R4
06FD 04              INC A
06FE 93              MOVC A,@A+DPTR
06FF F8              MOV R0,A
0700 800E            SJMP LASTH
0702 90091A  Work3:  MOV DPTR,#TableHD2
0705 AC00            MOV R4,R0
0707 EC              MOV A,R4
0708 93              MOVC A,@A+DPTR
0709 F9              MOV R1,A
070A EC              MOV A,R4
070B 04              INC A
070C 93              MOVC A,@A+DPTR
070D F8              MOV R0,A
070E 8000            SJMP LASTH
0710 C3      lastH:  CLR C

0711 D007            POP R7
0713 D004            POP R4
0715 D003            POP R3
0717 D002            POP R2
0719 22              RET

071A         HEXtoPER:
             ;-----------
             ; conversion code HEX to PERCENT
             ;-----------
071A C002            PUSH R2
071C C003            PUSH R3
071E C004            PUSH R4
0720 C007            PUSH R7
0722 C3              CLR C
0723 7402            MOV A,#02D        ;(In R0x1,R1x10,A, Out R0,R1)
0725 1201AB          LCALL MUL16_8
0728 AA00            MOV R2,R0         ;store R0
072A AB01            MOV R3,R1
072C 74C8            MOV A,#0200D      ; A=200
072E 12018B          LCALL SUB16_8     ;(In R0,R1,A,Out R0,R1)
0731 4014            JC WorkP1
0733 74FE            MOV A,#254D
0735 12018B          LCALL SUB16_8
0738 401B            JC WorkP2
073A 74C8            MOV A,#200D
073C 12018B          LCALL SUB16_8
073F 4022            JC WorkP3
0741 A802            MOV R0,R2
0743 A903            MOV R1,R3
0745 E171            AJMP LastHP
0747 9009E4  WorkP1: MOV DPTR,#TableHP0
074A AC02            MOV R4,R2         ;R4 = buff
074C EC              MOV A,R4
074D 93              MOVC A,@A+DPTR
074E F9              MOV R1,A
074F EC              MOV A,R4
0750 04              INC A
0751 93              MOVC A,@A+DPTR
0752 F8              MOV R0,A
0753 80BB            SJMP LastH
0755 900AAC  WorkP2: MOV DPTR,#TableHP1
0758 AC00            MOV R4,R0
075A EC              MOV A,R4
075B 93              MOVC A,@A+DPTR
075C F9              MOV R1,A
075D EC              MOV A,R4
075E 04              INC A
075F 93              MOVC A,@A+DPTR
0760 F8              MOV R0,A
0761 800E            SJMP LASTHP
0763 900B76  WorkP3: MOV DPTR,#TableHP2
0766 AC00            MOV R4,R0
0768 EC              MOV A,R4
0769 93              MOVC A,@A+DPTR
076A F9              MOV R1,A
076B EC              MOV A,R4
076C 04              INC A
076D 93              MOVC A,@A+DPTR
```

ลิขสิทธิ์เป็นของจุฬาลงกรณ์มหาวิทยาลัย สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น ยกเว้นห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของลิขสิทธิ์เอกสารทุกครั้งที่มีการนำไปใช้

```
076E F8             MOV R0,A
076F 8000           SJMP LASTHP
0771 C3     lastHP: CLR C
0772 D007           POP R7
0774 D004           POP R4
0776 D003           POP R3
0778 D002           POP R2
077A 22             RET

077B        initDisp:
            ;--------
            ; display setup
            ;--------
077B 90A003         MOV DPTR, #DISP1D
077E 7480           MOV A  , #80H
0780 F0             MOVX @DPTR, A
0781 90B003         MOV DPTR, #DISP2D
0784 7480           MOV A  , #80H
0786 F0             MOVX @DPTR,A
0787 22             RET

            ;-------------
            ;  Table Routine
            ;-------------
0788        TableHD0:
0788 0000000100     DFB 00H,00H,00H,01H,00H,02H,00H,03H,00H,04H
;0
0792 0005000600     DFB 00H,05H,00H,06H,00H,07H,00H,08H,00H,09H
079C 0010001100     DFB 00H,10H,00H,11H,00H,12H,00H,13H,00H,14H
;10
07A6 0015001600     DFB 00H,15H,00H,16H,00H,17H,00H,18H,00H,19H
07B0 0020002100     DFB 00H,20H,00H,21H,00H,22H,00H,23H,00H,24H
;20
07BA 0025002600     DFB 00H,25H,00H,26H,00H,27H,00H,28H,00H,29H
07C4 0030003100     DFB 00H,30H,00H,31H,00H,32H,00H,33H,00H,34H
;30
07CE 0035003600     DFB 00H,35H,00H,36H,00H,37H,00H,38H,00H,39H
07D8 0040004100     DFB 00H,40H,00H,41H,00H,42H,00H,43H,00H,44H
;40
07E2 0045004600     DFB 00H,45H,00H,46H,00H,47H,00H,48H,00H,49H
07EC 0050005100     DFB 00H,50H,00H,51H,00H,52H,00H,53H,00H,54H
;50
07F6 0055005600     DFB 00H,55H,00H,56H,00H,57H,00H,58H,00H,59H
0800 0060006100     DFB 00H,60H,00H,61H,00H,62H,00H,63H,00H,64H
;60
080A 0065006600     DFB 00H,65H,00H,66H,00H,67H,00H,68H,00H,69H
0814 0070007100     DFB 00H,70H,00H,71H,00H,72H,00H,73H,00H,74H
;70
081E 0075007600     DFB 00H,75H,00H,76H,00H,77H,00H,78H,00H,79H
0828 0080008100     DFB 00H,80H,00H,81H,00H,82H,00H,83H,00H,84H
;80
0832 0085008600     DFB 00H,85H,00H,86H,00H,87H,00H,88H,00H,89H
083C 0090009100     DFB 00H,90H,00H,91H,00H,92H,00H,93H,00H,94H
;90
0846 0095009600     DFB 00H,95H,00H,96H,00H,97H,00H,98H,00H,99H

0850        TableHD1:
0850 0000           DFB 00H,00H  ;compensation
0852 0100010101     DFB 01H,00H,01H,01H,01H,02H,01H,03H,01H,04H
;100
085C 0105010601     DFB 01H,05H,01H,06H,01H,07H,01H,08H,01H,09H
0866 0110011101     DFB 01H,10H,01H,11H,01H,12H,01H,13H,01H,14H
;110
0870 0115011601     DFB 01H,15H,01H,16H,01H,17H,01H,18H,01H,19H
087A 0120012101     DFB 01H,20H,01H,21H,01H,22H,01H,23H,01H,24H
;120
0884 0125012601     DFB 01H,25H,01H,26H,01H,27H,01H,28H,01H,29H
088E 0130013101     DFB 01H,30H,01H,31H,01H,32H,01H,33H,01H,34H
;130
0898 0135013601     DFB 01H,35H,01H,36H,01H,37H,01H,38H,01H,39H
08A2 0140014101     DFB 01H,40H,01H,41H,01H,42H,01H,43H,01H,44H
;140
08AC 0145014601     DFB 01H,45H,01H,46H,01H,47H,01H,48H,01H,49H
08B6 0150015101     DFB 01H,50H,01H,51H,01H,52H,01H,53H,01H,54H
;150
08C0 0155015601     DFB 01H,55H,01H,56H,01H,57H,01H,58H,01H,59H
08CA 0160016101     DFB 01H,60H,01H,61H,01H,62H,01H,63H,01H,64H
;160
08D4 0165016601     DFB 01H,65H,01H,66H,01H,67H,01H,68H,01H,69H
08DE 0170017101     DFB 01H,70H,01H,71H,01H,72H,01H,73H,01H,74H
;170
08E8 0175017601     DFB 01H,75H,01H,76H,01H,77H,01H,78H,01H,79H
08F2 0180018101     DFB 01H,80H,01H,81H,01H,82H,01H,83H,01H,84H
;180
08FC 0185018601     DFB 01H,85H,01H,86H,01H,87H,01H,88H,01H,89H
0906 0190019101     DFB 01H,90H,01H,91H,01H,92H,01H,93H,01H,94H
;190
0910 0195019601     DFB 01H,95H,01H,96H,01H,97H,01H,98H,01H,99H

091A        TableHD2:
091A 0200020102     DFB 02H,00H,02H,01H,02H,02H,02H,03H,02H,04H
;200
0924 0205020602     DFB 02H,05H,02H,06H,02H,07H,02H,08H,02H,09H
092E 0210021102     DFB 02H,10H,02H,11H,02H,12H,02H,13H,02H,14H
;210
0938 0215021602     DFB 02H,15H,02H,16H,02H,17H,02H,18H,02H,19H
0942 0220022102     DFB 02H,20H,02H,21H,02H,22H,02H,23H,02H,24H
;220
094C 02250226       DFB 02H,25H,02H,26H
0950 0227           DFB 02H,27H  ;compensate
0952 0227022802     DFB 02H,27H,02H,28H,02H,29H
0958 0230023102     DFB 02H,30H,02H,31H,02H,32H,02H,33H,02H,34H
;230
0962 0235023602     DFB 02H,35H,02H,36H,02H,37H,02H,38H,02H,39H
096C 0240024102     DFB 02H,40H,02H,41H,02H,42H,02H,43H,02H,44H
;240
0976 0245024602     DFB 02H,45H,02H,46H,02H,47H,02H,48H,02H,49H
0980 0250025102     DFB 02H,50H,02H,51H,02H,52H,02H,53H,02H,54H
;250
098A 0255025602     DFB 02H,55H,02H,56H,02H,57H,02H,58H,02H,59H
0994 0260026102     DFB 02H,60H,02H,61H,02H,62H,02H,63H,02H,64H
;260
099E 0265026602     DFB 02H,65H,02H,66H,02H,67H,02H,68H,02H,69H
09A8 0270027102     DFB 02H,70H,02H,71H,02H,72H,02H,73H,02H,74H
;270
09B2 0275027602     DFB 02H,75H,02H,76H,02H,77H,02H,78H,02H,79H
09BC 0280028102     DFB 02H,80H,02H,81H,02H,82H,02H,83H,02H,84H
;280
09C6 0285028602     DFB 02H,85H,02H,86H,02H,87H,02H,88H,02H,89H
09D0 0290029102     DFB 02H,90H,02H,91H,02H,92H,02H,93H,02H,94H
;290
09DA 0295029602     DFB 02H,95H,02H,96H,02H,97H,02H,98H,02H,99H

09E4        TableHP0:
09E4 0000000500     DFB 00H,00H, 00H,05H, 00H,10H, 00H,10H, 00H,15H
;0
09EE 0020002500     DFB 00H,20H, 00H,25H, 00H,25H, 00H,30H,
00H,35H
09F8 0040004500     DFB 00H,40H, 00H,45H, 00H,45H, 00H,51H, 00H,55H
;10
0A02 0060006000     DFB 00H,60H, 00H,60H, 00H,65H, 00H,70H,
00H,75H
0A0C 0080008000     DFB 00H,80H, 00H,80H, 00H,85H, 00H,90H,
00H,95H  ;20
0A16 0100010001     DFB 01H,00H, 01H,00H, 01H,05H, 01H,10H,
01H,15H
0A20 0120012001     DFB 01H,20H, 01H,20H, 01H,25H, 01H,30H,
01H,35H  ;30
0A2A 0135014001     DFB 01H,35H, 01H,40H, 01H,45H, 01H,50H,
01H,55H
0A34 0155016001     DFB 01H,55H, 01H,60H, 01H,65H, 01H,70H,
01H,70H  ;40
0A3E 0175018001     DFB 01H,75H, 01H,80H, 01H,85H, 01H,90H,
01H,90H
0A48 0195020002     DFB 01H,95H, 02H,00H, 02H,05H, 02H,10H,
02H,10H  ;50
0A52 0215022002     DFB 02H,15H, 02H,20H, 02H,25H, 02H,25H,
02H,30H
0A5C 0235024002     DFB 02H,35H, 02H,40H, 02H,45H, 02H,45H,
02H,50H  ;60
0A66 0255026002     DFB 02H,55H, 02H,60H, 02H,65H, 02H,65H,
02H,70H
0A70 0275028002     DFB 02H,75H, 02H,80H, 02H,80H, 02H,85H,
02H,90H  ;70
0A7A 0295030003     DFB 02H,95H, 03H,00H, 03H,00H, 03H,05H,
03H,10H
0A84 0315032003     DFB 03H,15H, 03H,20H, 03H,20H, 03H,25H,
03H,30H  ;80
0A8E 0335033503     DFB 03H,35H, 03H,35H, 03H,40H, 03H,45H,
03H,50H
0A98 0355035503     DFB 03H,55H, 03H,55H, 03H,60H, 03H,65H,
03H,70H  ;90
0AA2 0370037503     DFB 03H,70H, 03H,75H, 03H,80H, 03H,85H,
03H,90H

0AAC        TableHP1:
0AAC 0000           DFB 00H,00H  ;compensation
0AAE 0390039504     DFB 03H,90H, 03H,95H, 04H,00H, 04H,05H,
04H,10H  ;100
0AB8 0410041504     DFB 04H,10H, 04H,15H, 04H,20H, 04H,25H,
04H,25H
0AC2 0430043504     DFB 04H,30H, 04H,35H, 04H,40H, 04H,45H,
04H,45H  ;110
0ACC 0450045504     DFB 04H,50H, 04H,55H, 04H,60H, 04H,65H,
04H,65H
0AD6 0470047504     DFB 04H,70H, 04H,75H, 04H,80H, 04H,80H,
04H,85H  ;120
0AE0 0490049505     DFB 04H,90H, 04H,95H, 05H,00H, 05H,00H,
05H,05H
0AEA 0510051505     DFB 05H,10H, 05H,15H, 05H,20H, 05H,25H,
05H,25H  ;130
0AF4 0530053505     DFB 05H,30H, 05H,35H, 05H,35H, 05H,40H,
05H,45H
0AFE 0550055505     DFB 05H,50H, 05H,55H, 05H,55H, 05H,60H,
05H,65H  ;140
0B08 0570057005     DFB 05H,70H, 05H,70H, 05H,75H, 05H,80H,
05H,85H
0B12 0590059006     DFB 05H,90H, 05H,90H, 06H,00H, 06H,00H,
06H,05H  ;150
0B1C 0610061006     DFB 06H,10H, 06H,10H, 06H,15H, 06H,20H,
06H,25H
0B26 0625063006     DFB 06H,25H, 06H,30H, 06H,35H, 06H,40H,
06H,45H  ;160
0B30 0645065006     DFB 06H,45H, 06H,50H, 06H,55H, 06H,60H,
06H,65H
```

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0B3A 0665067006     DFB 06H,65H, 06H,70H, 06H,75H, 06H,80H,
06H,80H  ;170
0B44 0685069006     DFB 06H,85H, 06H,90H, 06H,95H, 07H,00H,
07H,00H
0B4E 0705071007     DFB 07H,05H, 07H,10H, 07H,15H, 07H,20H,
07H,20H  ;180
0B58 0725073007     DFB 07H,25H, 07H,30H, 07H,35H, 07H,40H,
07H,40H
0B62 0745075007     DFB 07H,45H, 07H,50H, 07H,55H, 07H,55H,
07H,60H  ;190
0B6C 0765077007     DFB 07H,65H, 07H,70H, 07H,70H, 07H,75H,
07H,80H

0B76        TableHP2:
0B76 0785079007     DFB 07H,85H, 07H,90H, 07H,90H, 07H,95H,
08H,00H  ;200
0B80 0805081008     DFB 08H,05H, 08H,10H, 08H,10H, 08H,15H,
08H,20H
0B8A 0825082508     DFB 08H,25H, 08H,25H, 08H,30H, 08H,35H,
08H,40H  ;210
0B94 0845085008     DFB 08H,45H, 08H,50H, 08H,50H, 08H,55H,
08H,60H
0B9E 0865086508     DFB 08H,65H, 08H,65H, 08H,70H, 08H,75H,
08H,80H  ;220
0BA8 0885008B5      DFB 08H,80H, 08H,85H
0BAC 0890           DFB 08H,90H  ;compensate
0BAE 0890089509     DFB 08H,90H, 08H,95H, 09H,00H
0BB4 0900090509     DFB 09H,00H, 09H,05H, 09H,05H, 09H,10H,
09H,15H  ;230
0BBE 0920092509     DFB 09H,20H, 09H,25H, 09H,30H, 09H,35H,
09H,40H
0BC8 0940094509     DFB 09H,40H, 09H,45H, 09H,50H, 09H,55H,
09H,55H  ;240
0BD2 0960096509     DFB 09H,60H, 09H,65H, 09H,70H, 09H,70H,
09H,75H
0BDC 0980098509     DFB 09H,80H, 09H,85H, 09H,90H, 09H,90H,
09H,95H  ;250
0BE6 0999025602     DFB 09H,99H, 02H,56H, 02H,57H, 02H,58H,
02H,59H
0BF0 0260026102     DFB 02H,60H, 02H,61H, 02H,62H, 02H,63H,
02H,64H  ;260
0BFA 0265026602     DFB 02H,65H, 02H,66H, 02H,67H, 02H,68H,
02H,69H
0C04 0270027102     DFB 02H,70H, 02H,71H, 02H,72H, 02H,73H,
02H,74H  ;270
0C0E 0275027602     DFB 02H,75H, 02H,76H, 02H,77H, 02H,78H,
02H,79H
0C18 0280028102     DFB 02H,80H, 02H,81H, 02H,82H, 02H,83H,
02H,84H  ;280
0C22 0285028602     DFB 02H,85H, 02H,86H, 02H,87H, 02H,88H,
02H,89H
0C2C 0290029102     DFB 02H,90H, 02H,91H, 02H,92H, 02H,93H,
02H,94H  ;290
0C36 0295029602     DFB 02H,95H, 02H,96H, 02H,97H, 02H,98H,
02H,99H

        ; Table1 HEX
        ;-------
0C40        Table1:
          ;  hgfedcba
0C40 3F       DFB 00111111B   ;0
0C41 18       DFB 00011000B   ;1
0C42 76       DFB 01110110B   ;2
0C43 7C       DFB 01111100B   ;3
0C44 59       DFB 01011001B   ;4
0C45 6D       DFB 01101101B   ;5
0C46 6F       DFB 01101111B   ;6
0C47 38       DFB 00111000B   ;7
0C48 7F       DFB 01111111B   ;8
0C49 7D       DFB 01111101B   ;9
0C4A 7B       DFB 01111011B   ;A
0C4B 4F       DFB 01001111B   ;B
0C4C 27       DFB 00100111B   ;C
0C4D 5E       DFB 01011110B   ;D
0C4E 67       DFB 01100111B   ;E
0C4F 63       DFB 01100011B   ;F

0C50        Table2:
          ;  hgfedcba
0C50 00       DFB 00000000B   ;Blank
0C51 18       DFB 00011000B   ;1
0C52 76       DFB 01110110B   ;2
0C53 7C       DFB 01111100B   ;3
0C54 59       DFB 01011001B   ;4
0C55 6D       DFB 01101101B   ;5
0C56 6F       DFB 01101111B   ;6
0C57 38       DFB 00111000B   ;7
0C58 7F       DFB 01111111B   ;8
0C59 7D       DFB 01111101B   ;9
0C5A 7B       DFB 01111011B   ;A
0C5B 4F       DFB 01001111B   ;B
0C5C 27       DFB 00100111B   ;C
0C5D 5E       DFB 01011110B   ;D
0C5E 67       DFB 01100111B   ;E
0C5F 63       DFB 01100011B   ;F
        ;-------
        ; End
```

```
        ;-------


0C60            INCL "AllKtabl.asm"
        ;-------------
        ; routine for K-selection
        ;-------------

        ; Num1Fg:   EQU       7H

0C60      lookAtKtabel:
        ;-------------
        ; R1 = K1 ,R0 = K1              R4  R3
        ; A : count                    | |
        ; R4 = FontDisp  and DigitCode      8 88
        ; R3 = BackDisp
        ;-------------
0C60 C007           PUSH R7
0C62 FF             MOV R7,A          ;data to buffer
0C63 900D3E           MOV DPTR,#TableK0
0C66 EF             MOV A,R7
0C67 93             MOVC A,@A+DPTR
0C68 F8             MOV R0,A
          ;  ----
0C69 900DC0           MOV DPTR,#TableK1
0C6C EF             MOV A,R7
0C6D 93             MOVC A,@A+DPTR
0C6E F9             MOV R1,A
          ;  ----
0C6F 900E2E           MOV DPTR,#TablePK0
0C72 EF             MOV A,R7
0C73 93             MOVC A,@A+DPTR
0C74 FB             MOV R3,A
          ;  ----
0C75 900E9C           MOV DPTR,#TablePK1
0C78 EF             MOV A,R7
0C79 93             MOVC A,@A+DPTR
0C7A FC             MOV R4,A
          ;  ----
0C7B D007           POP R7
0C7D 22             RET

0C7E      ShowKtab:
        ;-------------------------
        ; Show current gain of varible to three last LED
        ; R4: Font
        ; R3: back
        ; and numberdot in MSB of R4
        ;-------------------------
        ;       MOV DPTR,#DISP2D
        ;       MOV A,#80H      ; Set 8255
        ;       MOVX @DPTR,A
0C7E EC             MOV A,R4
0C7F 54F0           ANL A,#11110000B  ; Find digit Code in R4
0C81 C4             SWAP A
0C82 B40002           CJNE A,#00H,notDigit0
0C85 800C           SJMP showDg0
0C87 B40102   notDigit0:   CJNE A,#01H,notDigit1
0C8A 8042           SJMP ShowDg1
0C8C B40202   notDigit1:   CJNE A,#02H,Outnotdig
0C8F 807A           SJMP ShowDg2
0C91 A13C     outnotDig:   AJMP OutDig
          ;  ----
0C93 EC       ShowDg0:     MOV A,R4
0C94 540F             ANL A,#00001111B ;Chek Fist number
0C96 600A             JZ swblank0
0C98 900C40           MOV DPTR ,#Table1
0C9B 93             MOVC A,@A+DPTR        ; |
0C9C 90B000           MOV DPTR,#DISP2A      ; 888
0C9F F0             MOVX @DPTR,A
0CA0 8008           SJMP OutBk0
0CA2 7400     swblank0:  MOV A,#00000000B ;" "
0CA4 90B000           MOV DPTR,#DISP2A
0CA7 F0             MOVX @DPTR,A
0CA8 D207           SETB Num1Fg
0CAA 00       outBk0:    NOP
          ;  ----
0CAB EB             MOV A,R3
0CAC 540F             ANL A,#00001111B
0CAE 900C40           MOV DPTR, #Table1
0CB1 93             MOVC A, @A+DPTR       ; |
0CB2 90B002           MOV DPTR, #DISP2C     ; 888
0CB5 F0             MOVX @DPTR,A
0CB6 EB             MOV A,R3
0CB7 54F0           ANL A,#11110000B
0CB9 C4             SWAP A
0CBA 300707           JNB Num1Fg,LookS0
0CBD 900C50           MOV DPTR,#table2
0CC0 C207           CLR Num1Fg
0CC2 8003           SJMP outLookS0
0CC4 900C40   LooKS0:      MOV DPTR,#Table1
0CC7 93       outLooKs0:   MOVC A,@A+DPTR        ; |
0CC8 90B001           MOV DPTR,#DISP2B      ; 888
0CCB F0             MOVX @DPTR,A
0CCC 806E           SJMP outDig
          ;  ----
0CCE EC     ShowDg1:    MOV A,R4
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0CCF 540F             ANL  A,#00001111B      ; Chek Fist number
0CD1 600A             JZ   swblank1
0CD3 900C40           MOV  DPTR ,#Table1
0CD6 93               MOVC A,@A+DPTR
0CD7 90B000           MOV  DPTR,#DISP2A
0CDA F0               MOVX @DPTR,A
0CDB 8008             SJMP OutBk1
0CDD 7400       swblank1:  MOV  A,#00000000B      ;|
0CDF 90B000           MOV  DPTR,#DISP2A       ; 88.8
0CE2 F0               MOVX @DPTR,A            ; " "
0CE3 D207             SETB Num1Fg
0CE5 00         outBk1:   NOP
          ;     ---------
0CE6 EB               MOV  A,R3
0CE7 540F             ANL  A,#00001111B
0CE9 900C40           MOV  DPTR,#Table1
0CEC 93               MOVC A,@A+DPTR          ;  |
0CED 90B002           MOV  DPTR,#DISP2C       ; 88.8
0CF0 F0               MOVX @DPTR,A
          ;     ---------
0CF1 EB               MOV  A,R3
0CF2 54F0             ANL  A,#11110000B
0CF4 C4               SWAP A                  ;1111 <--> 0000
0CF5 300707           JNB  Num1Fg,LookS1
0CF8 900C50           MOV  DPTR,#table2
0CFB C207             CLR  Num1Fg
0CFD 8003             SJMP outLookS1
0CFF 900C40   LooKS1:     MOV  DPTR,#Table1
0D02 93       outLooKs1:  MOVC A,@A+DPTR
0D03 4480             ORL  A,#10000000B       ; |
0D05 90B001           MOV  DPTR,#DISP2B       ; 88.8
0D08 F0               MOVX @DPTR,A
0D09 8031             SJMP outDig
          ;     ---------
0D0B EC       ShowDg2:    MOV  A,R4
0D0C 540F             ANL  A,#00001111B ;Chek Fist number
0D0E 900C40           MOV  DPTR ,#Table1
0D11 93               MOVC A,@A+DPTR
0D12 4480             ORL  A,#10000000B       ;  |
0D14 90B000           MOV  DPTR,#DISP2A       ;  8.88
0D17 F0               MOVX @DPTR,A
0D18 00         outBk2:   NOP
0D19 EB               MOV  A,R3
0D1A 540F             ANL  A,#00001111B
0D1C 900C40           MOV  DPTR,#Table1
0D1F 93               MOVC A,@A+DPTR          ;  |
0D20 90B002           MOV  DPTR,#DISP2C       ;  8.88
0D23 F0               MOVX @DPTR,A
0D24 EB               MOV  A,R3
0D25 54F0             ANL  A,#11110000B
0D27 C4               SWAP A
0D28 300707           JNB  Num1Fg,LookS2
0D2B 900C50           MOV  DPTR,#table2
0D2E C207             CLR  Num1Fg
0D30 8003             SJMP outLookS2
0D32 900C40   LocKS2:     MOV  DPTR,#Table1
0D35 93       outLooKs2:  MOVC A,@A+DPTR          ;  |
0D36 90B001           MOV  DPTR,#DISP2B       ;  8.88
0D39 F0               MOVX @DPTR,A
0D3A 8000             SJMP outDig
0D3C 00       outdig:     NOP
0D3D 22               RET


0D3E          tableK0:    ;value of constant K0
0D3E 0305080A0D    DFB  03D, 05D, 08D, 10D, 13D, 15D, 18D, 20D,
23D, 26D
          ;       .01  .02  .03  .04  .05  .06  .07  .08  .09  .1
0D48 1E24292E33    DFB  30D, 36D, 41D, 46D, 51D, 56D, 62D, 71D,
77D, 89D
          ;       .12  .14  .16  .18  .20  .22  .24  .28  .30  .35
0D52 6673808C99    DFB  102D, 115D, 128D, 140D, 153D, 178D, 204D,
230D, 00D, 26D
          ;       .40  .45  .50  .55  .6   .7   .8   .9   1.0  1.1
0D5C 334D668099    DFB  51D, 77D, 102D, 128D, 153D, 178D, 204D,
230D, 00D, 51D
          ;       1.2  1.3  1.4  1.5  1.6  1.7  1.8  1.9  2.0  2.1
0D66 336699CC00    DFB  51D, 102D, 153D, 204D, 00D, 51D, 102D,
153D, 204D, 00D
          ;       2.2  2.4  2.6  2.8  3.0  3.2  3.4  3.6  3.8  4.0
0D70 336699CC00    DFB  51D, 102D, 153D, 204D, 00D, 51D, 00D,
00D, 00D, 00D
          ;       4.2  4.4  4.6  4.8  5.0  5.5  6.0  6.5  7.0  8.0
0D7A 8000800000    DFB  128D, 00D, 128D, 00D, 00D, 00D, 00D, 00D,
00D, 00D
          ;       8.5  9.0  9.5  10.0 11.0 11.0 .0   .0   .0   .0
0D84 0000000000    DFB  00D, 00D, 00D, 00D, 00D, 00D, 00D, 00D,
00D, 00D
          ;       .0   .0   .0   .0   .0   .0   .0   .0   .0   .0
0D8E 0000000000    DFB  00D, 00D, 00D, 00D, 00D, 00D, 00D, 00D,
00D, 00D
          ;       .0   .0   .0   .0   .0   .0   .0   .0   .0   .0
0D98 0000000000    DFB  00D, 00D, 00D, 00D, 00D, 00D, 00D, 00D,
00D, 00D
          ;       .0   .0   .0   .0   .0   .0   .0   .0   .0   .0
0DA2 0000000000    DFB  00D, 00D, 00D, 00D, 00D, 00D, 00D, 00D,
00D, 00D
          ;       .0   .0   .0   .0   .0   .0   .0   .0   .0   .0

0DAC 0000000000    DFB  00D, 00D, 00D, 00D, 00D, 00D, 00D, 00D,
00D, 00D
          ;       .0   .0   .0   .0   .0   .0   .0   .0   .0   .0
0DB6 0000000000    DFB  00D, 00D, 00D, 00D, 00D, 00D, 00D, 00D,
00D, 00D
          ;       .0   .0   .0   .0   .0   .0   .0   .0   .0   .0

0DC0          tableK1:    ;value of value of K1
0DC0 0000000000    DFB  00D, 00D, 00D, 00D, 00D, 00D, 00D, 00D,
00D, 00D
          ;       .01  .02  .03  .04  .05  .06  .07  .08  .09  .1
0DCA 0000000000    DFB  00D, 00D, 00D, 00D, 00D, 00D, 00D, 00D,
00D, 00D
          ;       .12  .14  .16  .18  .20  .22  .24  .28  .30  .35
0DD4 0000000000    DFB  00D, 00D, 00D, 00D, 00D, 00D, 00D, 00D,
01D, 01D
          ;       .40  .45  .50  .55  .60  .70  .80  .90  1.0  1.1
0DDE 0101010101    DFB  01D, 01D, 01D, 01D, 01D, 01D, 01D, 01D,
02D, 02D
          ;       1.2  1.3  1.4  1.5  1.6  1.7  1.8  1.9  2.0  2.1
0DE8 0202020203    DFB  02D, 02D, 02D, 02D, 03D, 03D, 03D, 03D,
03D, 04D
          ;       2.2  2.4  2.6  2.8  3.0  3.2  3.4  3.6  3.8  4.0
0DF2 0404040405    DFB  04D, 04D, 04D, 04D, 05D, 05D, 06D, 06D,
07D, 08D
          ;       4.2  4.4  4.6  4.8  5.0  5.5  6.0  6.5  7.0  8.0
0DFC 0809090A0B    DFB  08D, 09D, 09D, 10D, 11D, 12D, 13D, 14D,
15D, 16D
          ;       8.5  9.0  9.5  10.0 11.0 12.0 13   14   15   16
0E06 1112131416    DFB  17D, 18D, 19D, 20D, 22D, 24D, 26D, 28D,
30D, 32D
          ;       17   18   19   20   22   24   26   28   30   32
0E10 222426282A    DFB  34D, 36D, 38D, 40D, 42D, 44D, 46D, 48D,
50D, 55D
          ;       34   36   38   40   42   44   46   48   50   55
0E1A 3C41464B50    DFB  60D, 65D, 70D, 75D, 80D, 85D, 90D, 95D,
100D, 110D
          ;       60   65   70   75   80   85   90   95   100  110
0E24 78828C96A0    DFB  120D, 130D, 140D, 150D, 160D, 170D, 180D,
190D, 200D, 00D
          ;       120  130  140  150  160  170  180  190  200  00

0E2E          tablePK0:   ;value of disp
0E2E 0102030405    DFB  01h, 02h, 03h, 04h, 05h, 06h, 07h, 08h, 09h,
10h
          ;       .01  .02  .03  .04  .05  .06  .07  .08  .09  .10
0E38 1214161820    DFB  12h, 14h, 16h, 18h, 20h, 22h, 24h, 28h, 30h,
35h
          ;       .12  .14  .16  .18  .20  .22  .24  .28  .30  .35
0E42 4045505560    DFB  40h, 45h, 50h, 55h, 60h, 70h, 80h, 90h, 10h,
11h
          ;       .40  .45  .50  .55  .60  .70  .80  .90  1.0  1.1
0E4C 1213141516    DFB  12h, 13h, 14h, 15h, 16h, 17h, 18h, 19h,
20h, 21h
          ;       1.2  1.3  1.4  1.5  1.6  1.7  1.8  1.9  2.0  2.1
0E56 2224262830    DFB  22h, 24h, 26h, 28h, 30h, 32h, 34h, 36h, 38h,
40h
          ;       2.2  2.4  2.6  2.8  3.0  3.2  3.4  3.6  3.8  4.0
0E60 4244464850    DFB  42h, 44h, 46h, 48h, 50h, 55h, 60h, 65h, 70h,
80h
          ;       4.2  4.4  4.6  4.8  5.0  5.5  6.0  6.5  7.0  8.0
0E6A 8590951011    DFB  85h, 90h, 95h, 10h, 11h, 12h, 13h, 14h, 15h,
16h
          ;       8.5  9.0  9.5  10   11   12   13   14   15   16
0E74 1718192022    DFB  17h, 18h, 19h, 20h, 22h, 24h, 26h, 28h, 30h,
32h
          ;       17   18   19   20   22   24   26   28   30   32
0E7E 3436384042    DFB  34h, 36h, 38h, 40h, 42h, 44h, 46h, 48h, 50h,
55h
          ;       34   36   38   40   42   44   46   48   50   55
0E88 6065707580    DFB  60h, 65h, 70h, 75h, 80h, 85h, 90h, 95h, 00h,
10h
          ;       60   65   70   75   80   85   90   95   100  110
0E92 2030405060    DFB  20h, 30h, 40h, 50h, 60h, 70h, 80h, 90h, 00h,
00h
          ;       120  130  140  150  160  170  180  190  200  00

0E9C          tablePK1:
          ;-------------------
          ; output is number of digit and
          ; Digit FistNumber
          ;   ||
          ;   X,Xh
          ;-------------------
0E9C 2020202020    DFB  20h, 20h, 20h, 20h, 20h, 20h, 20h, 20h, 20h,
20h
          ;       .01  .02  .03  .04  .05  .06  .07  .08  .09  .10
0EA6 2020202020    DFB  20h, 20h, 20h, 20h, 20h, 20h, 20h, 20h, 20h,
20h
          ;       .12  .14  .16  .18  .20  .22  .24  .28  .30  .35
0EB0 2020202020    DFB  20h, 20h, 20h, 20h, 20h, 20h, 20h, 20h, 10h,
10h
          ;       .40  .45  .50  .55  .60  .70  .80  .90  1.0  1.1
0EBA 1010101010    DFB  10h, 10h, 10h, 10h, 10h, 10h, 10h, 10h, 10h,
10h
          ;       1.2  1.3  1.4  1.5  1.6  1.7  1.8  1.9  2.0  2.1
0EC4 1010101010    DFB  10h, 10h, 10h, 10h, 10h, 10h, 10h, 10h,
10h, 10h
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ;       2.2  2.4  2.6  2.8  3.0  3.2  3.4  3.6  3.8  4.0
0ECE 1010101010   DFB  10h, 10h, 10h, 10h, 10h, 10h, 10h, 10h, 10h, 10h,
10h
        ;       4.2  4.4  4.6  4.8  5.0  5.5  6.0  6.5  7.0  8.0
0ED8 1010100000   DFB  10h, 10h, 10h, 00h, 00h, 00h, 00h, 00h, 00h, 00h,
00h
        ;       8.5  9.0  9.5  10   11   12   13   14   15   16
0EE2 0000000000   DFB  00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h,
00h
        ;       17   18   19   20   22   24   26   28   30   32
0EEC 0000000000   DFB  00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h,
00h
        ;       34   36   38   40   42   44   46   48   50   55
0EF6 0000000000   DFB  00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 00h, 01h,
01h
        ;       60   65   70   75   80   85   90   95   100  110
0F00 0101010101   DFB  01h, 01h, 01h, 01h, 01h, 01h, 01h, 01h, 02h,
00h
        ;       120  130  140  150  160  170  180  190  200  00


0F0A            INCL "IntrKeyz.asm"         ;2
        ;*******************
        ; keybroad routine
        ;*******************

        ; CURBLINKFg:  EQU   04H
        ; EditFg:      EQU   06H
        ;-------
        ; Constant Var
        ;-------
0000 =      UPky:      EQU   00H    ;0
0001 =      DwKy:      EQU   01H    ;1
0002 =      MdKy:      EQU   02H    ;2
0003 =      EnKy:      EQU   03H    ;3
0004 =      SpKy:      EQU   04H    ;4
0005 =      ADky:      EQU   05H    ;5
0006 =      BDky:      EQU   06H    ;6
0007 =      PvKy:      EQU   07H    ;7
0008 =      EdKy:      EQU   08H    ;8
0009 =      KpKy:      EQU   09H    ;9
000A =      KIKy:      EQU   0AH    ;A
000B =      KdKy:      EQU   0BH    ;B

        ; KPCount:    EQU   30H    ;KP-buffer
        ; KDCount:    EQU   31H    ;KD-buffer
        ; KICount:    EQU   32H    ;KI-buffer
        ; MDCount:    EQU   33H    ;Mode
        ; MDWORK:     EQU   34H    ;use for run

        ; ADCount:    EQU   2FH    ;address-buffer
        ; BDCount:    EQU   2EH    ;baud-buffer

0000 =      ENCode:    EQU   00H    ;          0000
0001 =      PVCode:    EQU   01H    ;          0001
0002 =      SPCode:    EQU   02H    ; dddd,XXXX  0010
0003 =      KPCode:    EQU   03H    ;          0011
0004 =      KICode:    EQU   04H    ;          0100
0005 =      KDCode:    EQU   05H    ;          0101
0006 =      MDCode:    EQU   06H    ;          0110
0007 =      ADCode:    EQU   07H    ;          0111
0008 =      BDCode:    EQU   08H    ;          1000

0F0A        IntrIE0_2:
        ;-------
        ; keybroad interupt
        ;-------
0F0A C2A8         CLR  IE_0          ;disable keybroad
0F0C C3           CLR  C
0F0D 909000       MOV  DPTR,#KEYB
0F10 E0           MOVX A,@DPTR
0F11 F537         MOV  KEYBUFF,A     ;Read No. of Key Matice
0F13 B40306       CJNE A,#ENky,notcn ;chk enkey
0F16 1211EE       LCALL EnSt
0F19 020FA6       LJMP  OutSt
0F1C E53A   notcn:  MOV  A,STATUS    ;chk status to work
0F1E 540F         ANL  A,#00001111B
0F20 B40045       CJNE A,#ENCode,ST00  ;if it hasn't status code
0F23 E537         MOV  A,KEYBUFF
        ;---
0F25 B40205       CJNE A,#MDKy,Kyy0
0F28 753A06       MOV  STATUS,#MDcode  ;Put MDcode
0F2B 8079         SJMP  OutSt
        ;-
0F2D B40405 Kyy0:   CJNE A,#SPKy,Kyy1
0F30 753A02       MOV  STATUS,#SPcode  ;Put SPcode
0F33 8071         SJMP  OutSt
        ;-
0F35 B40705 Kyy1:   CJNE A,#PVKy,Kyy2
0F38 753A01       MOV  STATUS,#PVcode  ;Put PVcode
0F3B 8069         SJMP  OutSt
        ;-
0F3D B40905 Kyy2:   CJNE A,#KPKy,Kyy3
0F40 753A03       MOV  STATUS,#KPcode  ;Put KPcode
0F43 8061         SJMP  OutSt
        ;-
0F45 B40A05 Kyy3:   CJNE A,#KIKy,Kyy4
0F48 753A04       MOV  STATUS,#KIcode  ;Put KIcode
0F4B 8059         SJMP  OutSt
        ;-
0F4D B40B05 Kyy4:   CJNE A,#KDKy,Kyy5
0F50 753A05       MOV  STATUS,#KDcode  ;Put KDCode
0F53 8051         SJMP  OutSt
        ;-
0F55 B40505 Kyy5:   CJNE A,#ADKy,Kyy6
0F58 753A07       MOV  STATUS,#ADcode  ;Put ADCode
0F5B 8049         SJMP  OutSt
        ;-
0F5D B40605 Kyy6:   CJNE A,#BDKy,Kyy7
0F60 753A08       MOV  STATUS,#BDcode  ;Put BDCode
0F63 8041         SJMP  OutSt
        ;-
0F65 00     Kyy7:   NOP
0F66 803E         SJMP  OutSt
        ;--------
0F68 B40105 ST00:   CJNE A,#PVcode,ST01
0F6B 120FD1       LCALL PVSt         ;     Seclect
0F6E 8036         SJMP  outSt        ;     ||
        ;-                    0000,0111B
0F70 B40205 ST01:   CJNE A,#SPCode,ST02
0F73 120FF7       LCALL SPSt
0F76 802E         SJMP  OutSt
        ;-
0F78 B40305 ST02:   CJNE A,#KpCode,ST03
0F7B 1210C2       LCALL KPSt
0F7E 8026         SJMP  OutSt
        ;-
0F80 B40405 ST03:   CJNE A,#KICode,ST04
0F83 121126       LCALL KISt
0F86 801E         SJMP  OutSt
        ;-
0F88 B40505 ST04:   CJNE A,#KDcode,ST05
0F8B 12118A       LCALL KDSt
0F8E 8016         SJMP  OutSt
        ;-
0F90 B40605 ST05:   CJNE A,#MDcode,ST06
0F93 121258       LCALL MDst
0F96 800E         SJMP  OutSt
        ;-
0F98 B40705 ST06:   CJNE A,#ADcode,ST07
0F9B 12103C       LCALL ADst
0F9E 8006         SJMP  OutSt
        ;-
0FA0 B40803 ST07:   CJNE A,#BDcode,OutSt
0FA3 12107E       LCALL BDst
        ;-
0FA6 00     OutSt:  NOP
0FA7 D2A8         SETB IE_0          ;enable keyboard intr
0FA9 22           RET

0FAA        CurOnOff:
        ;-----------
        ; check cursorBlink
        ;-----------
0FAA 200406       JB CurBlinkFg,NotOn
0FAD 7440         MOV  A,#01000000B  ;"-"
0FAF 90A002       MOV  DPTR,#DISP1C
0FB2 F0           MOVX @DPTR,A
0FB3 00     NotOn:  NOP
0FB4 22           RET

0FB5        ChkEdit:
        ;-----------
        ; Subroutine
        ; for all Key
        ;-----------
0FB5 C206         CLR  EditFg
0FB7 E53A         MOV  A,Status      ;Chk edit code #00010000B
0FB9 5410         ANL  A,#00010000B
0FBB 30E404       JNB  A_4,KeyEd
0FBE D206         SETB EditFg
0FC0 800D         SJMP  OutEd
        ;   ---
0FC2 E537   KeyEd:  MOV  A,KEYBUFF     ;load Keybuff
0FC4 B40808       CJNE A,#EDky,outED
0FC7 D204         SETB CURBLinkFg      ; set Cursorblink in timer
0FC9 E53A         MOV  A,Status
0FCB 4410         ORL  A,#00010000B
0FCD F53A         MOV  Status,A        ;Put edit code in Status
        ;   ---
0FCF 00     OutEd:  NOP
0FD0 22           RET

0FD1        PVSt:
        ;--------
        ; Subroutine of IE0_2
        ; processer routine
        ;--------
0FD1 C000         PUSH R0
0FD3 C001         PUSH R1
0FD5 7473         MOV  A,#01110011B    ;"P"
0FD7 90A000       MOV  DPTR,#DISP1A
0FDA F0           MOVX @DPTR,A
0FDB 740E         MOV  A,#00001110B    ;"V"
0FDD 90A001       MOV  DPTR,#DISP1B
0FE0 F0           MOVX @DPTR,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0FE1 120FAA          LCALL CurOnOff
0FE4 A835            MOV R0,ADBUFF
0FE6 7900            MOV R1,#00H
0FE8 12071A          LCALL HextoPer
0FEB 12066D          LCALL ShowDisp3
0FEE D204            SETB CurBlinkFg
0FF0 D208            SETB PVFg
0FF2 D001            POP R1
0FF4 D000            POP R0
0FF6 22              RET

0FF7      SPst:
          ;-----------
          ; Subroutine IE0_2
          ; SetPoint
          ;-----------
0FF7 746D            MOV  A,#01101101B     ;"S"
0FF9 90A000          MOV  DPTR ,#DISP1A
0FFC F0              MOVX @DPTR,A
0FFD 7473            MOV  A,#01110011B     ;"P"
0FFF 90A001          MOV  DPTR ,#DISP1B
1002 F0              MOVX @DPTR,A
1003 120FAA          LCALL CurOnOFF
1006 A839            MOV R0,SETPOINT
1008 7900            MOV R1,#00H
100A 12071A          LCALL HextoPer        ;            Edit
100D 12066D          LCALL ShowDisp3       ;            |
1010 120FB5          LCALL ChkEdit
1013 300624          JNB EditFg,outChkEDS
          ;   ---
1016 D21A            SETB SPFg             ;for Clear SUM0,SUM1 in ENst
          ;-
1018 E537            MOV A,KEYBUFF         ;load KeyBuff
101A B4000E          CJNE A,#UpKy,Dwwork
101D 0539            INC  SETPOINT         ;SetPoint is buffer,DSP is real
101F A839            MOV  R0,SETPOINT
1021 7900            MOV  R1,#00H
1023 12071A          LCALL HextoPer
1026 12066D          LCALL ShowDisp3
1029 800F            SJMP outChkEDS
          ;   ---
102B B4010C  DwWork: CJNE A,#DwKy ,outChkEDS
102E 1539            DEC  SETPOINT
1030 A839            MOV  R0,SETPOINT
1032 7900            MOV  R1,#00H
1034 12071A          LCALL HextoPer
1037 12066D          LCALL ShowDisp3
103A 00    outChkEDS: NOP
103B 22              RET

103C      ADst:
          ;-----------
          ; Subroutine IE0_2
          ; Sampling time
          ;-----------
103C 747B            MOV  A,#01111011B     ;"A"
103E 90A000          MOV  DPTR ,#DISP1A
1041 F0              MOVX @DPTR,A
1042 745E            MOV  A,#01011110B     ;"D"
1044 90A001          MOV  DPTR ,#DISP1B
1047 F0              MOVX @DPTR,A
          ;-----
1048 120FAA          LCALL CurOnOFF
104B 121305          LCALL ShowAD
104E 120FB5          LCALL ChkEdit
1051 300628          JNB EditFg,outADS
          ;   ---
1054 E537            MOV A,KEYBUFF         ;load KeyBuff
1056 B40012          CJNE A,#UpKy,DwdoAD
1059 052F            INC  ADCount
105B C3              CLR  C
105C 740B            MOV  A,#11D
105E 952F            SUBB A,ADCount
1060 5003            JNC  OutMoreADS
1062 752F00          MOV  ADCount ,#00h    ; ADCount isn't more than
11
1065 00    OutMoreADS:NOP
1066 121305          LCALL SHOWAD
1069 8011            SJMP outADS
          ;   ---
106B 00    DwdoAD:   NOP
106C B401BD          CJNE A,#DwKy ,outADS
106F E52F            MOV  A,ADCount
1071 7003            JNZ  UnADcount
1073 752F0B          MOV  ADCount,#11D
1076 00    UnADcount: NOP
1077 152F            DEC  ADCount
1079 121305          LCALL ShowAD
107C 00    outADS:   NOP
107D 22              RET


107E      BDst:
          ;-----------
          ; Subroutine IE0_2
          ; Sampling time
          ;-----------
107E 744F            MOV  A,#01001111B     ;"B"
1080 90A000          MOV  DPTR ,#DISP1A
1083 F0              MOVX @DPTR,A
1084 745E            MOV  A,#01011110B     ;"D"
1086 90A001          MOV  DPTR ,#DISP1B
1089 F0              MOVX @DPTR,A
          ;-----
108A 120FAA          LCALL CurOnOFF
108D 1213AD          LCALL ShowBD
1090 120FB5          LCALL ChkEdit
1093 30062A          JNB EditFg,outBDS
          ;-----
1096 D219            SETB BaudSetFg        ;for EN key
1098 E537            MOV A,KEYBUFF         ;load KeyBuff
109A B40012          CJNE A,#UpKy,DwdoBD
109D 052E            INC  BDCount
109F C3              CLR  C
10A0 7406            MOV  A,#6D
10A2 952E            SUBB A,BDCount
10A4 5003            JNC  OutMoreBDS
10A6 752E00          MOV  BDCount ,#00h    ; BDCount isn't more than 7
10A9 00    OutMoreBDS:NOP
10AA 1213AD          LCALL SHOWBD
10AD 8011            SJMP outBDS
          ;   ---
10AF 00    DwdoBD:   NOP
10B0 B4010D          CJNE A,#DwKy ,outBDS
10B3 E52E            MOV  A,BDCount
10B5 7003            JNZ  UnBDcount
10B7 752E06          MOV  BDCount,#6D
10BA 00    UnBDcount: NOP
10BB 152E            DEC  BDCount
10BD 1213AD          LCALL ShowBD
10C0 00    outBDS:   NOP
10C1 22              RET


10C2      KPst:
          ;-----------
          ; Subroutine IE0_2
          ; SetPoint
          ;-----------
10C2 C000            PUSH  R0
10C4 C001            PUSH  R1
10C6 C002            PUSH  R2
10C8 C003            PUSH  R3
10CA C004            PUSH  R4
10CC 7457            MOV  A,#01010111B     ;"K"
10CE 90A000          MOV  DPTR ,#DISP1A
10D1 F0              MOVX @DPTR,A
10D2 7473            MOV  A,#01110011B     ;"P"
10D4 90A001          MOV  DPTR ,#DISP1B
10D7 F0              MOVX @DPTR,A
10D8 120FAA          LCALL CurOnOff
10DB E530            MOV  A,KPcount
10DD 120C60          LCALL LookatKtabel    ;K0->R0,K1->R1,R3-
>88,R4->8
10E0 120C7E          LCALL ShowKtab
10E3 120FB5          LCALL ChkEdit
10E6 300631          JNB  EditFg,outChkEdP
          ;   ---
10E9 E537            MOV A,KEYBUFF         ;load KeyBuff
10EB B40017          CJNE A,#UpKy,DwworkP
10EE 0530            INC  KpCount          ;SetPoint is buffer,DSP is real
10F0 E530            MOV  A,KPCount
10F2 7A6C            MOV  R2,#KMaxCon
10F4 0A              INC  R2               ;KMaxCon+1->R2
10F5 B50203          CJNE A,R2,CtUpMaxP
10F8 753000          MOV  KpCount,#00H
10FB E530  ctupMaxP: MOV  A,KPcount
10FD 120C60          LCALL LookatKtabel    ;K0->R0,K1->R1,R3-
>88,R4->8
1100 120C7E          LCALL ShowKtab
1103 8015            SJMP outChkEDP
          ;   ---
1105 B40112  DwWorkP: CJNE A,#DwKy ,outChkEDP
1108 1530            DEC  KPcount
110A E530            MOV  A,KPcount
110C B4FF03          CJNE A,#0FFH,CtDwMaxP
110F 75306C          MOV  KPcount,#KMaxCon
1112 E530  CtDwMaxP: MOV  A,KPcount
1114 120C60          LCALL LookatKtabel    ;K0->R0,K1->R1,R3-
>88,R4->8
1117 120C7E          LCALL ShowKtab
111A 00    outChkEDP: NOP
111B D004            POP R4
111D D003            POP R3
111F D002            POP R2
1121 D001            POP R1
1123 D000            POP R0
1125 22              RET

1126      Klst:
          ;-----------
          ; Subroutine IE0_2
          ; SetPoint
          ;-----------
1126 C000            PUSH  R0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
1128 C001          PUSH  R1
112A C002          PUSH  R2
112C C003          PUSH  R3
112E C004          PUSH  R4
1130 7457          MOV   A,#01010111B      ;"K"
1132 90A000          MOV   DPTR ,#DISP1A
1135 F0            MOVX  @DPTR,A
1136 7408          MOV   A,#00001000B      ;"T"
1138 90A001          MOV   DPTR ,#DISP1B
113B F0            MOVX  @DPTR,A
113C 120FAA          LCALL  CurOnOff
113F E532          MOV   A,KIcount
1141 120C60          LCALL  LookatKtabel     ;K0->R0,K1->R1,R3-
>88,R4->8
1144 120C7E          LCALL  ShowKtab
1147 120FB5          LCALL  ChkEdit
114A 300631          JNB  EditFg ,outChkEDI
        ;   ---
114D E537          MOV A,KEYBUFF          ;load KeyBuff
114F B40017          CJNE  A,#UpKy,DwworkI
1152 0532          INC  KICount          ;SetPoint is buffer,DSP is real
1154 E532          MOV  A,KICount
1156 7A6C          MOV  R2,#KMaxCon
1158 0A            INC  R2               ;KMaxCon+1->R2
1159 B50203          CJNE  A,R2,CtupMaxI
115C 753200          MOV  KICount,#00H
115F E532     ctupMaxI: MOV   A,KIcount
1161 120C60          LCALL  LookatKtabel     ;K0->R0,K1->R1,R3-
>88,R4->8
1164 120C7E          LCALL  ShowKtab
1167 8015          SJMP  outChkEDI
        ;   ---
1169 B40112   DwWorkI:  CJNE  A,#DwKy ,outChkEDI
116C 1532          DEC  KIcount
116E E532          MOV  A,KIcount
1170 B4FF03          CJNE  A,#0FFH,CtDwMaxI
1173 75326C          MOV  KIcount,#KmaXcon
1176 E532     CtDwMaxI: MOV   A,KIcount
1178 120C60          LCALL  LookatKtabel     ;K0->R0,K1->R1,R3-
>88,R4->8
117B 120C7E          LCALL  ShowKtab
117E 00       outChkEDI: NOP
117F D004          POP  R4
1181 D003          POP  R3
1183 D002          POP  R2
1185 D001          POP  R1
1187 D000          POP  R0
1189 22            RET

118A        KDst:
        ;-----------
        ; Subroutine IE0_2
        ; SetPoint
        ;-----------
118A C000          PUSH  R0
118C C001          PUSH  R1
118E C002          PUSH  R2
1190 C003          PUSH  R3
1192 C004          PUSH  R4
1194 7457          MOV   A,#01010111B      ;"K"
1196 90A000          MOV        ; R1,R0 - R6 = R1,R0    R6:not distroys
1199 F0            MOVX  @DPTR,A
119A 745E          MOV   A,#01011110B      ;"D"
119C 90A001          MOV   DPTR ,#DISP1B
119F F0            MOVX  @DPTR,A
11A0 120FAA          LCALL  CurOnOff
11A3 E531          MOV   A,KDcount
11A5 120C60          LCALL  LookatKtabel     ;K0->R0,K1->R1,R3-
>88,R4->8
11A8 120C7E          LCALL  ShowKtab
11AB 120FB5          LCALL  ChkEdit
11AE 300631          JNB  EditFg,outChkEDD
        ;   ---
11B1 E537          MOV   A,KEYBUFF          ;load KeyBuff
11B3 B40017          CJNE  A,#UpKy,DwworkD
11B6 0531          INC  KDCount          ;SetPoint is buffer,DSP is real
11B8 E531          MOV  A,KDCount
11BA 7A6C          MOV  R2,#KMaxCon
11BC 0A            INC  R2               ;KMaxCon+1->R2
11BD B50203          CJNE  A,R2,CtUpMaxD
11C0 753100          MOV  KDCount,#00H
11C3 E531     ctupMaxD: MOV   A,KDcount
11C5 120C60          LCALL  LookatKtabel     ;K0->R0,K1->R1,R3-
>88,R4->8
11C8 120C7E          LCALL  ShowKtab
11CB 8015          SJMP  outChkEDD
        ;   ---
11CD B40112   DwWorkD:  CJNE  A,#DwKy ,outChkEDD
11D0 1531          DEC  KDcount
11D2 E531          MOV  A,KDCount
11D4 B4FF03          CJNE  A,#0FFH,CtDwMaxD
11D7 75316C          MOV  KDcount,#KmaxCon
11DA E531     CtDwMaxD: MOV   A,KDcount
11DC 120C60          LCALL  LookatKtabel     ;K0->R0,K1->R1,R3-
>88,R4->8
11DF 120C7E          LCALL  ShowKtab
11E2 00       outChkEDD: NOP
11E3 D004          POP  R4
11E5 D003          POP  R3
11E7 D002          POP  R2
11E9 D001          POP  R1
11EB D000          POP  R0
11ED 22            RET

11EE        ENst:
        ;-----------
        ; load buff to Variable
        ;
        ;-----------
11EE C000          PUSH  R0
11F0 C001          PUSH  R1
11F2 85393F          MOV  DSP,SETPOINT  ;update SetPoint
        ;-
11F5 E530          MOV  A,KPcount
11F7 120C60          LCALL  LookatKtabel
11FA 8842          MOV  KP0,R0
11FC 8943          MOV  KP1,R1       ;update Kp
        ;-
11FE E532          MOV  A,KIcount
1200 120C60          LCALL  LookAtKtabel
1203 8845          MOV  KI0,R0
1205 8946          MOV  KI1,R1       ;update KI
        ;-
1207 E531          MOV  A,KDcount
1209 120C60          LCALL  LookatKtabel
120C 884D          MOV  KD0,R0
120E 894E          MOV  KD1,R1       ;update Kd
        ;-
1210 852F3B          MOV  ADDRESS,ADcount ;update AD
        ;-
1213 301905          JNB  BaudSetFg,OutBaudEN     ;initialized new serial
1216 12055C          LCALL  InitSM
1219 C219          CLR  BaudSetFg
121B 00       OutBaudEN:  NOP
        ;-
121C 301A0F          JNB  SPFg,OutSPEN
121F 754800          MOV SUM0,#00H
1222 754900          MOV SUM1,#00H
        ;          MOV DC0,#00H
        ;          MOV DV0,#00H
1225 752D00          MOV  PIDdelay,#00H
1228 C21B          CLR SUMFg
122A C21C          CLR BKSUMFg
122C C21A          CLR  SPFg
122E 00       OutSPEN:   NOP
        ;-
122F C204          CLR  CURBlinkFg
1231 753A00          MOV  Status,#00H
1234 853334          MOV  MDwork,MDcount
        ;-----------
1237 E533          MOV  A,MDcount
1239 B40300          CJNE A,#03H,EndPLLSet
123C 1219E8   EndPLLSet:  LCALL  SETP_PLL     ;Put SetPoint in
PLL_con
        ;-----------
123F C208          CLR  PVFg
1241 7427          MOV  A,#00100111b
1243 90A000          MOV  DPTR,#DISP1A
1246 F0            MOVX @DPTR,A
1247 743C          MOV  A,#00111100B
1249 90A001          MOV  DPTR,#DISP1B
124C F0            MOVX @DPTR,A
124D 7440          MOV  A,#01000000B
124F 90A002          MOV  DPTR,#DISP1C
1252 F0            MOVX @DPTR,A
1253 D001          POP  R1
1255 D000          POP  R0
1257 22            RET

1258        MDst:
        ;-----------
        ; Select mode
        ;-----------
1258 C000          PUSH  R0
125A C001          PUSH  R1
125C C003          PUSH  R3
125E C004          PUSH  R4
1260 744A          MOV   A,#01001010B      ;"M"
1262 90A000          MOV   DPTR ,#DISP1A
1265 F0            MOVX  @DPTR,A
1266 745E          MOV   A,#01011110B      ;"D"
1268 90A001          MOV   DPTR ,#DISP1B
126B F0            MOVX  @DPTR,A
126C 120FAA          LCALL  CurOnOff
126F E533          MOV   A,MdCount
1271 51A9          ACALL  MDcountDisp      ;show "P,PI,PID,PLL"
1273 120FB5          LCALL  ChkEdit
1276 300626          JNB  EditFg,outChkEDM
        ;   ---
1279 E537          MOV A,KEYBUFF          ;load KeyBuff
127B B40010          CJNE  A,#UpKy,DwworkM
127E 0533          INC  MDCount          ;SetPoint is buffer,DSP is real
1280 E533          MOV  A,MDCount
1282 B40403          CJNE  A,#4D,ctupmaxM
1285 753300          MOV  MDCount,#00H
1288 E533     ctupMaxM:  MOV   A,MDcount
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
128A 51A9          ACALL MDCountDisp
128C 8011          SJMP  outChkEDM
        ;-----
128E B4010E  DwWorkM:  CJNE  A,#DwKy ,outChkEDM
1291 1533          DEC   MDcount
1293 E533          MOV   A,MDcount
1295 B4FF03        CJNE  A,#0FFH,ctDwMaxM
1298 753303        MOV   MDcount,#3D
129B E533    CtDwMaxM: MOV   A,MDcount
129D 51A9          ACALL MDCountDisp
        ;-----
129F 00      outChkEDM: NOP
12A0 D004          POP R4
12A2 D003          POP R3
12A4 D001          POP R1
12A6 D000          POP R0
12A8 22            RET

12A9        MDcountDisp:
        ;-------------------
        ; A:count
        ; 00->P_cont
        ; 01->PI_cont
        ; 02->PID_cont
        ; 03->PLL_cont
        ;-------------------
12A9 B40014        CJNE  A,#00H,MCD1
12AC 7400          MOV   A,#00000000B     ;" "
12AE 90B000        MOV   DPTR ,#DISP2A
12B1 F0            MOVX  @DPTR,A
12B2 7400          MOV   A,#00000000B     ;" "
12B4 90B001        MOV   DPTR ,#DISP2B
12B7 F0            MOVX  @DPTR,A
12B8 7473          MOV   A,#01110011B     ;"P"
12BA 90B002        MOV   DPTR ,#DISP2C
12BD F0            MOVX  @DPTR,A
12BE 8043          SJMP  OutMCD
        ;-----
12C0 B40114  MCD1:     CJNE A,#01H,MCD2
12C3 7400          MOV   A,#00000000B     ;" "
12C5 90B000        MOV   DPTR ,#DISP2A
12C8 F0            MOVX  @DPTR,A
12C9 7473          MOV   A,#01110011B     ;"P"
12CB 90B001        MOV   DPTR ,#DISP2B
12CE F0            MOVX  @DPTR,A
12CF 7408          MOV   A,#00001000B     ;"i"
12D1 90B002        MOV   DPTR ,#DISP2C
12D4 F0            MOVX  @DPTR,A
12D5 802C          SJMP  OutMCD
        ;-----
12D7 B40214  MCD2:     CJNE A,#02H,MCD3
12DA 7473          MOV   A,#01110011B     ;"P"
12DC 90B000        MOV   DPTR ,#DISP2A
12DF F0            MOVX  @DPTR,A
12E0 7408          MOV   A,#00001000B     ;"i"
12E2 90B001        MOV   DPTR ,#DISP2B
12E5 F0            MOVX  @DPTR,A
12E6 745E          MOV   A,#01011110B     ;"d"
12E8 90B002        MOV   DPTR ,#DISP2C
12EB F0            MOVX  @DPTR,A
12EC 8015          SJMP  OutMCD
        ;-----
12EE B40312  MCD3:     CJNE A,#03H,OutMCD
12F1 7473          MOV   A,#01110011B     ;"P"
12F3 90B000        MOV   DPTR ,#DISP2A
12F6 F0            MOVX  @DPTR,A
12F7 7407          MOV   A,#00000111B     ;"L"
12F9 90B001        MOV   DPTR ,#DISP2B
12FC F0            MOVX  @DPTR,A
12FD 7407          MOV   A,#00000111B     ;"L"
12FF 90B002        MOV   DPTR ,#DISP2C
1302 F0            MOVX  @DPTR,A
        ;-----
1303 00      OutMCD:   NOP
1304 22            RET

        ;-----
        ; other
        ;-----

1305        showAD:
        ;-------------------------------
        ; routine show status of address for KEY
        ;-------------------------------
1305 7400          MOV   A,#00000000B     ;"0"
1307 90B000        MOV   DPTR ,#DISP2A
130A F0            MOVX  @DPTR,A
130B 7400          MOV   A,#00000000B     ;"0"
130D 90B001        MOV   DPTR ,#DISP2B
1310 F0            MOVX  @DPTR,A
        ;---
1311 E52F          MOV A,ADCount
1313 B40009        CJNE A,#00,AD01
1316 7403          MOV   A,#00000011B     ;"1"
1318 90B002        MOV   DPTR ,#DISP2C
131B F0            MOVX  @DPTR,A
131C 0213AB        LJMP OutADD            ;
        ;-
131F B40109  AD01:     CJNE A,#01D,AD02
1322 7476          MOV   A,#01110110B     ;"2"
1324 90B002        MOV   DPTR ,#DISP2C
1327 F0            MOVX  @DPTR,A
1328 0213AB        LJMP  OutADD
        ;-
132B B40209  AD02:     CJNE A,#02D,AD03
132E 747C          MOV   A,#01111100B     ;"3"
1330 90B002        MOV   DPTR ,#DISP2C
1333 F0            MOVX  @DPTR,A
1334 0213AB        LJMP  OutADD
        ;-
1337 B40309  AD03:     CJNE A,#03D,AD04
133A 7459          MOV   A,#01011001B     ;"4"
133C 90B002        MOV   DPTR ,#DISP2C
133F F0            MOVX  @DPTR,A
1340 0213AB        LJMP  OutADD
        ;-
1343 B40408  AD04:     CJNE A,#04D,AD05
1346 746D          MOV   A,#01101101B     ;"5"
1348 90B002        MOV   DPTR ,#DISP2C
134B F0            MOVX  @DPTR,A
134C 805D          SJMP  OutADD
        ;-
134E B40508  AD05:     CJNE A,#05D,AD06
1351 746F          MOV   A,#01101111B     ;"6"
1353 90B002        MOV   DPTR ,#DISP2C
1356 F0            MOVX  @DPTR,A
1357 8052          SJMP  OutADD
        ;-
1359 B40608  AD06:     CJNE A,#06D,AD07
135C 7438          MOV   A,#00111000B     ;"7"
135E 90B002        MOV   DPTR ,#DISP2C
1361 F0            MOVX  @DPTR,A
1362 8047          SJMP  OutADD
        ;-
1364 B40708  AD07:     CJNE A,#07D,AD08
1367 747F          MOV   A,#01111111B     ;"8"
1369 90B002        MOV   DPTR ,#DISP2C
136C F0            MOVX  @DPTR,A
136D 803C          SJMP  OutADD
        ;-
136F B40808  AD08:     CJNE A,#08D,AD09
1372 747D          MOV   A,#01111101B     ;"9"
1374 90B002        MOV   DPTR ,#DISP2C
1377 F0            MOVX  @DPTR,A
1378 8031          SJMP  OutADD
        ;-
137A B4090E  AD09:     CJNE A,#09D,AD10
137D 7403          MOV   A,#00000011B     ;"1"
137F 90B001        MOV   DPTR ,#DISP2B
1382 F0            MOVX  @DPTR,A
1383 743F          MOV   A,#00111111B     ;" 0"
1385 90B002        MOV   DPTR ,#DISP2C
1388 F0            MOVX  @DPTR,A
1389 8020          SJMP  OutADD
        ;-
138B B40A0E  AD10:     CJNE A,#010D,AD11
138E 7403          MOV   A,#00000011B     ;"1"
1390 90B001        MOV   DPTR ,#DISP2B
1393 F0            MOVX  @DPTR,A
1394 7403          MOV   A,#00000011B     ;"1"
1396 90B002        MOV   DPTR ,#DISP2C
1399 F0            MOVX  @DPTR,A
139A 800F          SJMP  OutADD
        ;-
139C B40B0C  AD11:     CJNE A,#011D,OutADD
139F 7403          MOV   A,#00000011B     ;"1"
13A1 90B001        MOV   DPTR ,#DISP2B
13A4 F0            MOVX  @DPTR,A
13A5 7476          MOV   A,#01110110B     ;"2"
13A7 90B002        MOV   DPTR ,#DISP2C
13AA F0            MOVX  @DPTR,A
        ;-
13AB 00      OutADD:   NOP
13AC 22            RET

13AD        showBD:
        ;-------------------
        ; show status BAUD
        ;-------------------
13AD E52E          MOV A,BDCount
13AF B40018        CJNE A,#00,BD01
13B2 7403          MOV   A,#00000011B     ;"1"
13B4 90B000        MOV   DPTR ,#DISP2A
13B7 F0            MOVX  @DPTR,A
13B8 746D          MOV   A,#01101101B     ;"5"
13BA 90B001        MOV   DPTR ,#DISP2B
13BD F0            MOVX  @DPTR,A
13BE 743F          MOV   A,#00111111B     ;"0"
13C0 90B002        MOV   DPTR ,#DISP2C
13C3 F0            MOVX  @DPTR,A
13C4 753D40        MOV   BAUDRATE,#BR150
13C7 021465        LJMP OutBD             ;
        ;-                    0000,0111B
13CA B40118  BD01:     CJNE A,#01,BD02

13CD 747C          MOV   A,#01111100B     ;"3"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
13CF 90B000          MOV   DPTR ,#DISP2A
13D2 F0             MOVX  @DPTR,A
13D3 743F            MOV   A,#00111111B     ;"0"
13D5 90B001          MOV   DPTR ,#DISP2B
13D8 F0             MOVX  @DPTR,A
13D9 743F            MOV   A,#00111111B     ;"0"
13DB 90B002          MOV   DPTR ,#DISP2C
13DE F0             MOVX  @DPTR,A
13DF 753DA0          MOV   BAUDRATE,#BR300
13E2 021465          LJMP  OutBD
                    ;-
13E5 B40217  BD02:   CJNE A,#02,BD03
13E8 746F            MOV   A,#01101111B     ;"6"
13EA 90B000          MOV   DPTR ,#DISP2A
13ED F0             MOVX  @DPTR,A
13EE 743F            MOV   A,#00111111B     ;"0"
13F0 90B001          MOV   DPTR ,#DISP2B
13F3 F0             MOVX  @DPTR,A
13F4 743F            MOV   A,#00111111B     ;"0"
13F6 90B002          MOV   DPTR ,#DISP2C
13F9 F0             MOVX  @DPTR,A
13FA 753DD0          MOV   BAUDRATE,#BR600
13FD 8066           SJMP  OutBD
                    ;-
13FF B40317  BD03:   CJNE A,#03,BD04

1402 7483            MOV   A,#10000011B     ;"1."
1404 90B000          MOV   DPTR ,#DISP2A
1407 F0             MOVX  @DPTR,A
1408 7476            MOV   A,#01110110B     ;"2"
140A 90B001          MOV   DPTR ,#DISP2B
140D F0             MOVX  @DPTR,A
140E 7457            MOV   A,#01010111B     ;"K"
1410 90B002          MOV   DPTR ,#DISP2C
1413 F0             MOVX  @DPTR,A
1414 753DE8          MOV   BAUDRATE,#BR1200
1417 804C           SJMP  OutBD
                    ;-
1419 B40417  BD04:   CJNE A,#04,BD05
141C 74F6            MOV   A,#11110110B     ;"2."
141E 90B000          MOV   DPTR ,#DISP2A
1421 F0             MOVX  @DPTR,A
1422 7459            MOV   A,#01011001B     ;"4"
1424 90B001          MOV   DPTR ,#DISP2B
1427 F0             MOVX  @DPTR,A
1428 7457            MOV   A,#01010111B     ;"K"
142A 90B002          MOV   DPTR ,#DISP2C
142D F0             MOVX  @DPTR,A
142E 753DF4          MOV   BAUDRATE,#BR2400
1431 8032           SJMP  OutBD
                    ;-
1433 B40517  BD05:   CJNE A,#05,BD06

1436 74D9            MOV   A,#11011001B     ;"4."
1438 90B000          MOV   DPTR ,#DISP2A
143B F0             MOVX  @DPTR,A
143C 747F            MOV   A,#01111111B     ;"8"
143E 90B001          MOV   DPTR ,#DISP2B
1441 F0             MOVX  @DPTR,A
1442 7457            MOV   A,#01010111B     ;"K"
1444 90B002          MOV   DPTR ,#DISP2C
1447 F0             MOVX  @DPTR,A
1448 753DFA          MOV   BAUDRATE,#BR4800
144B 8018           SJMP  OutBD
                    ;-
144D B40615  BD06:   CJNE A,#06,OutBD
1450 74FD            MOV   A,#11111101B     ;"9."
1452 90B000          MOV   DPTR ,#DISP2A
1455 F0             MOVX  @DPTR,A
1456 746F            MOV   A,#01101111B     ;"6"
1458 90B001          MOV   DPTR ,#DISP2B
145B F0             MOVX  @DPTR,A
145C 7457            MOV   A,#01010111B     ;"K"
145E 90B002          MOV   DPTR ,#DISP2C
1461 753DFD          MOV   BAUDRATE,#BR9600
1464 F0             MOVX  @DPTR,A
                    ;-
1465 00      OutBD:  NOP
1466 22              RET
        ;-------------
        ; end routine
        ;-------------

1467            INCL  "UsePLLz1.asm"

        ;-------------------------------
        ; all of routine of algorithm of controller
        ;-------------------------------

        ;  DSP:   EQU   3FH  ;working setpoint
        ;  DCP:   EQU   40H  ;Controller  output
        ;  DC0:   EQU   41H  ;Old Adbuff
        ;  KP0:   EQU   42H  ;Proprotional gain LSB
        ;  KP1:   EQU   43H  ;Proprotional gain MSB
        ;  DC1:   EQU   44H
        ;  KI0:   EQU   45H
        ;  KI1:   EQU   46H
        ;  DT:    EQU   47H
        ;  SUM0:  EQU   48H
        ;  SUM1:  EQU   49H
        ;  DCD:   EQU   4AH
        ;  DV0:   EQU   4BH
        ;  DE:    EQU   4CH
        ;  KD0:   EQU   4DH
        ;  KD1:   EQU   4EH
        ;  DCPI:  EQU   4FH
        ;  DV   =  ADBuff
        ;  DSP  =  SetPoint
        ;  DE   =  R3
        ;  SGNFg:  EQU   00H
        ;  SGNDFg: EQU   01H
        ;  SGNIFg: EQU   03H
        ;  N0_Digit: EQU   08H
        ;  N0_DTDig: EQU   04H

1467        Init_Control:
            ;-------------
            ; Initial value for P_Controller ,PI_controller ,PID_controller
            ;-------------
1467 753F80          MOV DSP,#80H
146A 754201          MOV KP0,#01H
146D 754300          MOV KP1,#00H
1470 754500          MOV KI0,#00H
1473 754600          MOV KI1,#00H
1476 754D00          MOV KD0,#00H
1479 754E00          MOV KD1,#00H
147C 754800          MOV SUM0,#00H
147F 754900          MOV SUM1,#00H
1482 22              RET

1483        P_contl:
            ;-------------
            ;DE:control variable error
            ;CORR  : R3
            ;-------------
1483 C000            PUSH R0
1485 C001            PUSH R1
1487 C002            PUSH R2
1489 C003            PUSH R3
148B C005            PUSH R5
148D C006            PUSH R6
148F C007            PUSH R7
1491 C200            CLR  SGNFg
1493 C3              CLR  C
1494 E535            MOV  A,ADBUFF
1496 953F            SUBB A,DSP
1498 5005            JNC  NoC_P       ;error is carry as sign = 1
149A D200              SETB SGNFg     ;Sgn 0 =Plus,Sgn 1=Minus
149C F4                CPL  A         ;complement
149D 3400              ADDC A,#00H
149F F54C    NoC_P:    MOV  DE,A
14A1 AD42              MOV  R5,KP0    ;define var
14A3 AE43              MOV  R6,KP1    ;0000 0000 . 0000 0000
14A5 E54C              MOV  A,DE      ;KP*DE->CORR
14A7 120211            LCALL MULT16C  ;out R2,R1,R0
14AA 7F08              MOV  R7,#N0_digit
14AC 121989            LCALL ShiftDig
14AF 12195A            LCALL OverLimit1
14B2 AB00              MOV  R3,R0     ;R0->DORR
14B4 20000B  OutLimitK:  JB  SGNFg, minusP0
14B7 C3                CLR  C
14B8 E53F              MOV  A,DSP     ;load Zero Error DSP-CORR-
>DCP
14BA 9B                SUBB A,R3
14BB FB                MOV  R3,A
14BC 5005              JNC  NoaddP0
14BE 7B00    maxP:       MOV  R3,#00H
14C0 8001            SJMP NoAddP0
14C2 C3      minusP0:    CLR  C        ;SGNFg = 0 work minus
             ;           MOV  A,R3
             ;           SUBB A,DC0     ;CORR-DC0 ->DCP
             ;           MOV  R3,A
             ;           JNC  NoAddP0
             ;minP:      MOV  R3,#00H
14C3 8B40    NoaddP0:    MOV  DCP,R3    ;DCP
14C5 853541              MOV  DC0,ADBuff
14C8 D007                POP  R7
14CA D006                POP  R6
14CC D005                POP  R5
14CE D003                POP  R3
14D0 D002                POP  R2
14D2 D001                POP  R1
14D4 D000                POP  R0
14D6 22                  RET

14D7        I_contl:
            ;-------------
            ; DE :DE
            ;-------------
14D7 C000            PUSH  R0
14D9 C001            PUSH  R1
14DB C002            PUSH  R2
14DD C003            PUSH  R3
14DF C004            PUSH  R4
14E1 C005            PUSH  R5
14E3 C006            PUSH  R6
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
14E5 C007            PUSH R7
14E7 C200            CLR SGNFg
14E9 C3              CLR C
14EA E535            MOV A,ADBUFF
14EC 953F            SUBB A,DSP
14EE 5005            JNC NoC_I       ;error is carry sign 1
14F0 D200              SETB SGNFg
14F2 F4              CPL A       ;complement
14F3 3400            ADDC A,#00H
14F5 F54C    NoC_I:    MOV DE,A
14F7 30030D          JNB SGNIFg, Slopdw
14FA 300005            JNB SGNFg ,minusUp
14FD 1217FA              LCALL ADDSUMUP      ;plusup
1500 8010              SJMP  OutSlop
1502 121830  minusUp:       LCALL  SubSumC
1505 800B              SJMP  outSlop
1507 200005  SlopDw:       JB SGNFg , minusDw
150A 1217FA              LCALL  ADDSUMUP
150D 8003              SJMP  outSlop
150F 121830  minusDw:       LCALL  SubSumC
1512 00      outSlop:   NOP
1513 AD48            MOV R5,SUM0     ;multiplicand
1515 AE49            MOV R6,SUM1     ;multiplicand
1517 E545            MOV A, KI0      ;multiplier
1519 AC46            MOV R4,KI1
151B 1202B0            LCALL MULT16_16C   ;KI*SUM
151E 7F08              MOV R7,#N0_Digit
1520 121989            LCALL ShiftDig
1523 12196C          LCALL OverLimit2
1526 AD00            MOV R5,R0
1528 AE01            MOV R6,R1
152A E547            MOV A,DT
152C 120211            LCALL MULT16C   ;KI*SUM*DT
152F 7F0A              MOV R7,#N0_DtDig
1531 121989            LCALL ShiftDig
1534 12195A          LCALL OverLimit1
1537 8844            MOV DCI ,R0
1539 D007            POP R7
153B D006            POP R6
153D D005            POP R5
153F D004            POP R4
1541 D003            POP R3
1543 D002            POP R2
1545 D001            POP R1
1547 D000            POP R0
1549 22              RET


154A       D_Cont:
           ;--------
           ; R3 : DDE
           ; R4 : adbuff
           ;--------
154A C000            PUSH R0
154C C001            PUSH R1
154E C002            PUSH R2
1550 C003            PUSH R3
1552 C004            PUSH R4
1554 C005            PUSH R5
1556 C006            PUSH R6
1558 C007            PUSH R7
155A C3              CLR C
155B C200            CLR SGNFg
155D AC35            MOV R4,ADBUFF     ;DV-DSP->DE
155F EC              MOV A,R4
1560 953F            SUBB A,DSP
1562 8B4C            MOV DE,R3
1564 5007            JNC NoC_D
1566 D200            SETB SGNFg      ;Sgn 0 =Plus,Sgn 1=Minus
1568 F4              CPL A       ;complement
1569 3400            ADDC A,#00H
156B 8B4C            MOV DE,R3       ;get DE
156D C3      NoC_D:    CLR C
156E C201            CLR SGNDFg
1570 EC              MOV A,R4
1571 9541            SUBB A,DC0
1573 FB              MOV R3,A        ;DDE-> R3
1574 5007            JNC NoCyD
1576 D201              SETB SGNDFg
1578 EB              MOV A,R3
1579 F4              CPL A
157A 3400              ADDC A,#00H
157C FB              MOV R3,A
157D 8C41    NoCyD:    MOV DC0,R4
157F AD4D            MOV R5,KD0
1581 AE4E            MOV R6,KD1
1583 EB              MOV A,R3        ;KD*DDE
1584 120211            LCALL MULT16C     ;product R2,R1,R0
1587 12196C            LCALL OVERLIMIT2
158A 00      OutChkD:   NOP
158B B80008            CJNE R0,#00H,toDt
158E B90005            CJNE R1,#00H,toDt
1591 BA0002            CJNE R2,#00H,toDt
1594 8005            SJMP nonDt
1596 AF47    toDt:     MOV R7,DT       ;KD*DDE/DT
1598 120344            LCALL DIV16     ;product R1,R0
159B C3      nonDt:      CLR C
159C 7408              MOV A,#N0_Digit
159E 940A              SUBB A,#N0_DTDig    ;
15A0 FF              MOV R7,A
15A1 121989            LCALL ShiftDIG
15A4 12195A          LCALL OverLimit1
15A7 8B4A    OutMore:   MOV DCD,R0
15A9 D007            POP R7
15AB D006            POP R6
15AD D005            POP R5
15AF D004            POP R4
15B1 D003            POP R3
15B3 D002            POP R2
15B5 D001            POP R1
15B7 D000            POP R0
15B9 22              RET

15BA       PI_cont:
           ;--------
           ; DE :DE
           ;--------
15BA C000            PUSH R0
15BC C001            PUSH R1
15BE C002            PUSH R2
15C0 C003            PUSH R3
15C2 C004            PUSH R4
15C4 C005            PUSH R5
15C6 C006            PUSH R6
15C8 C007            PUSH R7
15CA C200            CLR SGNFg
15CC C3              CLR C
15CD E535            MOV A,ADBUFF
15CF 953F            SUBB A,DSP
15D1 5005            JNC NoC_PI      ;DE is carry sign 1
15D3 D200              SETB SGNFg
15D5 F4              CPL A       ;complement
15D6 3400              ADDC A,#00H
15D8 F54C    NoC_PI:   MOV DE,A
15DA 30030D          JNB SGNIFg, SlopIdw
15DD 300005            JNB SGNFg ,minusIUp
15E0 1217FA              LCALL ADDSUMUP      ;plusup
15E3 8010              SJMP  OutSlopI
15E5 121830  minusIUp:       LCALL  SubSumC
15E8 800B              SJMP  outSlopI
15EA 200005  SlopIDw:       JB SGNFg , minusIDw
15ED 1217FA              LCALL  ADDSUMUP
15F0 8003              SJMP  outSlopI
15F2 121830  minusIDw:       LCALL  SubSumC
15F5 00      outSlopI:  NOP
15F6 AD48            MOV R5,SUM0     ;multiplicand
15F8 AE49            MOV R6,SUM1     ;multiplicand
15FA E545            MOV A, KI0      ;multiplier
15FC AC46            MOV R4,KI1
15FE 1202ED            LCALL MULT16_16C2   ;KI*SUM ->
R3,R2,R1,R0
1601 7F08              MOV R7,#N0_Digit
1603 301B01            JNB  SUMFg,ReduceDig
1606 1F              DEC R7        ;Dec for Over SUM
1607 121999  ReduceDig:     LCALL ShiftDig2
160A 121978            LCALL OverLimit2_1
160D AD00            MOV R5,R0
160F AE01            MOV R6,R1
1611 E547            MOV A,DT
1613 120258            LCALL MULT16C2      ;KI*SUM*DT ->
R3,R2,R1,R0
1616 7F0A              MOV R7,#N0_DtDig
1618 121999            LCALL ShiftDig2
161B 121978            LCALL OverLimit2_1
161E AD00            MOV R5,R0
1620 AE01            MOV R6,R1
1622 E542            MOV A,KP0
1624 AC43            MOV R4,KP1
1626 1202B0            LCALL  MULT16_16C
1629 7F08              MOV R7,#N0_Digit
162B 121989              LCALL ShiftDig    ;KI*SUM*DT*KP
162E 12195A            LCALL OverLimit1
1631 8844            MOV DCI ,R0       ;end of I_cont
           ;-
1633 AD42            MOV R5,KP0
1635 AE43            MOV R6,KP1
1637 E54C            MOV A,DE
1639 120211            LCALL MULT16C
163C 7F08              MOV R7,#N0_Digit
163E 121989              LCALL ShiftDig
1641 12195A            LCALL OverLimit1
1644 8840            MOV DCP,R0        ;keep  P_cont
1646 30030D            JNB SGNIFg ,slopPIdw
1649 300005              JNB SGNFg ,minusPIup
164C 121874              LCALL PcPiPp      ;DC0+DCI+DCP->DCPI
164F 8010              SJMP OutSlopPI
1651 1218A0  minusPIUp:       LCALL PcPiMp      ;DC0+DCI-DCP-
>DCPI
1654 800B              SJMP  OutSlopPI
1656 300005  slopPIdw:       JNB  SGNFg,minusPIdw
1659 1218CE              LCALL PcPpMi      ;DC0+DCP-DCI->DCPI
165C 8003              SJMP  OutSlopPI
165E 1218FC  minusPIdw:       LCALL PcMpMi      ;DC0-DCP-DCI-
>DCPI
1661 00      outSlopPI:  NOP
1662 853541            MOV DC0,ADBuff
```

ลิขสิทธิ์เป็นของจุฬาลงกรณ์มหาวิทยาลัย สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
1665 D007            POP R7
1667 D006            POP R6
1669 D005            POP R5
166B D004            POP R4
166D D003            POP R3
166F D002            POP R2
1671 D001            POP R1
1673 D000            POP R0
1675 22             RET

1676      PID_cont1:
          ;-------------------
          ; DE :DE
          ;-------------------
1676 C000            PUSH R0
1678 C001            PUSH R1
167A C002            PUSH R2
167C C003            PUSH R3
167E C004            PUSH R4
1680 C005            PUSH R5
1682 C006            PUSH R6
1684 C007            PUSH R7
1686 C0F0            PUSH B          ;represent DDE
1688 C200            CLR SGNFg
168A C3             CLR C
168B E535            MOV A,ADBUFF      ;Save Previously ADBUFF
168D 953F            SUBB A,DSP        ; DE = DV - DSP if carry set
SGNFg
168F 5005            JNC NoC_PID       ;error is carry sign 1
1691 D200             SETB SGNFg
1693 F4              CPL A          ;complement
1694 3400            ADDC A,#00H
1696 F54C     NoC_PID:     MOV DE,A
          ;-
1698 7007            JNZ ChkOldFg      ;if DV=0,DSP=0 ,give SGNFg =
1
169A E53F            MOV A,DSP
169C B40002          CJNE A,#00h,ChkOldFg
169F D200            SETB SGNFg
          ;-
16A1 300005   ChkOldFg:   JNB SGNFg,QuestEQ   ;find
SGNFg=OldSGNFg? if true then do finddif:
16A4 300A1B          JNB OldSGNFg,ForceDC   ;
16A7 8003            SJMP finddif
16A9 200A16   QuestEQ:    JB  OldSGNFg,ForceDC
          ;-
16AC C3       finddif:    CLR C          ;D_cont
16AD C201            CLR SGNDFg        ;find DDE = DE-DV0
16AF E54C            MOV A,DE          ;DV0 = previouly DE,
16B1 954B            SUBB A,DV0        ;
16B3 F5F0            MOV B,A           ;keep DDE-> B
16B5 5009            JNC NoCyPID
16B7 D201             SETB SGNDFg
16B9 E5F0             MOV A,B
16BB F4              CPL A
16BC 3400             ADDC A,#00H
16BE F5F0             MOV B,A
16C0 800E    NoCyPID:    SJMP PIDbeginD
          ;-
16C2 C3      ForceDC:    CLR C
16C3 D201            SETB SGNDFg
16C5 E54C            MOV A,DE
16C7 254B            ADD A,DV0         ;DE+DV0
16C9 F5F0            MOV B,A           ;DDE
16CB 5003            JNC PIDbeginD
16CD 75F0FF           MOV B,#0FFH       ;DDE
          ;-
16D0 E52D    PIDbeginD:  MOV A,PIDdelay    ;make  width of plus for D
effection
16D2 701C            JNZ beginDr
16D4 ADF0            MOV R5,B          ;buffer of B for moment
16D6 E5F0            MOV A,B           ;load DDE to A
16D8 75F00F          MOV B,#0FH        ;set divider
16DB 84             DIV AB
16DC F52D            MOV PIDdelay,A    ;result for width of plus
16DE 8DF0            MOV B,R5          ;return value for value
          ;-
16E0 600E            JZ  beginDr       ;if isn't STEP SP jump
16E2 ADF0            MOV R5,B          ;buffer again repeat for
16E4 75F005          MOV B,#05H
16E7 E531            MOV A,KDCount
16E9 84             DIV AB
16EA 252D            ADD A,PIDdelay    ;compensate width of D
16EC F52D            MOV PIDdelay,A
16EE 8DF0            MOV B,R5          ;return value DDE
          ;-
16F0 AD4D    beginDr:    MOV R5,KD0
16F2 AE4E            MOV R6,KD1
16F4 E5F0            MOV A,B           ;KD*DDE
16F6 120211          LCALL MULT16C     ;product R2,R1,R0
16F9 12196C          LCALL OVERLIMIT2
          ;-
16FC AD42            MOV  R5,KP0
16FE AE43            MOV  R6,KP1
1700 E8              MOV  A,R0
1701 AC01            MOV  R4,R1
1703 1202B0          LCALL MULT16_16C  ;KD*DDE*KP->DCD
1706 B80008          CJNE R0,#00H,toPIDt

1709 B90005          CJNE R1,#00H,toPIDt
170C BA0002          CJNE R2,#00H,toPIDt
170F 8005           SJMP nonPIDt
1711 AF47    toPIDt:     MOV R7,DT      ;KD*DDE/DT
1713 12037A          LCALL DIV24       ;product R2,R1,R0
1716 C3      nonPIDt:    CLR C
1717 7408            MOV A,#N0_Digit
1719 2408            ADD A,#N0_Digit
171B 2407            ADD A,#07h        ;compensate
171D 940A            SUBB A,#N0_DTDig     ;
171F 5001            JNC Beginshift
1721 E4              CLR A             ;
1722 FF      Beginshift:    MOV R7,A
1723 121989          LCALL ShiftDIG
1726 12195A          LCALL OverLimit1
1729 8D4A            MOV DCD,R0        ;end D_cont, KD*DDE/DT ->
DCD
          ;-
172B 201C1B          JB  BKSUMFg,outSlopPD     ;if not PIDCount = 0
Jump
          ;-
172E 30030D          JNB SGNIFg, SlopPDw
1731 300005            JNB SGNFg ,minusPDUp
1734 1217FA              LCALL ADDSUMUP      ;plusup
1737 8010              SJMP  OutSlopPD
1739 121830   minusPDUp:       LCALL  SubSumC
173C 800B                SJMP   outSlopPD
173E 200005   SlopPDw:       JB  SGNFg , minusPDw
1741 1217FA              LCALL  ADDSUMUP
1744 8003              SJMP   outSlopPD
1746 121830   minusPDw:       LCALL  SubSumC
1749 00       outSlopPD:    NOP
          ;-
174A AD48            MOV R5,SUM0       ;multiplicand
174C AE49            MOV R6,SUM1       ;multiplicand
174E E545           MOV A, KI0        ;multiplier
1750 AC46           MOV R4,KI1
1752 1202ED          LCALL MULT16_16C2 ;KI*SUM -
>R3,R2,R1,R0
1755 7F08              MOV R7,#N0_Digit
1757 301B01            JNB  SUMFg,ReduceDigD
175A 1F              DEC R7         ;Dec for Over SUM
175B 121999   ReduceDigD:    LCALL ShiftDig2
175E 121978            LCALL OverLimit2_1
1761 AD00            MOV R5,R0
1763 AE01            MOV R6,R1
1765 E547           MOV A,DT
1767 120258          LCALL MULT16C2    ;KI*SUM*DT
176A 7F0A              MOV R7,#N0_DtDig
176C 121999            LCALL ShiftDig2
176F 121978            LCALL OverLimit2_1
1772 AD00            MOV R5,R0
1774 AE01            MOV R6,R1
1776 E542           MOV A,KP0
1778 AC43           MOV R4,KP1
177A 1202B0          LCALL MULT16_16C
177D 7F08              MOV R7,#N0_Digit
177F 121989            LCALL ShiftDig    ;KI*SUM*DT*KP -> DCI
1782 12195A          LCALL OverLimit1
1785 8844           MOV DCI,R0        ;end of I_cont
          ;-
1787 AD42            MOV R5,KP0
1789 AE43            MOV R6,KP1
178B E54C            MOV A,DE
178D 120211          LCALL MULT16C
1790 7F08              MOV R7,#N0_Digit
1792 121989            LCALL ShiftDig
1795 12195A          LCALL OverLimit1
1798 8840           MOV DCP,R0        ;keep P_cont
179A 30030D          JNB SGNIFg ,slopPIDdw
179D 300005            JNB SGNFg ,minusPIDup
17A0 121874            LCALL PcPiPp      ;DC0+DCI+DCP->DCPI
17A3 8010              SJMP OutSlopPID
17A5 1218A0   minusPIDUp:       LCALL PcPiMp      ;DC0+DCI-DCP-
>DCPI
17A8 800B              SJMP OutSlopPID
17AA 300005   slopPIDdw:       JNB  SGNFg,minusPIDdw
17AD 1218CE              LCALL PcPpMi      ;DC0+DCP-DCI->DCPI
17B0 8003              SJMP OutSlopPID
17B2 1218FC   minusPIDdw:       LCALL PcMpMi      ;DC0-DCP-DCI-
>DCPI
17B5 00       outSlopPID:  NOP
          ;-
17B6 30000D          JNB SGNFg,DVmoreDSP
17B9 200105          JB SGNDFg,difminus
17BC 1219AC            LCALL PlusDCD
17BF 8010            SJMP outDif
17C1 1219B9   difminus:   LCALL MinusDCD
17C4 800B              SJMP  outDif
          ;-
17C6 200105   DVmoreDSP:  JB  SGNDFg,difplusD
17C9 1219B9            LCALL MinusDCD
17CC 8003              SJMP  outdif
17CE 1219AC   difplusD:   LCALL PlusDCD
          ;-
17D1 00       outDif:     NOP
          ;-
17D2 E52D            MOV A,PIDdelay
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
17D4 6005           JZ  inUpdate
17D6 D21C           SETB BKSUMFg      ;for SUM inactive
17D8 D52D09         DJNZ PIDdelay,outUpdate
17DB 854C4B inUpdate:  MOV DV0,DE
17DE A200           MOV C,SGNFg
17E0 920A           MOV OldSGNFg,C
17E2 C21C           CLR BKSUMFg       ;for SUM active
                    ;-
17E4 853541 outUpdate:  MOV DC0,ADBuff
17E7 D0F0           POP B
17E9 D007           POP R7
17EB D006           POP R6
17ED D005           POP R5
17EF D004           POP R4
17F1 D003           POP R3
17F3 D002           POP R2
17F5 D001           POP R1
17F7 D000           POP R0
17F9 22             RET           ;end PID-control

17FA      AddsumUp:
          ;-----------
          ; subrutine of I_contl
          ; Sum +DE->Sum
          ; ->FFFFH
          ;-----------
17FA C000           PUSH R0
17FC C001           PUSH R1
17FE C006           PUSH R6
1800 C3             CLR C
1801 A848           MOV R0,SUM0
1803 A949           MOV R1,SUM1
1805 AE4C           MOV R6,DE
1807 120132         LCALL ADD16_8C
180A 8848           MOV SUM0,R0
180C 8949           MOV SUM1,R1
                    ;-
180E 5019           JNC ncadd
1810 201B0F         JB SUMFg,SumAddOver
1813 D21B           SETB SUMFg
                    ;
1815 D3             SETB C
1816 E549           MOV A,SUM1
1818 13             RRC A
1819 F549           MOV SUM1,A
181B E548           MOV A,SUM0
181D 13             RRC A
181E F548           MOV SUM0,A
                    ;
1820 8007           SJMP ncadd
1822 7548FF SumAddOver:  MOV SUM0,#0FFH
1825 7549FF         MOV SUM1,#0FFH
1828 C3             CLR C
1829 D006    ncadd:     POP R6
182B D001           POP R1
182D D000           POP R0
182F 22             RET

1830      SubSumC:
          ;-----------
          ; subroutine of I_contl
          ; SUM-DE->SUM ->C
          ; C->complement SUM
          ;-----------
1830 C000           PUSH R0
1832 C001           PUSH R1
1834 C006           PUSH R6
1836 C3             CLR C
1837 A848           MOV R0,SUM0
1839 A949           MOV R1,SUM1
                    ;-
183B AE4C           MOV R6,DE
183D 12019A         LCALL SUB16_8C
1840 8848           MOV SUM0,R0
1842 8949           MOV SUM1,R1
                    ;-
1844 5013           JNC ncSub
1846 E8             MOV A,R0
1847 F4             CPL A
1848 F8             MOV R0,A
1849 E9             MOV A,R1
184A F4             CPL A
184B F9             MOV R1,A
                    ;-
184C 7E01           MOV R6,#01H
184E 120132         LCALL ADD16_8C
1851 8848           MOV SUM0,R0
1853 8949           MOV SUM1,R1     ;Complement SUM+1 ->SUM
                    ;-
1855 B203           CPL SGNIFg      ;comp SGNIFg
1857 8000           SJMP ncSub
                    ;-
1859 301B11 ncSub:     JNB SUMFg,OutNcSub
185C C3             CLR C
185D E548           MOV A,SUM0
185F 33             RLC A
1860 F8             MOV R0,A
1861 E549           MOV A,SUM1
1863 33             RLC A
1864 F9             MOV R1,A
1865 4006           JC OutNcSub
1867 8848           MOV SUM0,R0
1869 8949           MOV SUM1,R1
186B C21B           CLR SUMFg
                    ;-
186D D006    OutNcSub:     POP R6
186F D001           POP R1
1871 D000           POP R0
1873 22             RET

1874      PcPiPp:
          ;-----------
          ; Subroutine
          ; DC0 + DCI + DCP -> DCPI
          ;-----------
1874 C000           PUSH R0
1876 C001           PUSH R1
1878 C006           PUSH R6
187A C3             CLR C
187B 7900           MOV R1,#00H
187D A841           MOV R0,DC0
187F AE44           MOV R6,DCI
1881 120132         LCALL ADD16_8C     ;DCI+DC0
1884 5005           JNC caryPPPp
1886 79FF           MOV R1,#0FFH
1888 78FF           MOV R0,#0FFH
188A C3             CLR C
188B AE40    caryPPPp:  MOV R6,DCP
188D 120132         LCALL ADD16_8C     ;DC0+DCI+DCP -> R1,R0
1890 121929         LCALL OVERDCPI
1893 5003           JNC outPcPiPp
1895 754FFF         MOV DCPI,#0FFH
1898 00      outPcPiPp:  NOP
1899 D006           POP R6
189B D001           POP R1
189D D000           POP R0
189F 22             RET

18A0      PcPiMp:
          ;-----------
          ; subroutine
          ; DC0 + DCI -DCP -> DCPI
          ;-----------
18A0 C000           PUSH R0
18A2 C001           PUSH R1
18A4 C006           PUSH R6
18A6 C3             CLR C
18A7 7900           MOV R1,#00H
18A9 A841           MOV R0,DC0
18AB AE44           MOV R6,DCI
18AD 120132         LCALL ADD16_8C     ;DC0 + DCI -> R1,R0
18B0 5005           JNC CarryPPMp
18B2 79FF           MOV R1,#0FFH
18B4 78FF           MOV R0,#0FFH
18B6 C3             CLR C
18B7 AE40    CarryPPMp:  MOV R6,DCP
18B9 12019A         LCALL SUB16_8C     ;DC0 + DCI -DCP -> R1,R2
18BC 4005           JC ZeroDCPI
18BE 121929         LCALL OverDCPI
18C1 8003           SJMP outPcPiMp
18C3 754F00  ZeroDCPI:  MOV DCPI,#00H
18C6 00      outPcPiMp:  NOP
18C7 D006           POP R6
18C9 D001           POP R1
18CB D000           POP R0
18CD 22             RET

18CE      PcPpMi:
          ;-----------
          ; Subroutine
          ; DC0+DCP-DCP->DCPI
          ;-----------
18CE C000           PUSH R0
18D0 C001           PUSH R1
18D2 C006           PUSH R6
18D4 C3             CLR C
18D5 7900           MOV R1,#00H
18D7 A841           MOV R0,DC0
18D9 AE40           MOV R6,DCP
18DB 120132         LCALL ADD16_8C     ;DC0 + DCP -> R1,R0
18DE 5005           JNC CarryPPMi
18E0 78FF           MOV R0,#0FFH
18E2 79FF           MOV R1,#0FFH
18E4 C3             CLR C
18E5 AE44    CarryPPMi:  MOV R6,DCI
18E7 12019A         LCALL SUB16_8C     ;DC0+ DCP - DCI ->R1,R0
18EA 4005           JC UnderPPMi
18EC 121929         LCALL OVERDCPI
18EF 8003           SJMP outPPMi
18F1 754F00  UnderPPMi:  MOV DCPI,#00H
18F4 00      outPPMi:   NOP
18F5 D006           POP R6
18F7 D001           POP R1
18F9 D000           POP R0
18FB 22             RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
18FC        PcMpMi:
            ;-----------
            ; Subroutine
            ; DC0-DCP-DCI->DCPI
            ;-----------
18FC C000               PUSH R0
18FE C001               PUSH R1
1900 C006               PUSH R6
1902 C3                 CLR C
1903 7900               MOV R1,#00H
1905 A841               MOV R0,DC0
1907 AE40               MOV R6,DCP
1909 12019A             LCALL SUB16_8C   ;DC0-DCP -> R1,R0
190C 5005               JNC carryPMMi
190E 754F00             MOV DCPI,#00H
1911 800F               SJMP outPMMi
1913 AE44   carryPMMi:  MOV R6,DCI
1915 12019A             LCALL SUB16_8C   ;DC0-DCP-DCI -> R1,R0
1918 4005               JC UnderPMMi
191A 121929             LCALL OverDCPI
191D 8003               SJMP outPMMi
191F 754F00 UnderPMMi:  MOV DCPI,#00H
1922 D006   outPMMi:    POP R6
1924 D001               POP R1
1926 D000               POP R0
1928 22                 RET

1929        OVERDCPI:
            ;-----------
            ; Subroutine
            ;-----------
1929 C3                 CLR C
192A 7400               MOV A,#00H
192C 99                 SUBB A,R1
192D 884F               MOV DCPI,R0
192F 5003               JNC endDCPI
1931 754FFF             MOV DCPI,#0FFH
1934 00     endDCPI:    NOP
1935 22                 RET

1936        OverSUM1:
            ;-----------
            ; IN SUM0,SUM1
            ; OUT SUM0
            ;-----------
1936 C000               PUSH R0
1938 C001               PUSH R1
193A C006               PUSH R6
193C C007               PUSH R7
193E 79F0               MOV R1,#0F0H
1940 7800               MOV R0,#00H
1942 AF49               MOV R7,SUM1
1944 AE48               MOV R6,SUM0
1946 1201A3             LCALL SUB16_16C
1949 5006               JNC OutOverSum
194B 7549F0             MOV SUM1,#0F0H
194E 754800             MOV SUM0,#00H
1951 D007   OutOverSum: POP R7
1953 D006               POP R6
1955 D001               POP R1
1957 D000               POP R0
1959 22                 RET


195A        OverLimit1:
            ;-----------
            ; IN R2,R1,R0
            ; OUT R0
            ;-----------
195A C3                 CLR C
195B E4                 CLR A
195C 9A                 SUBB A,R2
195D 5004               JNC Chk0
195F 78FF               MOV R0,#0FFH
1961 8007               SJMP OutChk1
1963 C3     Chk0:       CLR C
1964 E4                 CLR A
1965 99                 SUBB A,R1
1966 5002               JNC OutChk1
1968 78FF               MOV R0,#0FFH
196A 00     OutChk1:    NOP
196B 22                 MOV A,#N0_Digit

196C        OverLimit2:
            ;-----------
            ; in R2,R1,R0
            ; out R1,R0
            ;-----------
196C C3                 CLR C
196D E4                 CLR A     ;Chk limits
196E 9A                 SUBB A,R2  ;
196F 5006               JNC OutChkP0
1971 78FF               MOV R0,#0FFH
1973 79FF               MOV R1,#0FFH
1975 7A00               MOV R2,#00H
1977 22     OutChkP0:   RET
1978        OverLimit2_1:
            ;-----------
            ; in R3,R2,R1,R0
            ; out R1,R0
            ;-----------
1978 C3                 CLR C
1979 E4                 CLR A     ;Chk limits
197A 9A                 SUBB A,R2  ;
197B 4005               JC MAXX1
197D C3                 CLR C
197E E4                 CLR A
197F 9B                 SUBB A,R3
1980 5006               JNC OutChkP0_1
1982 78FF   MAXX1:      MOV R0,#0FFH
1984 79FF               MOV R1,#0FFH
1986 7A00               MOV R2,#00H
1988 22     OutChkP0_1: RET

1989        ShiftDig:
            ;-----------
            ; input R2,R1,R0
            ; Shift left R7 times
            ;-----------
1989 EF                 MOV A,R7
198A 600C               JZ EndShif
198C EA     ShiftLoop:  MOV A,R2
198D C3                 CLR C
198E 13                 RRC A
198F FA                 MOV R2,A
1990 E9                 MOV A,R1
1991 13                 RRC A
1992 F9                 MOV R1,A
1993 E8                 MOV A,R0
1994 13                 RRC A
1995 F8                 MOV R0,A
1996 DFF4               DJNZ R7,Shiftloop
1998 22     endshif:    RET

1999        ShiftDig2:
            ;-----------
            ; input R3,R2,R1,R0
            ; Shift left R7 times
            ;-----------
1999 EF                 MOV A,R7
199A 600F               JZ EndShif2
199C C3     ShiftLoop2: CLR C
199D EB                 MOV A,R3
199E 13                 RRC A
199F FB                 MOV R3,A
19A0 EA                 MOV A,R2
19A1 13                 RRC A
19A2 FA                 MOV R2,A
19A3 E9                 MOV A,R1
19A4 13                 RRC A
19A5 F9                 MOV R1,A
19A6 E8                 MOV A,R0
19A7 13                 RRC A
19A8 F8                 MOV R0,A
19A9 DFF1               DJNZ R7,Shiftloop2
19AB 22     endshif2:   RET

19AC        PlusDCD:
            ;-----------
            ; Subroutine
            ;-----------
19AC C3                 CLR C
19AD E54F               MOV A,DCPI
19AF 254A               ADD A,DCD
19B1 F53E               MOV DCPID,A
19B3 5003               JNC outPDCD
19B5 753EFF             MOV DCPID,#0FFH
19B8 22     outPDCD:    RET

19B9        MinusDCD:
            ;-----------
            ; Subroutine
            ;-----------
19B9 C3                 CLR C
19BA E54F               MOV A,DCPI
19BC 954A               SUBB A,DCD
19BE F53E               MOV DCPID,A
19C0 5003               JNC outMDCD
19C2 753E00             MOV DCPID,#00H
19C5 22     outMDCD:    RET

            ;---------- PLL routine ----------

19C6        TurnOnUp:
            ;-----------
            ; Sub Of PLL_Contl
            ;-----------
19C6 7420               MOV A,#00100000B    ;output at PORT C_5
19C8 90D002               MOV DPTR,#DtoAadd2  ;Port C_4 is Clear Signal
19CB F0                 MOVX @DPTR,A
19CC 22                 RET
19CD        TurnOnDw:
            ;-----------
            ; Sub Of PLL_V4
            ;-----------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
19CD 7440          MOV  A,#01000000B    ;output at PORT C_5
19CF 90D002        MOV  DPTR,#DtoAadd2  ;Port C_4 is Clear Signal
19D2 F0            MOVX @DPTR ,A
19D3 22            RET
19D4      TurnOff:
          ;-------
          ;-------
19D4 7400          MOV  A,#00000000B
19D6 90D002        MOV  DPTR,#DtoAadd2
19D9 F0            MOVX @DPTR ,A
19DA 22            RET


19DB      ChkF_Sig:
          ;-------------------------------------
          ; Sub of IntrTF0
          ; :used for check logic of feedback frequency
          ;-------------------------------------
19DB 9217          MOV  BufferFg,C
19DD 90D002        MOV  DPTR,#DtoAadd2   ;Check F-signal of
PLL mode
19E0 E0            MOVX A,@DPTR
19E1 A2E1          MOV  C,A_1            ;PortA_1 is F_feedbackSig
19E3 920C          MOV  HighFg,C
19E5 A217          MOV  C,BufferFg
19E7 22            RET

19E8      SetP_PLL:
          ;-------------------------------------
          ; Routine load SetP for Intr
          ; :used for Put 8-bit of DSP to N-counter(74HC161x2)
          ;-------------------------------------
19E8 E53F          MOV  A,DSP            ;define N-counter
19EA 90D000        MOV  DPTR,#DtoAadd0
19ED F0            MOVX @DPTR,A
          ;-------
19EE 90D002        MOV  DPTR,#DtoAadd2   ; generate ONE SHOT
19F1 7400          MOV  A,#00000000B
19F3 F0            MOVX @DPTR,A
19F4 7410          MOV  A,#00010000B
19F6 F0            MOVX @DPTR,A
19F7 00            NOP
19F8 00            NOP
19F9 00            NOP
19FA 7400          MOV  A,#00000000B
19FC F0            MOVX @DPTR,A
          ;-------
19FD D2AA          SETB IE_2             ;Enable External Interupt 1
19FF 22            RET

1A00      PLL_V5:
          ;-------
          ; subroutine PLL_V5
          ;-------
1A00 C20E          CLR  Intr2ACTFg
1A02 31DB    BeginV5:  ACALL ChkF_sig    ;while(LOW)
1A04 200E12        JB   Intr2ACTFg,IntrBefore
1A07 300CF8        JNB  HighFg,BeginV5
1A0A C20E          CLR  Intr2ACTFg
1A0C 31D4          ACALL TurnOff
1A0E 31CD          ACALL TurnOnDw
1A10 300EFD  WaitIntr1: JNB  Intr2ACTFg,WaitIntr1
1A13 31D4          ACALL TurnOff
1A15 C20E          CLR  Intr2ACTFg
1A17 8012          SJMP EndV5

1A19 31D4    IntrBefore: ACALL TurnOff
1A1B 31C6          ACALL TurnOnUp
1A1D C20E          CLR  Intr2ACTFg
1A1F 31DB    WhileLow1:  ACALL ChkF_sig    ;while(LOW)
1A21 200EF5        JB   Intr2ACTFg,IntrBefore
1A24 300CF8        JNB  HighFg,WhileLow1
1A27 31D4          ACALL TurnOff
1A29 C20E          CLR  Intr2ACTFg
1A2B 22      EndV5:    RET
          ;-------
          ; end routine
          ;-------


1A2C              INCL "RunModez.asm"     ;2
          ;-------
          ; All Magnage of run Program
          ;-------
1A2C      SelectRun:
          ;-------
          ; RunCurent Data variable
          ;-------
          ;---- P_controller-------
1A2C E534          MOV  A,MDwork
1A2E B40010        CJNE A,#00H,PIRun     ;P_control
1A31 C2AA                        CLR IE_2
1A33 120496          LCALL WriteAD
1A36 121483          LCALL P_Contl
1A39 854036          MOV  DABuff,DCP
1A3C 1205D6          LCALL WriteDA
1A3F 8034          SJMP OutSelRun

          ;---- PI_controller-------
1A41 E534    PIRun:    MOV  A,MDwork
1A43 B40110          CJNE A,#01H,PIDRUN
1A46 C2AA                        CLR IE_2
1A48 120496          LCALL WriteAD
1A4B 1215BA          LCALL PI_Contl
1A4E 854F36          MOV  DABuff,DCPI
1A51 1205D6          LCALL WriteDA
1A54 801F          SJMP OutSelRun

          ;---- PID_controller--------
1A56 E534    PIDRun:   MOV  A,MDwork
1A58 B40210          CJNE A,#02H,PLLrun
1A5B C2AA                        CLR IE_2
1A5D 120496          LCALL WriteAD
1A60 121676          LCALL PID_Contl
1A63 853E36          MOV  DABuff,DCPID
1A66 1205D6          LCALL WriteDA
1A69 800A          SJMP OutSelRun

          ;---- PLL_controller-------
1A6B E534    PLLRun:   MOV  A,MDwork
1A6D B40305          CJNE A,#03H,OutSelRun
1A70 D2AA                        SETB IE_2
1A72 121A00          LCALL PLL_V5

          ;---------
1A75 120509  OutSelRun: LCALL togglePV ;toggle for blink
1A78 22            RET

1A79      PvInLED:
          ;---------
          ; Program show LED for P1 indicate PV
          ;---------
1A79 C3            CLR  C
1A7A E535          MOV  A,ADbuff
1A7C 941F          SUBB A,#1FH
1A7E 5005          JNC  LED1On
1A80 759000        MOV  P1,#00000000B
1A83 8042          SJMP OutLEDon
1A85 941F    LED1On:   SUBB A,#1FH
1A87 5005          JNC  LED2On
1A89 759001        MOV  P1,#00000001B
1A8C 8039          SJMP OutLEDon
1A8E 941F    LED2On:   SUBB A,#1FH
1A90 5005          JNC  LED3On
1A92 759003        MOV  P1,#00000011B
1A95 8030          SJMP OutLEDon
1A97 941F    LED3On:   SUBB A,#1FH
1A99 5005          JNC  LED4On
1A9B 759007        MOV  P1,#00000111B
1A9E 8027          SJMP OutLEDon
1AA0 941F    LED4On:   SUBB A,#1FH
1AA2 5005          JNC  LED5On
1AA4 75900F        MOV  P1,#00001111B
1AA7 801E          SJMP OutLEDon
1AA9 941F    LED5On:   SUBB A,#1FH
1AAB 5005          JNC  LED6On
1AAD 75901F        MOV  P1,#00011111B
1AB0 8015          SJMP OutLEDon
1AB2 941F    LED6On:   SUBB A,#1FH
1AB4 5005          JNC  LED7On
1AB6 75903F        MOV  P1,#00111111B
1AB9 800C          SJMP OutLEDon
1ABB 941F    LED7On:   SUBB A,#1FH
1ABD 5005          JNC  LED8On
1ABF 75907F        MOV  P1,#01111111B
1AC2 8003          SJMP OutLEDon
1AC4 7590FF  LED8On:   MOV  P1,#11111111B
1AC7 22      OutLEDon: RET
          ;-------
          ; end routine
          ;-------


1AC8              INCL "Titleaho.asm"
          ;-------------------
          ;              ||
          ; routine for Show ####
          ;-------------------
1AC8      MainTitle:
1AC8 C002          PUSH R2
1ACA C003          PUSH R3
1ACC C004          PUSH R4
1ACE 7B00          MOV  R3,#00H
1AD0 7C00          MOV  R4,#00H
1AD2 7480          MOV  A,#80H
1AD4 90A003        MOV  DPTR,#DISP1D
1AD7 F0            MOVX @DPTR,A      ; initial 8255
1AD8 90B003        MOV  DPTR,#DISP2D
1ADB F0            MOVX @DPTR,A      ; initial 8255
1ADC 7590F0        MOV  P1,#0F0H
1ADF 7AFF          MOV  R2,#0FFH
1AE1 120100        LCALL DELAY2
          ;---------
1AE4 901B35  cometo:  MOV  DPTR ,#TitleTab
1AE7 EC            MOV  A,R4
1AE8 93            MOVC A,@A+DPTR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
1AE9 90A000          MOV DPTR,#DISP1A
1AEC F0              MOVX @DPTR,A
1AED 0C              INC R4
                ;-----------
1AEE 901B35          MOV DPTR ,#TitleTab
1AF1 EC              MOV A,R4
1AF2 93              MOVC A,@A+DPTR
1AF3 90A001          MOV DPTR,#DISP1B
1AF6 F0              MOVX @DPTR,A
1AF7 0C              INC R4
                ;-----------
1AF8 901B35          MOV DPTR,#TitleTab
1AFB EC              MOV A,R4
1AFC 93              MOVC A,@A+DPTR
1AFD 90A002          MOV DPTR,#DISP1C
1B00 F0              MOVX @DPTR,A
1B01 0C              INC R4
                ;-----------
1B02 901B35          MOV DPTR,#TitleTab
1B05 EC              MOV A,R4
1B06 93              MOVC A,@A+DPTR
1B07 90B000          MOV DPTR,#DISP2A
1B0A F0              MOVX @DPTR,A
1B0B 0C              INC R4
                ;-----------
1B0C 901B35          MOV DPTR,#TitleTab
1B0F EC              MOV A,R4
1B10 93              MOVC A,@A+DPTR
1B11 90B001          MOV DPTR,#DISP2B
1B14 F0              MOVX @DPTR,A
1B15 0C              INC R4
                ;-----------
1B16 901B35          MOV DPTR,#TitleTab
1B19 EC              MOV A,R4
1B1A 93              MOVC A,@A+DPTR
1B1B 90B002          MOV DPTR,#DISP2C
1B1E F0              MOVX @DPTR,A
1B1F 0C              INC R4
                ;-----------
1B20 7AFF            MOV R2,#0FFH
1B22 120100          LCALL DELAY2
1B25 0B              INC R3
1B26 AC03            MOV R4,R3
1B28 75900F          MOV P1,#0FH
1B2B BB0DB6          CJNE R3,#13D,cometo
1B2E D004            POP R4
1B30 D003            POP R3
1B32 D002            POP R2
1B34 22              RET
1B35        TitleTab:
1B35 00              DFB 00000000B      ;" "
1B36 00              DFB 00000000B      ;" "
1B37 00              DFB 00000000B      ;" "
1B38 40              DFB 01000000B      ;"-"
1B39 00              DFB 00000000B      ;" "
1B3A 00              DFB 00000000B      ;" "

1B3B 00              DFB 00000000B      ;" "
1B3C 5B              DFB 01011011B      ;"H"
1B3D 67              DFB 01100111B      ;"E"
1B3E 07              DFB 00000111B      ;"L"
1B3F 07              DFB 00000111B      ;"L"
1B40 3F              DFB 00111111B      ;"O"

1B41 00              DFB 00000000B      ;" "
1B42 43              DFB 01000011B      ;"r"
1B43 67              DFB 01100111B      ;"E"
1B44 7B              DFB 01111011B      ;"A"
1B45 5E              DFB 01011110B      ;"d"
1B46 5D              DFB 01011101B      ;"Y"

        ;-----------
        ; end routine
        ;-----------


0000            ORG 0000H
0000 021C00       LJMP Main

0003            ORG 0003H
0003 120F0A       LCALL IntrIE0_2   ;keyb:
0006 32           RETI

000B            ORG 000BH
000B 1204F1       LCALL IntrTF0     ;Timer interupt
000E 32           RETI

0013            ORG 0013H
0013 1204DF       LCALL IntrIE1     ;reference PLL
0016 32           RETI

0023            ORG 0023H
0023 120573       LCALL IntrSM      ;serial interupt
0026 32           RETI

1C00            ORG 1C00H
1C00 758150  Main:  MOV SP,#50H
1C03 C2D3         CLR PSW_3         ;use register bank0
1C05 C2D4         CLR PSW_4         ;use register bank0
1C07 7AFF         MOV R2,#0FFH
1C09 120100       LCALL DELAY2
1C0C 1203E0       LCALL InitAllVar
1C0F 12047A       LCALL InitMn
1C12 121AC8       LCALL MainTitle
1C15 00      Ma1: NOP
1C16 121A2C       LCALL SelectRun
        ;      LCALL PVinLED     ;show LED of P1 for indicate PV
1C19 80FA         SJMP Ma1
        ;-----------
        ; End Main
        ;-----------
0000            END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง

ซอฟต์แวร์ลิสต์ภาษาซีของโปรแกรมพล้านต์มอนิเตอริง

ซอฟต์แวร์ลิสต์ภาษาซี ของโปรแกรมพล้านต์มอนิเตอริง ซึ่งใช้กับโฮสคอมพิวเตอร์เพื่อสำหรับดูแลเหตุการณ์ต่างที่เกิดขึ้นในพล้านต์ โดยรับข้อมูลมาจากตัวดิจิทัลคอนโทรลเลอร์ ดังกล่าวในเนื้อเรื่อง

ซอฟต์แวร์ของโปรแกรมพล้านมอนิเตอริ่งนี้ จัดแบ่งเป็น ไฟล์ไว้ดังนี้

1. pntmnt.cpp
2. pulldown.cpp
3. dopull.cpp
4. tagset.cpp
5. trendrec.cpp
6. T&Sintr.cpp
7. bargraph.cpp
8. submenu.cpp
9. underbar.cpp
10. wingraph.cpp
11. mouseobj.cpp

นำไฟล์ทั้ง 11 ไฟล์มาคอมไพล์โดย โปรแกรม TURBO C ++ และนำมาลิงค์รวมกัน ก็จะได้โปรแกรมพล้านมอนิเตอริงอย่างสมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// pntmnt.cpp
/*****************************************/
/*  Main program of PLANT MONITORING   */
/*****************************************/

#include <graphics.h>
#include <conio.h>
#include <stdlib.h>
#include <stdio.h>
#include "pulldown.cpp"


void graphinit(void)
{
  int gdriver, gmode, errorcode;
  gdriver = VGA;
  gmode   = VGAHI;
  initgraph(&gdriver, &gmode, " ");
  errorcode = graphresult();
  if (errorcode != grOk)  /* an error occurred */
  {
            printf("Graphics error: %s\n", grapherrormsg(errorcode));
            printf("This program supports VGA only");
            printf("Press any key to halt:");
            getch();
            exit(1);            /* return with error code */
  }
}

void main(void)
{
          graphinit();
          showmain();
          loadcfg();
          barhead();
          headselect();
          initserial();
          closegraph();
          printf(" Thank for you had run program.");
          exit(0);
}

//==== end PNTMNT.CPP ========

// pulldown.cpp
/*********************************/
/*  main pulldown menu routine */
/*********************************/

#include <graphics.h>
#include <string.h>
#include "dopull.cpp"

#define  MAXHEAD 4
#define  HIGHBAR 16
#define  YGAP 6

void dopulldown(void);

typedef  char str15[15];  /* head of pulldownmenu */
typedef struct  hdtxt {
            int backcl;
            int headcl;
            int txthcl;
            int barcl;
            int bartxtcl;
            int nohead;
            str15  txth[MAXHEAD];
};
const hdtxt htxt = { BLUE,
          DARKGRAY,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                WHITE,
                YELLOW,
                BLACK,
                4,
                " FILE ",
                " OPTIONS ",
                " MONITOR ",
                " CONFIG " };

        windobj barpull;
        int statushead;

//=====================
// pulldown routine
//=====================

void barhead(void)
{
        umouse.show(FALSE);
        int oldlen = 0;
        settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
        setviewport(0,0,getmaxx(),getmaxy(),1); // CLIP_ON = 1
        for(int i = 0 ;i<statushead;i++) oldlen += strlen(htxt.txth[i])*8;
        barpull.window_open(oldlen,2,oldlen+strlen(htxt.txth[statushead])*8,
                                                HIGHBAR);
        barpull.window_create(oldlen,2,oldlen+strlen(htxt.txth[statushead])*8,
                                                HIGHBAR,htxt.barcl);
        setcolor(htxt.bartxtcl);
        settextjustify(LEFT_TEXT,TOP_TEXT);
        outtextxy(oldlen,YGAP,htxt.txth[statushead]);
        umouse.show(TRUE);
}

void headoff(void)
{
        umouse.show(FALSE);
        barpull.window_close();
        umouse.show(TRUE);
}

void pullshow(void)
{
        umouse.show(FALSE);
        switch (statushead) {
        case 0 : subp0.showsub();subp0.select();subp0.subclose();break;
        case 1 : subp1.showsub();subp1.select();subp1.subclose();break;
        case 2 : subp2.showsub();subp2.select();subp2.subclose();break;
        case 3 : subp3.showsub();subp3.select();subp3.subclose();break;
        }
        umouse.show(TRUE);
}

void choiceIf(void)
{
        if (statushead - 1 < 0) statushead = MAXHEAD - 1;
        else statushead -= 1;
        headoff();
        barhead();
        pullshow();
}

void choicert(void)
{
        if (statushead + 1 >= MAXHEAD) statushead = 0;
        else statushead += 1 ;
        headoff();
        barhead();
        pullshow();
}
void headselect(void)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
        #define TXTROW 16
        #define RETURN 7181
        #define ESC   283
        #define LEFT   19200
        #define RIGHT  19712

        boolean exith = FALSE;
        int len0 = strlen(htxt.txth[0]);        // tail of txt0
        int len1 = strlen(htxt.txth[1]) + len0;
        int len2 = strlen(htxt.txth[2]) + len1;
        int len3 = strlen(htxt.txth[3]) + len2;   // tail of all txt
        do {
          umouse.show(TRUE);
          if(bioskey(1)!=0) {
                int ch = bioskey(0);
                switch (ch) {
                        case RETURN:  pullshow(); break;
                        case   ESC:  exith = TRUE;break;
                        case  LEFT:  choicelf(); break;
                        case  RIGHT:  choicert(); break;
                }
          }
          umouse.querybtndn(BUTTONL);  // mouse action chiooe
          if(umouse.mbutton.opcount > 0 ) {
                 if( umouse.mbutton.ypos < HIGHBAR) {
                                int oldstat = 0;
                                if(    umouse.mbutton.xpos < len0*8) oldstat = 0;
                                else if(umouse.mbutton.xpos < len1*8) oldstat = 1;
                                else if(umouse.mbutton.xpos < len2*8) oldstat = 2;
                                else if(umouse.mbutton.xpos < len3*8) oldstat = 3;
                                statushead = oldstat;
                                headofl();
                                barhead();
                                pullshow();
                }
          }
        dopulldown();     // DO function
        }while(exith == FALSE );
}

//=============
// Title text
//=============
void showtitle(void)
{
        char *title[] = {
                  "  Industial process control schemetic    ",
                  "    PLANT  MONITORING                    ",
                  "              by                         ",
                  "      Dr. Yothin   Prempraneerach        ",
                  "      Mr. Jetanar  Singsin               ",
                  "  Control engineering department KMITL.  ",
                  };
                  settextstyle(0,HORIZ_DIR,2);
                  setcolor(YELLOW);
                  outtextxy(10 ,60 ,title[0]);      //Write title line 1

                  settextstyle(0,HORIZ_DIR,3);
                  setcolor(RED);
                  outtextxy(10 ,150 ,title[1]);      //Write title line 2

                  settextstyle(0,HORIZ_DIR,2);
                  setcolor(WHITE);
                  outtextxy(10,230 ,title[2]);      //Write title line 3

                  settextstyle(0,HORIZ_DIR,2);
                  setcolor(GREEN);
                  outtextxy(10,270,title[3]);      //Write title line 4

                  settextstyle(0,HORIZ_DIR,2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    setcolor(GREEN);
                    outtextxy(10,310,title[4]);     //Write title line 4

                    settextstyle(0,HORIZ_DIR,2);
                    setcolor(CYAN);
                    outtextxy(10,360,title[5]);     //Write title line 4

                    getch();
}

//=======================
// introduction text
//=======================

void showintro(void)
{
        char *title[] = {
" DISCUSSION:                    ",
"   - First,You  ought to read instruction manual before program running.",
"   - In your mind ,if your TAG-file  don,t exist then creat the new.    ",
"   - TAG-file is already when you use menu OPEN SETTING.                ",
"   - Check to macthing  of serial interface and boudrate.               ",
"   - THANK YOU!                                                         ",
"                                                                        ",
                    };
                    setviewport(10,50,getmaxx()-10,getmaxy()-50,0);
                    clearviewport();
                    setviewport(0,0,getmaxx(),getmaxy(),0);
                    settextstyle(0,HORIZ_DIR,1);
                    setcolor(YELLOW);
                    outtextxy(10 ,60  ,title[0]);
                    outtextxy(10 ,80  ,title[1]);
                    outtextxy(10 ,100 ,title[2]);
                    outtextxy(10 ,120 ,title[3]);
                    outtextxy(10 ,140 ,title[4]);
                    outtextxy(10 ,160 ,title[5]);
                    outtextxy(10 ,180 ,title[6]);


}

//=============================================
// Auto load configulation file to run
//=============================================

void loadcfg(void)
{
        str25  strname = "plant.cfg";
        windobj mesgwin;
        cfgstrct cfglist;
        FILE *in;

        if((in = fopen(strname,"r")) == NULL)
        {
                mesgwin.window_open(250,350,500,400);
                mesgwin.window_creatsh(250,350,500,400,RED,"",YELLOW,BLACK);
                setcolor(WHITE);
                outtextxy(266,366," Config not found <ENTER>");
                getch();
                mesgwin.window_close();
                fclose(in);
               }
               else{
                fread(&cfglist,sizeof(cfgstrct),1,in);
                fclose(in);
                statusport  =  cfglist.statusport;
                statusbaud  =  cfglist.statusbaud;
                statustgraph =  cfglist.statustgraph;
                time_samp   =  cfglist.time_samp;
                statusvar   =  cfglist.statusvar;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                realmodefg  =  cfglist.realmodefg;
                for(int i = 0;i<3;i++)
                        if(0>cfglist.trend_cal[i]>99) trend_cal[i] = 0.0;
                        else trend_cal[i] = cfglist.trend_cal[i];
                strcpy(gfilename, cfglist.gfilename);
                //..
                baudconst  =  statusbaud+1;
               }
}

//=====================
// show main menu on CRT
//=====================

void showmain(void)
{

        choiceact acttf;      // set submenu ACTIVE or INACTIVE
        if(!umouse.testmouse()) {
                closegraph();
                printf(" This program requires a mouse ");
                exit(1);
        }
        umouse.initialize(arrow);

        for(int j = 0;j < MAXROW; j++)
                        acttf[j] = TRUE;
        subp0.subact(acttf);
        subp1.subact(acttf);
        subp2.subact(acttf);
        acttf[2] = FALSE;
        acttf[5] = FALSE;
        subp3.subact(acttf);
        cleardevice();
        setbkcolor(htxt.backcl);
        setfillstyle(SOLID_FILL,htxt.headcl);
        bar(0,0,getmaxx(),HIGHBAR+3);
        setcolor(LIGHTGREEN);
        rectangle(0,0,getmaxx(),HIGHBAR+3);
        int oldlen = 0;
        for(int i=0;i<htxt.nohead;i++) {
                setcolor(htxt.txthcl);
                outtextxy(oldlen,YGAP,htxt.txth[i]);
                oldlen += strlen(htxt.txth[i])*8;
        }
        showtitle();    // write title names
        showintro();
}


void dopulldown(void) {
        if(subp0.subactive == TRUE ||
                subp1.subactive == TRUE ||
                subp2.subactive == TRUE ||
                subp3.subactive == TRUE) {
                headoff();
                switch(statushead) {
                        case 0: if(subp0.subactive == TRUE)
                                                switch(subp0.statusrow) {
                                                case 0:do00();break;
                                                case 1:do01();break;
                                                case 2:do02();break;
                                                case 3:do03();break;
                                                };
                                        subp0.subactive = FALSE;
                                        break;
                        case 1:  if(subp1.subactive == TRUE)
                                                switch(subp1.statusrow) {
                                                case 0:do10();break;
                                                case 1:do11();break;
                                                case 2:do12();break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        case 3:do13();break;
                                        case 4:do14();break;
                                        };
                                    subp1.subactive = FALSE;
                                    break;
                        case 2:  if(subp2.subactive == TRUE)
                                        switch(subp2.statusrow) {
                                        case 0:do20();break;
                                        case 1:do21();break;
                                        case 2:do22();break;
                                        case 3:do23();break;
                                        case 4:do24();break;
                                        case 5:do25();break;
                                        };
                                    subp2.subactive = FALSE;
                                    break;
                        case 3:  if(subp3.subactive == TRUE)
                                        switch(subp3.statusrow) {
                                        case 0:do30();break;
                                        case 1:do31();break;
                                        case 2:do32();break;
                                        case 3:do33();break;
                                        case 4:do34();break;
                                        case 5:do35();break;
                                        case 6:do36();break;
                                        };
                                    subp3.subactive = FALSE;
                                    break;
                    };//end case
                    barhead();
                    }; //end if
};

// ======== end PULLDOWN.CPP ==========

// dopull.cpp
/*********************************/
/*  submenu working for task  */
/*********************************/

#include "submenu.cpp"
#include "trendrec.cpp"
#include "bargrph.cpp"
#include "tagset.cpp"
#include "t&sintr.cpp"

//=============================
// define Submenu constant
//=============================
const subm  subm0 = { LIGHTGRAY,
                                    LIGHTGREEN,
                                    DARKGRAY,
                                    WHITE,
                                    DARKGRAY,
                                    10,
                                    30,
                                    16,
                                    2,
                                    "< FILE >",
                                    YELLOW,
                                    " OPEN TAG FILE ",
                                    " QUIT PROGRAM  ",
                                    "NO.3","NO.4","NO.5","NO.6","NO.7","NO.8"};
const subm  subm1 = { LIGHTGRAY,
                                    LIGHTGREEN,
                                    DARKGRAY,
                                    WHITE,
                                    DARKGRAY,
                                    70,
                                    30,
                                    15,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        5,
                                        "< OPTIONS >",
                                        YELLOW,
                                        " SERIAL PORT ",
                                        " BAUD RATE   ",
                                        " NO.  GRAPH  ",
                                        " SET SAMPLING ",
                                        " TREND VAR   ",
                                        "NO.6","NO.7","NO.8"};
const subm  subrr2 = { LIGHTGRAY,
                                        LIGHTGREEN,
                                        DARKGRAY,
                                        WHITE,
                                        DARKGRAY,
                                        130,
                                        30,
                                        15,
                                        4,
                                        "< MONITOR >",
                                        YELLOW,
                                        " TAG EDITING ",
                                        " PLANT MEMIC ",
                                        " TREND REC ",
                                        " BAR GRAPH ",
                                        "NO.5","NO.6","NO.7","NO.8"};
const subm  subrr3 = { LIGHTGRAY,
                                        LIGHTGREEN,
                                        DARKGRAY,
                                        WHITE,
                                        DARKGRAY,
                                        200,
                                        40,
                                        14,
                                        7,
                                        "< CONFIG >",
                                        YELLOW,
                                        " SAVE CONFIG ",  //0
                                        " LIST CONFIG ",  //1
                                        "-----------",    //2
                                        " RANDOMIZE   ",  //3
                                        " REAL  TIME  ",  //4
                                        "-----------",    //5
                                        " TREND CALB  ",  //6
                                        "NO.7","NO.8"};

sub subp0(subrr0),subp1(subrr1),subp2(subrr2),subp3(subrr3);//submenu of main

boolean  realmodefg; // It's flag that indicate the working mode.
                                                          // use in "Dopull.cpp"
float trend_cal[3];  //trend calibating
//=============================
// run pulldown routine =
//=============================

void   blankscr(void);        // head declare


// --------< File Menu >---------
//=============================
// Open TAG file
//=============================

str25 gfilename; //open file name

void do00(void)
{
        FILE *steam;
        char *inname;
        const namefm namefile={" Put TAG file name : <_          .>*",
                                                    _          .tag "};
```

เอกสารนี้มีลิขสิทธิ์ สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        blankscr();
        // make help text
        box_creatl(50,40,getmaxx()-50,120,DARKGRAY,YELLOW);
        setcolor(WHITE);
        outtextxy(120,60, "   - Please, put you TAG file name in YELLOW box. ");
        outtextxy(120,80, "   - If hasn't TAG file then it will be created   ");
        outtextxy(120,100,"     by TAG SETTING menu.                         ");

        inname = getname(100,200,namefile);
        strcpy(gfilename,inname);       // data in GOLBAL
        strcat(gfilename,".tag");
        if((steam = fopen(gfilename,"rb")) == NULL){
                beep();
                subp0.massage(" FILE not exist <ENTER>",CYAN);
        }else subp0.massage(" Open TAG file OK <ENTER>",GREEN);
        fclose(steam);
}

//======
// Quit
//======
void do01(void)
{
        closegraph();
        exit(1);
}

void do02(void){}   //reserve
void do03(void){}   //reserve

//---------- < SETTING >-------
//=========================
// Serial port setting
//=========================
int statusport;

void do10(void){

const subm  portst = {LIGHTGRAY,
                      YELLOW,
                      DARKGRAY,
                      WHITE,
                      BLACK,
                      100,
                      200,
                      17,
                      4,
                      " <PORT SETTING> ",
                      YELLOW,
                      "  COM 1:(3F8)  ",
                      "  COM 2:(2F8)  ",
                      "    COM 3:   ",
                      "    COM 4:   ",
                      " ",
                      " ",
                      " ",
                      " "};

        sub portsub(portst);    // decare  submenu variable of baud
        blankscr();

        // make help text
        box_creatl(30,40,getmaxx()-30,170,DARKGRAY,YELLOW);
        setcolor(LIGHTGREEN);
        outtextxy(50,80, " SERIAL PORT SETTING ");
        outtextxy(50,100," - Choose COM: port of serial interface to controller. ");
        outtextxy(50,120," - Up,Down and ENTER key  for choosing.                 ");
        outtextxy(50,140," - Might be indentical COM: with controller             ");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            portsub.subact(bartype1);
            portsub.showsub();
            portsub.select();
            portsub.subclose();
            statusport = portsub.statusrow;
//          comconst  = status+1;
            subp1.massage(" COM: setting OK! ",RED);
}

//=======================
// Baud rate setting
//=======================
int statusbaud;
void do11(void){

const subm  baudst = {LIGHTGRAY,
                                                YELLOW,
                                                DARKGRAY,
                                                WHITE,
                                                BLACK,
                                                100,
                                                200,
                                                17,
                                                7,
                                                " <BAUD SETTING> ",
                                                YELLOW,
                                                "   150  BAUD ",
                                                "   300  BAUD ",
                                                "   600  BAUD ",
                                                "  1200  BAUD ",
                                                "  2400  BAUD ",
                                                "  4800  BAUD ",
                                                "  9600  BAUD ",
                                                " "};


            sub baudsub(baudst);    // decare  submenu variable of baud
            blankscr();

            // make help text
            rectangle(30,40,getmaxx()-30,170);
            box_creatl(30,40,getmaxx()-30,170,DARKGRAY,YELLOW);
            setcolor(LIGHTGREEN);
            outtextxy(50,80, " BAUD RATE SETTING ");
            outtextxy(50,100," - Choose BAUD port of serial interface to controller. ");
            outtextxy(50,120," - Up,Down and ENTER key  for choosing.                ");
            outtextxy(50,140," - Might be indentical BAUD with controller.           ");

            baudsub.subact(bartype1);
            baudsub.showsub();
            baudsub.select();
            baudsub.subclose();
            statusbaud = baudsub.statusrow;
            baudconst  = statusbaud+1;       //decare in T&SINTR.CPP
            subp1.massage(" BAUD OK! ",RED);
            initserial();
}

//=======================
// No. graph on CRT
//=======================
void do12(void)
{
                const subm  numst = {LIGHTGRAY,
                                                YELLOW,
                                                DARKGRAY,
                                                WHITE,
                                                DARKGRAY,
                                                150,
                                                300,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                                14,
                                                3,
                                                "< NO.>",
                                                YELLOW,
                                                "  1-graph  ",
                                                "  2-graph  ",
                                                "  4-graph  ",
                                                "  6-graph  ",
                                                " ",
                                                " ",
                                                " ",
                                                " "};

        sub numsub(numst);      // decare  submenu variable of NO. graph

        choiceact acttf;

        umouse.show(FALSE);
        blankscr();
        box_creatl(30,50,getmaxx()-30,120,DARKGRAY,YELLOW);
        setcolor(WHITE);
        outtextxy(100,70," You need to set the number of trend recorder graph ");
        outtextxy(100,90,"          that show on monitor.                ");
        for(int j = 0;j < MAXROW; j++) acttf[j] = TRUE;
        numsub.subact(acttf);
        numsub.showsub();
        numsub.select();
        numsub.subclose();
        statustgraph = numsub.statusrow;
        umouse.show(TRUE);
        subp1.massage(" No. TREND graph OK! <RETURN> ",RED);
}

//=======================
// set samplingtime
//=======================
float  time_samp;

void do13(void)
{
        char *name;
        name  = " Put sampling time.<sec>:";

        double oldsampt;

        blankscr();
        box_creatl(30,40,getmaxx()-30,220,DARKGRAY,YELLOW);
        setcolor(WHITE);
        outtextxy(100,60, "SAMPLING SETTING");
        outtextxy(100,80, "- This is effected on the TREND recoder.");
        outtextxy(100,100,"- Default value of sampling time is fastest.");
        outtextxy(100,120,"- If you want to set the sampling value then ");
        outtextxy(100,140,"    you put  numeric in YELLOW box.    ");
        outtextxy(100,160,"- Keep in mind,very small samling value is very ");
        outtextxy(100,180,"    much detail of graph and very leak to control.");
        oldsampt  = (double)time_samp;
        time_samp = getfloat(120,300,oldsampt,name);
        subp1.massage(" Sampling time setting OK! ",LIGHTRED);
}

//--- Trend Recoder variable ----


void do14(void){

const subm  varst = {LIGHTGRAY,
                                                YELLOW,
                                                DARKGRAY,
                                                WHITE,
                                                BLACK,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                100,
                                200,
                                20,
                                5,
                                " <GRAPH VARIABLE> ",
                                YELLOW,
                                "     PV       ",
                                "   PV & SP    ",
                                " PV & SP & CV  ",
                                "     CV       ",
                                "   CV & SP    ",
                                " ",
                                " ",
                                " "};

        sub varsub(varst);    // decare  submenu variable of graph varible
        blankscr();

        // make help text
        box_creatl(30,40,getmaxx()-30,170,DARKGRAY,YELLOW);
        setcolor(LIGHTGREEN);
        outtextxy(50,80, " TREND GRAPH VARIBALE");
        outtextxy(50,100," - Choose the graph line vairble for plotting in TREND.");
        outtextxy(50,120," - Up,Down and ENTER key  for choosing.          ");
        outtextxy(50,140,"- PV:process variable CV:control variavle SP:setpoint ");

        varsub.subact(bartype1);
        varsub.showsub();
        varsub.select();
        varsub.subclose();
        statusvar = varsub.statusrow;
        switch(statusvar){
        case 0: pvflag = TRUE;cvflag = FALSE;spflag = FALSE;break;
        case 1: pvflag = TRUE;cvflag = FALSE;spflag = TRUE ;break;
        case 2: pvflag = TRUE;cvflag = TRUE ;spflag = TRUE ;break;
        case 3: pvflag = FALSE;cvflag = TRUE;spflag = FALSE;break;
        case 4: pvflag = FALSE;cvflag = TRUE;spflag = TRUE ;break;
        }
        subp1.massage(" VARIABLE: OK! ",RED);
}


//-------------< MONITOR >-----------------
//=============================
// TAG setting
//=============================
void do20(void)
{
        tagwork();
        subp2.massage("  TAG OK! ",RED);
}

//=============================
// Plant memic
//=============================
void do21(void)
{
        subp2.massage(" This option don't finish. ",RED);
}

//=============================
//== trendrecorder ==
//=============================
#define  GRAPHCL  DARKGRAY    //out bound
#define  TRENDCL  DARKGRAY    //in  bound

//=============================
// TREND recorder display
//=============================
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  trend tunit0( 5, 30,GRAPHCL,TRENDCL,YELLOW);
  trend tunit1( 5,220,GRAPHCL,TRENDCL,YELLOW);
  trend tunit2(319,30,GRAPHCL,TRENDCL,YELLOW);
  trend tunit3(319,220,GRAPHCL,TRENDCL,YELLOW);

void do22(void){

const undertxt undtrend = {DARKGRAY,
                                                WHITE,
                                                YELLOW,
                                                BLACK,
                                                9,
                                                " EXIT ", //1
                                                " TRACE ", //2
                                                " STOP ", //3
                                                " CLEAR ", //4
                                                " VALUE ", //5
                                                " SAVE ", //6
                                                " LOAD ", //7
                                                " RECORD ", //8
                                                " PLAY "}; //9

  acttf stpgrpfg = {0,0,1,0,0,0,0,0,0,0}; //TRUE = 1: FALSE = 0
  acttf stpvalfg = {0,0,1,0,1,0,0,0,0,0};
  acttf rungrpfg = {1,1,0,1,1,1,1,1,1,0};
  acttf loadgrpfg ={1,0,0,1,1,1,1,1,1,0};
  acttf plygrpfg = {0,1,1,0,1,0,0,0,0,0};
  acttf recgrpfg = {0,1,1,0,0,0,0,0,0,0};

  extern void getserialdata(float &,float &,float &,float &,int,boolean &);
  extern void do22sub(str15 t0,str15 u0,float m0,float r0,float x0,int l0,int h0,
                      str15 t1,str15 u1,float m1,float r1,float x1,int l1,int h1,
                      str15 t2,str15 u2,float m2,float r2,float x2,int l2,int h2,
                      str15 t3,str15 u3,float m3,float r3,float x3,int l3,int h3);

  extern void do22sub0();   //free data mem
  extern void  do22sub1(FILE *);
  extern void  do22sub2(int &,int &,int &,int & ,float &,float &);
  extern void  do22sub3(int,int,int,int);
  extern void  do22sub4(float usemagnit,float useratio,int trec0);//load 1grp
  extern void  do22sub41(float,float,int,int);
  extern void  do22sub42(float,float,int,int,int,int);
  extern void  do22sub5(int trec0,int trec1,int trec2,int trec3);//dsp box
  extern char  *do22sub60();
  extern void  do22sub6(FILE *recfile,int trec0,int trec1,int trec2,int trec3,
                                                boolean &recordfg);
  extern void  do22sub7(FILE *recfile,float usemagnit,
                                                float useratio,long &curpos);
  extern void  outmesg(int left,int top,char *buff); //serial attempt

  int trec0 = 1;
  int trec1 = 1;
  int trec2 = 1;  // initialized
  int trec3 = 1;


  boolean okflag; //flag for serial port
  boolean recordfg = FALSE;
  boolean playfg   = FALSE;

  FILE *inf,*recfile;

  str25 buff;
  float usemagnit,useratio;
  int toggle1,toggle2,toggle3,toggle4;
  long curpos;

  initserial();
  if((inf = fopen(gfilename,"rb")) == NULL){
        beep();
        subp2.massage(" TAG file is not open.",YELLOW);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            fclose(inf);
            return;
}
do22sub1(inf);
fclose(inf);
do22sub2(trec0,trec1,trec2,trec3,usemagnit,useratio);

blankscr();

//creat bound graph

do22sub(ctrtag[trec0].tagname, ctrtag[trec0].engunits,
                                usemagnit, useratio,
                                ctrtag[trec0].maxmt,
                                ctrtag[trec0].lowscl,
                                ctrtag[trec0].upperscl,
            ctrtag[trec1].tagname,
                                ctrtag[trec1].engunits,
                                usemagnit,
                                useratio,
                                ctrtag[trec1].maxmt,
                                ctrtag[trec1].lowscl,
                                ctrtag[trec1].upperscl,
            ctrtag[trec2].tagname,
                                ctrtag[trec2].engunits,
                                usemagnit,
                                useratio,
                                ctrtag[trec2].maxmt,
                                ctrtag[trec2].lowscl,
                                ctrtag[trec2].upperscl,
            ctrtag[trec3].tagname,
                                ctrtag[trec3].engunits,
                                usemagnit,
                                useratio,
                                ctrtag[trec3].maxmt,
                                ctrtag[trec3].lowscl,
                                ctrtag[trec3].upperscl);


    // uses underbar
    //..
    boolean exitdo = FALSE;
    float pvu,cvu,spu,timeu;
    float timebuf1,timebuf2,timebuf3,timebuf4;

    underbar bartrend(undtrend);
    bartrend.setact(rungrpfg);   //all TRUE
    bartrend.show();


    do{
        bartrend.baroff();
        bartrend.baron();
        do{
            bartrend.select();
        }while(bartrend.exith == FALSE);
        switch(bartrend.statusund){
            case 0:exitdo = TRUE;
                   do22sub0();
                   tunit0.reset_timecountbox();
                   tunit1.reset_timecountbox();
                   tunit2.reset_timecountbox();
                   tunit3.reset_timecountbox();
                   break; //exit
            case 1:bartrend.setact(stpgrpfg);  //trace
                   bartrend.show();
                   switch(statustgraph){
                   // plot one graph
                   case 0:bartrend.baroff();bartrend.baron();
                          do{
                              if(playfg&&feof(recfile)) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    bartrend.setact(stpvalfg);
                    bartrend.show();
                    outmesg(tunit0.posx+3,
                                    tunit0.posy+3," file end ");
                    break;
                  }
                  //..
                  if(playfg){
                   int i;
                   fseek(recfile, curpos, SEEK_SET);
                   fscanf(recfile," %i %f %f %f %f ",
                                        &i,&pvu,&cvu,&spu,&timeu);
                   curpos = ftell(recfile);
                   if(i!=1) {
                              outmesg(tunit0.posx+3,tunit0.posy+3,
                                                  " reading file error #0 ");
                              fclose(recfile);
                              playfg = FALSE;
                              break;
                   }
                   okflag = TRUE;
                  }else{ getserialdata(pvu,cvu,spu,timeu,
                                                              trec0,okflag);
                         timeu = timeu + time_samp+trend_cal[0];
                  }
                  if(okflag){
                   tunit0.graphplot(pvu,cvu,spu,timeu);
                   if(recordfg);
                         fprintf(recfile," %i %f %f %f %f ",
                                              1,pvu,cvu,spu,timeu);
                  }
                  else outmesg(tunit0.gphposx1,tunit0.gphposy1,
                                    " Serial attempt ");
                  if(!playfg){
                   int delayloop = (int)(time_samp*20);
                   for(int i=0;i<=delayloop;i++){
                        delay(50);  // sec to msec
                        bartrend.select();
                        if(bartrend.statusund == 2
                              && bartrend.undactive == TRUE) break;
                        }
                  }
                  else bartrend.select();
                 }while(!(bartrend.statusund == 2
                        && bartrend.undactive == TRUE));
          break;
// plot two graph
case 1:bartrend.baroff();bartrend.baron();
          timebuf1 = timebuf2 = 0.0;
          do{
               if(playfg&&feof(recfile)) {
                bartrend.setact(stpvalfg);
                bartrend.show();
                outmesg(tunit1.posx+3,
                                    tunit1.posy+3," file end ");
                outmesg(tunit0.posx+3,
                                    tunit0.posy+3," file end ");
                break;
               }
               if(playfg){
                int i;
                fseek(recfile, curpos, SEEK_SET);
                fscanf(recfile," %i %f %f %f %f ",
                                        &i,&pvu,&cvu,&spu,&timeu);
                curpos = ftell(recfile);
                if(i!=1) {
                              outmesg(tunit0.posx+3,tunit0.posy+3,
                                                  " reading file error #0 ");
                              fclose(recfile);
                              playfg = FALSE;
                              break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    }
                        okflag = TRUE;
                    }else {getserialdata(pvu,cvu,spu,timeu,
                                                        trec0,okflag);
                        timebuf1 = timeu; // approximating 2 times time.
                        timeu = timebuf1+timebuf2+time_samp
                                                        +trend_cal[1];
                    }
                    if(okflag){
                     tunit0.graphplot(pvu,cvu,spu,timeu);
                     if(recordfg);
                        fprintf(recfile," %i %f %f %f %f ",
                                                    1,pvu,cvu,spu,timeu);
                    }
                    else outmesg(tunit0.gphposx1,tunit0.gphposy1,
                            " Serial attempt ");

                    if(playfg){
                     int i;
                     fseek(recfile, curpos, SEEK_SET);
                     fscanf(recfile," %i %f %f %f %f ",
                                        &i,&pvu,&cvu,&spu,&timeu);
                     curpos = ftell(recfile);
                     if(i!=2) {
                            outmesg(tunit1.posx+3,tunit1.posy+3,
                                        " reading file error #1 ");
                            fclose(recfile);
                            playfg = FALSE;
                            break;
                     }
                      okflag = TRUE;
                    }else { getserialdata(pvu,cvu,spu,timeu,
                                                        trec1,okflag);
                            timebuf2 = timeu;
                            timeu = timebuf1+timebuf2+time_samp
                                                        +trend_cal[1];
                    }
                    if(okflag){
                     tunit1.graphplot(pvu,cvu,spu,timeu);
                     if(recordfg);
                          fprintf(recfile," %i %f %f %f %f ",
                                                    2,pvu,cvu,spu,timeu);
                    }
                    else outmesg(tunit1.gphposx1,tunit1.gphposy1,
                                    " Serial attempt ");
                    if(!playfg){
                         int delayloop = (int)(time_samp*20);
                         for(int i=0;i<=delayloop;i++){
                              delay(50);  // sec to msec
                              bartrend.select();
                              if(bartrend.statusund == 2
                                && bartrend.undactive == TRUE) break;

                         }
                    }
                     else bartrend.select();
               }while(!(bartrend.statusund == 2
                         && bartrend.undactive == TRUE));
               break;
// plot four graph
case 2:bartrend.baroff();bartrend.baron();
               timebuf1 = timebuf2 = timebuf3 = timebuf4 = 0.0;
               do{
                    if(playfg&&feof(recfile)){
                     bartrend.setact(stpvalfg);
                     bartrend.show();
                     outmesg(tunit3.posx+3,
                                    tunit3.posy+3," file end ");
                                   outmesg(tunit2.posx+3,
                                    tunit2.posy+3," file end ");
                     outmesg(tunit1.posx+3,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        tunit1.posy+3," file end ");
    outmesg(tunit0.posx+3,
                        tunit0.posy+3," file end ");
    break;
  }
  //..
  if(playfg){
   int i;
   fseek(recfile, curpos, SEEK_SET);
   fscanf(recfile," %i %f %f %f %f ",
                              &i,&pvu,&cvu,&spu,&timeu);
   curpos = ftell(recfile);
        if(i!=1) {
              outmesg(tunit0.posx+3,tunit0.posy+3,
                              " reading file error #0 ");
              fclose(recfile);
              playfg = FALSE;
              break;
        }
        okflag = TRUE;
  }else {getserialdata(pvu,cvu,spu,timeu,
                                          trec0,okflag);
        timebuf1 = timeu;
        timeu =  timebuf1+timebuf2+timebuf3
                        +timebuf4+time_samp+trend_cal[2];
  }
  if(okflag){
   tunit0.graphplot(pvu,cvu,spu,timeu);
   if(recordfg);
        fprintf(recfile," %i %f %f %f %f ",
                                          1,pvu,cvu,spu,timeu);
  }
  else outmesg(tunit0.gphposx1,tunit0.gphposy1,
                        " Serial attempt ");

  if(playfg){
   int i;
        fseek(recfile, curpos, SEEK_SET);
        fscanf(recfile," %i %f %f %f %f ",
                              &i,&pvu,&cvu,&spu,&timeu);
        curpos = ftell(recfile);
        if(i!=2) {
              outmesg(tunit1.posx+3,tunit1.posy+3,
                              " reading file error #1 ");
              fclose(recfile);
              playfg = FALSE;
              break;
        }
       okflag = TRUE;
  }else { getserialdata(pvu,cvu,spu,timeu,
                                          trec1,okflag);
        timebuf2 = timeu;
        timeu = timebuf1+timebuf2+timebuf3
                        +timebuf4+time_samp+trend_cal[2];
  }
  if(okflag){
   tunit1.graphplot(pvu,cvu,spu,timeu);
   if(recordfg);
        fprintf(recfile," %i %f %f %f %f ",
                                  2,pvu,cvu,spu,timeu);
  }
  else outmesg(tunit1.gphposx1,tunit1.gphposy1,
                " Serial attempt ");

  if(playfg){
   int i;
   fseek(recfile, curpos, SEEK_SET);
   fscanf(recfile," %i %f %f %f %f ",
                        &i,&pvu,&cvu,&spu,&timeu);
   curpos = ftell(recfile);
   if(i!=3) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        outmesg(tunit2.posx+3,tunit2.posy+3,
                                        " reading file error #2");
                        fclose(recfile);
                        playfg = FALSE;
                        break;
                    }
                    okflag = TRUE;
                  }else {getserialdata(pvu,cvu,spu,timeu,
                                                        trec2,okflag);
                        timebuf3 = timeu;
                        timeu =  timebuf1+timebuf2+timebuf3
                                +timebuf4+time_samp+trend_cal[2];
                  }
                  if(okflag){
                    tunit2.graphplot(pvu,cvu,spu,timeu);
                    if(recordfg);
                     fprintf(recfile," %i %f %f %f %f %f ",
                                                3,pvu,cvu,spu,timeu);
                  }
                  else outmesg(tunit2.gphposx1,tunit2.gphposy1,
                                " Serial attempt ");

                  if(playfg){
                    int i;
                    fseek(recfile, curpos, SEEK_SET);
                    fscanf(recfile," %i %f %f %f %f ",
                                        &i,&pvu,&cvu,&spu,&timeu);
                    curpos = ftell(recfile);
                        if(i!=4) {
                            outmesg(tunit3.posx+3,tunit3.posy+3,
                                        " reading file error #3 ");
                            fclose(recfile);
                            playfg = FALSE;
                            break;
                        }
                        okflag = TRUE;
                    }else {getserialdata(pvu,cvu,spu,timeu,
                                                        trec3,okflag);
                        timebuf4 = timeu;
                        timeu =  timebuf1+timebuf2+timebuf3
                                +timebuf4+time_samp+trend_cal[2];
                    }
                    if(okflag){
                     tunit3.graphplot(pvu,cvu,spu,timeu);
                     if(recordfg);
                          fprintf(recfile," %i %f %f %f %f ",
                                                4,pvu,cvu,spu,timeu);
                    }
                    else outmesg(tunit3.gphposx1,tunit3.gphposy1,
                                " Serial attempt ");
                    if(!playfg){
                      int delayloop = (int)(time_samp*20);
                      for(int i=0;i<=delayloop;i++){
                            delay(50);  // sec to msec
                            bartrend.select();
                            if(bartrend.statusund == 2
                                && bartrend.undactive == TRUE) break;
                      }
                    }
                        else bartrend.select();
                }while(!(bartrend.statusund == 2
                            && bartrend.undactive == TRUE));
                break;
        }
        if(recordfg||playfg) bartrend.setact(plygrpfg);
        else bartrend.setact(rungrpfg);
        bartrend.show();
        break;
case 2:fclose(recfile);
        recordfg = FALSE;
        switch(statustgraph){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                case 2:outmesg(tunit3.posx+3,tunit3.posy+3,"      ");
                        outmesg(tunit2.posx+3,tunit2.posy+3,"      ");
                case 1:outmesg(tunit1.posx+3,tunit1.posy+3,"      ");
                case 0:outmesg(tunit0.posx+3,tunit0.posy+3,"      ");
                        break;
        }
        if(playfg){
                do22sub(ctrtag[trec0].tagname, ctrtag[trec0].engunits,
                                                        usemagnit, useratio,
                                                        ctrtag[trec0].maxmt,
                                                        ctrtag[trec0].lowscl,
                                                        ctrtag[trec0].upperscl,
                        ctrtag[trec1].tagname,
                                                        ctrtag[trec1].engunits,
                                                        usemagnit,
                                                        useratio,
                                                        ctrtag[trec1].maxmt,
                                                        ctrtag[trec1].lowscl,
                                                        ctrtag[trec1].upperscl,
                        ctrtag[trec2].tagname,
                                                        ctrtag[trec2].engunits,
                                                        usemagnit,
                                                        useratio,
                                                        ctrtag[trec2].maxmt,
                                                        ctrtag[trec2].lowscl,
                                                        ctrtag[trec2].upperscl,
                        ctrtag[trec3].tagname,
                                                        ctrtag[trec3].engunits,
                                                        usemagnit,
                                                        useratio,
                                                        ctrtag[trec3].maxmt,
                                                        ctrtag[trec3].lowscl,
                                                        ctrtag[trec3].upperscl);
        playfg = FALSE;
        }
        bartrend.setact(rungrpfg);  //trace
        bartrend.show();
        break;
case 3:switch(statustgraph){    //clear
        case 2:tunit3.reset_taxis();
                tunit3.cleargraph();
                tunit2.reset_taxis();
                tunit2.cleargraph();
        case 1:tunit1.reset_taxis();
                tunit1.cleargraph();
        case 0:tunit0.reset_taxis();
                tunit0.cleargraph();
                break;
        }
        bartrend.setact(rungrpfg);
        bartrend.show();
        break;

case 4://display_box
        do22sub5(trec0,trec1,trec2,trec3); //dsp_box
        bartrend.show();
        break;
case 5://save graph
        do22sub3(trec0,trec1,trec2,trec3);
        break;
case 6://load graph
        switch(statustgraph){
            case 0: do22sub4(usemagnit,useratio,trec0); //load 1gp
                    break;
            case 1: do22sub41(usemagnit,useratio,trec0,trec1);
                    break;
            case 2: do22sub42(usemagnit,useratio,trec0,trec1,
                                                        trec2,trec3);
                    break;
        }
        do22sub0(); //free datamem
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    bartrend.setact(loadgrpfg);
                    bartrend.show();
                    break;
                case 7: //record
                    char *recname;
                    str15 rfn;
                    recname =  do22sub60();
                    strcpy(rfn,recname);
                    if((recfile = fopen(rfn,"wb"))!=NULL){
                        do22sub6(recfile,trec0,trec1,trec2,trec3,recordfg);
                        bartrend.setact(recgrpfg);
                        bartrend.show();
                    }
                    break;
                case 8: //play
                    char *playname;
                    playname =  do22sub60();
                    strcpy(rfn,playname);
                    if((recfile = fopen(rfn,"rb"))==NULL)
                        subp2.massage("  FILE not found. ",YELLOW);
                    else {do22sub7( recfile,usemagnit, useratio,curpos);
                            playfg = TRUE;
                            }
                    bartrend.setact(plygrpfg);
                    bartrend.show();
                    break;
            }
        }while(exitdo == FALSE);
        do22sub0();
}

void  do22sub(str15 t0,str15 u0,float m0,float r0,float x0,int l0,int h0,
              str15 t1,str15 u1,float m1,float r1,float x1,int l1,int h1,
              str15 t2,str15 u2,float m2,float r2,float x2,int l2,int h2,
              str15 t3,str15 u3,float m3,float r3,float x3,int l3,int h3){


   switch(statustgraph) {
            case 2:
                tunit3.reset_taxis();
                tunit3.trendcreat(t3,u3,m3,r3,x3,l3,h3);

                tunit2.reset_taxis();
                tunit2.trendcreat(t2,u2,m2,r2,x2,l2,h2);
        case 1:
                tunit1.reset_taxis();
                tunit1.trendcreat(t1,u1,m1,r1,x1,l1,h1);
        case 0:
                tunit0.reset_taxis();
                tunit0.trendcreat(t0,u0,m0,r0,x0,l0,h0);
                break;
       }

}
void do22sub0(void){
        tunit0.freedatamem();
        tunit1.freedatamem();
        tunit2.freedatamem();
        tunit3.freedatamem();
}

void do22sub1(FILE *inf){

        str15 buff;

        // clear view
        umouse.show(FALSE);
        blankscr();
        umouse.show(TRUE);
        setviewport(0,0,getmaxx(),getmaxy(),1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        // show status
        setcolor(WHITE);
        rectangle(1,25,getmaxx()-1,310);
        setcolor(CYAN);
        settextstyle(DEFAULT_FONT,HORIZ_DIR,2);
        outtextxy(150,35," Status of controller ");
        setcolor(YELLOW);
        settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
        outtextxy(200,55," TAG file name is : ");
        outtextxy(360,55,gfilename);
        setfillstyle(SOLID_FILL,LIGHTGREEN);
        bar(1,70,getmaxx()-1,85);
        setcolor(DARKGRAY);
        outtextxy(3,75,
        "No# tag  time(min) description      ENG/Units  HRange LRange");
        for(int i = 1;i< MAXCTL;i++)
              fread(&ctrtag[i],sizeof(taglist),1,inf);
        setcolor(LIGHTRED);
        for(i = 1;i<MAXCTL;i++) {
              sprintf(buff,"%2d",i);
              outtextxy(UPOS+4  ,YTEXT + (i*16),buff);
              outtextxy(TAGPOS+4,YTEXT + (i*16),ctrtag[i].tagname);
              outtextxy(DISPOS+10,YTEXT + (i*16),ctrtag[i].discript);
              outtextxy(ENGPOS+4,YTEXT + (i*16),ctrtag[i].engunits);
              sprintf(buff,"%6.2f",ctrtag[i].maxmt);
              outtextxy(SAMPOS+4,YTEXT + (i*16),buff);
              sprintf(buff,"%6i",ctrtag[i].upperscl);
              outtextxy(UPPOS+4,YTEXT + (i*16),buff);
              sprintf(buff,"%6i",ctrtag[i].lowscl);
              outtextxy(LWPOS+4,YTEXT + (i*16),buff);
        }

}

void do22sub2(int &trec0,int &trec1, int &trec2,int &trec3,
                        float &usemagnit ,float &useratio) {

        setcolor(YELLOW);
        outtextxy(100,350,
                        "** Which of No. of controllor is going to plot?<1..13> **");
        setcolor(CYAN);
        // show graph of trendrecorder
        switch(statustgraph) {
              case 0:
                        trec0 = getunit(200,370);
                        usemagnit = 3.0  ;useratio = 0.52 ;
                        break;
              case 1:
                        outtextxy(100,370," graph #1 ");
                        trec0 = getunit(200,370);
                        setviewport(0,0,getmaxx(),getmaxy(),1);
                        setcolor(CYAN);
                        outtextxy(100,390," graph #2 ");
                        trec1 = getunit(200,390);
                        usemagnit = 2.7  ;useratio = 0.32  ;
                        break;
              case 2:
                        outtextxy(100,370," graph #1 ");
                        trec0 = getunit(200,370);
                        setviewport(0,0,getmaxx(),getmaxy(),1);
                        setcolor(CYAN);
                        outtextxy(100,390," graph #2 ");
                        trec1 = getunit(200,390);
                        setviewport(0,0,getmaxx(),getmaxy(),1);
                        setcolor(CYAN);
                        outtextxy(100,410," graph #3 ");
                        trec2 = getunit(200,410);
                        setviewport(0,0,getmaxx(),getmaxy(),1);
                        setcolor(CYAN);
                        outtextxy(100,430," graph #4 ");
                        trec3 = getunit(200,430);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        usemagnit = 1.55 ;useratio = 0.56  ;
                        break;

        }

}

void do22sub3(int trec0,int trec1,int trec2,int trec3){ // sav gph

  FILE *out;
  char *inname;
  str15 uname;
  str15 filen = "";
  float pvbf,cvbf,spbf,samtbf;
  const namefm namefile={" Put save file name : < _              >",
                                        "                 _          .gph "};

        inname = getname(100,getmaxy()-70,namefile);
        strcpy(uname,inname);
        strcpy(filen,uname);       // data in GOLBAL
        switch(statustgraph){
                case 0: strcat(filen,".1gr");break;
                case 1: strcat(filen,".2g1");break;
                case 2: strcat(filen,".4g1");break;
        }
        out = fopen(filen,"w");
        fprintf(out,"%s %s %f %i %i %i %f",ctrtag[trec0].tagname,
                                                ctrtag[trec0].engunits,
                                                tunit0.maxtime,
                                                tunit0.lowlimt,
                                                tunit0.hilimt,
                                                tunit0.timecount,
                                                tunit0.taxis);
        for(int i = 0;i <= tunit0.timecount-1 ;i++)
                if(tunit0.findsav(i,pvbf,cvbf,spbf,samtbf))
                        fprintf(out," %i %f %f %f %f ",i,pvbf,cvbf,spbf,samtbf);
        fclose(out);

        //..
        if(statustgraph >= 1){
                strcpy(filen,uname);
                switch(statustgraph){
                        case 0: break;
                        case 1: strcat(filen,".2g2");break;
                        case 2: strcat(filen,".4g2");break;
                }
                out = fopen(filen,"w");
                fprintf(out,"%s %s %f %i %i %i %f",ctrtag[trec1].tagname,
                                                        ctrtag[trec1].engunits,
                                                        tunit1.maxtime,
                                                        tunit1.lowlimt,
                                                        tunit1.hilimt,
                                                        tunit1.timecount,
                                                        tunit1.taxis);
                for(int i = 0;i <= tunit1.timecount-1 ;i++)
                        if(tunit1.findsav(i,pvbf,cvbf,spbf,samtbf))
                                fprintf(out," %i %f %f %f %f ",i,pvbf,cvbf,spbf,samtbf);
                fclose(out);
        }
        //..
        if(statustgraph >=2 ){
                strcpy(filen,uname);
                strcat(filen,".4g3");
                out = fopen(filen,"w");
                fprintf(out,"%s %s %f %i %i %i %f",ctrtag[trec2].tagname,
                                                        ctrtag[trec2].engunits,
                                                        tunit2.maxtime,
                                                        tunit2.lowlimt,
                                                        tunit2.hilimt,
                                                        tunit2.timecount,
                                                        tunit2.taxis);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            for(int i = 0;i <= tunit2.timecount-1 ;i++)
                if(tunit2.findsav(i,pvbf,cvbf,spbf,samtbf))
                    fprintf(out," %i %f %f %f %f ",i,pvbf,cvbf,spbf,samtbf);
            fclose(out);
            //..
            strcpy(filen,uname);
            strcat(filen,".4g4");
            out = fopen(filen,"w");
            fprintf(out,"%s %s %f %i %i %i %f",ctrtag[trec3].tagname,
                                                ctrtag[trec3].engunits,
                                                tunit3.maxtime,
                                                tunit3.lowlimt,
                                                tunit3.hilimt,
                                                tunit3.timecount,
                                                tunit3.taxis);
            for(i = 0;i <= tunit3.timecount-1 ;i++)
             if(tunit3.findsav(i,pvbf,cvbf,spbf,samtbf))
                    fprintf(out," %i %f %f %f %f ",i,pvbf,cvbf,spbf,samtbf);
            fclose(out);
        }

        subp2.massage(" Save graph finished. ",LIGHTRED);
}

void do22sub4(float usemagnit,float useratio,int trec0) { //load 1grp

 extern void do22sub5(int,int,int,int);
 extern void outmesg(int,int,char *);

 const undertxt undload = {DARKGRAY,
                           WHITE,
                           YELLOW,
                           BLACK,
                           2,
                           " EXIT ",
                           " VALUE "};


FILE *in,*temp;
char *inname;
str25 buff = " File : ";
str15 filen,tagbuf,engbuf;
float pvbf,cvbf,spbf,samtbf;
int  j;
const namefm namefile={" Put load file name : < _          >",
                       "                  _          .gph "};

    underbar barload(undload);
    barload.setact(bartype1);   //all TRUE

    inname = getname(100,getmaxy()-70,namefile);
    strcpy(filen,inname);
    strcat(filen,".1gr");
    if((in = fopen(filen,"r")) == NULL)
        {
         subp2.massage(" file not found. ",LIGHTRED);
         fclose(in);
         return;
        }
         // save in temp
   temp = fopen("TEMP.$$$","w");
   fprintf(temp,"%f %i %i %i %f ",tunit0.maxtime,
                                  tunit0.lowlimt,
                                  tunit0.hilimt,
                                  tunit0.timecount,tunit0.taxis);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
for(int i = 0;i <= tunit0.timecount-1 ;i++)
        if(tunit0.findsav(i,pvbf,cvbf,spbf,samtbf))
                fprintf(temp," %i %f %f %f %f ",i,pvbf,cvbf,spbf,samtbf);
fclose(temp);

//load file
fscanf(in,"%s %s %f %i %i %i %f",&tagbuf,
                                                &engbuf,
                                                &tunit0.maxtime,
                                                &tunit0.lowlimt,
                                                &tunit0.hilimt,
                                                &tunit0.timecount,
                                                &tunit0.taxis);
tunit0.cleargraph();
tunit0.trendcreat(tagbuf,
                        engbuf,
                        usemagnit,
                        useratio,
                        tunit0.maxtime,
                        tunit0.lowlimt,
                        tunit0.hilimt);
strcat(buff,filen);
outmesg(tunit0.posx+3,tunit0.posy+3,buff);
i = 0;
tunit0.reset_timecountbox();
while(!feof(in)){
        fscanf(in," %i %f %f %f %f ",&j,&pvbf,&cvbf,&spbf,&samtbf);
        tunit0.graphplot(pvbf,cvbf,spbf,samtbf);
        if(i != j){
                subp2.massage(" error file reading .",LIGHTRED);
                break;
        }
        i++;
}
fclose(in);

        //display value
do{
        barload.show();
        barload.baroff();
        barload.baron();
        do{
                barload.select();
        }while(barload.exith == FALSE);
        switch(barload.statusund){
                case 0:break;
                case 1:tunit0.value_box(10,getmaxy()-70,tagbuf,engbuf);
                        break;
        }
}while(barload.statusund != 0);
//reteive old
temp = fopen("TEMP.$$$","r");
fscanf(temp,"%f %i %i %i %f ",&tunit0.maxtime,
                                                &tunit0.lowlimt,
                                                &tunit0.hilimt,
                                                &tunit0.timecount,
                                                &tunit0.taxis);
tunit0.trendcreat(ctrtag[trec0].tagname,
                        ctrtag[trec0].engunits,
                        usemagnit,
                        useratio,
                        ctrtag[trec0].maxmt,
                        ctrtag[trec0].lowscl,
                        ctrtag[trec0].upperscl);
tunit0.cleargraph();
i=0;
tunit0.reset_timecountbox();
while(!feof(temp)){
        fscanf(temp," %i %f %f %f %f ",&j,&pvbf,&cvbf,&spbf,&samtbf);
        tunit0.graphplot(pvbf,cvbf,spbf,samtbf);
        if(i != j){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    break;
                }
                i++;
        }
        fclose(temp);

}

void do22sub41(float usemagnit,float useratio,int trec0,int trec1){ //load 2gp


  extern void outmesg(int,int,char *);

 const undertxt undload = {DARKGRAY,
                                                WHITE,
                                                YELLOW,
                                                BLACK,
                                                3,
                                                " EXIT ",
                                                " VAL #1 ",
                                                " VAL #2 "};



 FILE *in,*temp;
 char *inname;
 str25 buff = " File : ";
 str15 filen,tagbuf0,engbuf0,tagbuf1,engbuf1;
 str15 uname = "";
 float pvbf,cvbf,spbf,samtbf;
 int  i,j;
 const namefm namefile={" Put load file name : < _                    >",
                                  "                                  .gph "};

      inname = getname(100,getmaxy()-70,namefile);
      strcpy(uname,inname);

       // save in temp
       temp = fopen("TEMP21.$$$","w");
       fprintf(temp,"%f %i %i %i %f ",tunit0.maxtime,
                                                tunit0.lowlimt,
                                                tunit0.hilimt,
                                                tunit0.timecount,
                                                tunit0.taxis);
       for(i = 0;i <= tunit0.timecount-1 ;i++)
             if(tunit0.findsav(i,pvbf,cvbf,spbf,samtbf))
                    fprintf(temp," %i %f %f %f %f ",i,pvbf,cvbf,spbf,samtbf);
       fclose(temp);

            // save in temp
       temp = fopen("TEMP22.$$$","w");
       fprintf(temp,"%f %i %i %i %f ",tunit1.maxtime,
                                                tunit1.lowlimt,
                                                tunit1.hilimt,
                                                tunit1.timecount,
                                                tunit1.taxis);
       for(i = 0;i <= tunit1.timecount-1 ;i++)
             if(tunit1.findsav(i,pvbf,cvbf,spbf,samtbf))
                    fprintf(temp," %i %f %f %f %f ",i,pvbf,cvbf,spbf,samtbf);
       fclose(temp);

           //load file
       strcpy(filen,uname);
       strcat(filen,".2g1");
       if((in = fopen(filen,"r")) == NULL){
             subp2.massage(" file not found. ",LIGHTRED);
             fclose(in);
             return;
       }
            //load file
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
fscanf(in,"%s %s %f %i %i %i %f",&tagbuf0,
                                                &engbuf0,
                                                &tunit0.maxtime,
                                                &tunit0.lowlimt,
                                                &tunit0.hilimt,
                                                &tunit0.timecount,
                                                &tunit0.taxis);
tunit0.cleargraph();
tunit0.trendcreat(tagbuf0,
                        engbuf0,
                        usemagnit,
                        usecratio,
                        tunit0.maxtime,
                        tunit0.lowlimt,
                        tunit0.hilimt);
strcat(buff,filen);
outmesg(tunit0.posx+3,tunit0.posy+3,buff);

//plot load
i = 0;
tunit0.reset_timecountbox();
while(!feof(in)){
        fscanf(in," %i %f %f %f %f ",&j,&pvbf,&cvbf,&spbf,&samtbf);
        tunit0.graphplot(pvbf,cvbf,spbf,samtbf);
        if(i != j){
                subp2.massage(" error file reading .",LIGHTRED);
                break;
        }
        i++;
}
fclose(in);
//..
strcpy(filen,uname);
strcat(filen,".2g2");
if((in = fopen(filen,"r")) == NULL){
        subp2.massage(" file not found. ",LIGHTRED);
        fclose(in);
        return;
}
 //load file
fscanf(in,"%s %s %f %i %i %i %f",&tagbuf1,
                                                &engbuf1,
                                                &tunit1.maxtime,
                                                &tunit1.lowlimt,
                                                &tunit1.hilimt,
                                                &tunit1.timecount,
                                                &tunit1.taxis);
tunit1.cleargraph();
tunit1.trendcreat(tagbuf1,
                        engbuf1,
                        usemagnit,
                        usecratio,
                        tunit1.maxtime,
                        tunit1.lowlimt,
                        tunit1.hilimt);
strcpy(buff," File : ");
strcat(buff,filen);
outmesg(tunit1.posx+3,tunit1.posy+3,buff);

//plot load
i = 0;
tunit1.reset_timecountbox();
while(!feof(in)){
        fscanf(in," %i %f %f %f %f ",&j,&pvbf,&cvbf,&spbf,&samtbf);
        tunit1.graphplot(pvbf,cvbf,spbf,samtbf);
        if(i != j){
                subp2.massage(" error file reading .",LIGHTRED);
                break;
        }
        i++;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
fclose(in);

 //display value
underbar barload(undload);
barload.setact(bartype1);   //all TRUE
do{
        barload.show();
        barload.baroff();
        barload.baron();
        do{
              barload.select();
        }while(barload.exith == FALSE);
        switch(barload.statusund){
              case 0:break;
              case 1:tunit0.value_box(10,getmaxy()-70,tagbuf0,engbuf0);
                          break;
              case 2:tunit1.value_box(10,getmaxy()-70,tagbuf1,engbuf1);
                           break;
        }
}while(barload.statusund != 0);

//reteive old
 do22sub0();
 temp = fopen("TEMP21.$$$","r");
 fscanf(temp,"%f %i %i %i %f ",&tunit0.maxtime,
                                              &tunit0.lowlimt,
                                              &tunit0.hilimt,
                                              &tunit0.timecount,
                                              &tunit0.taxis);
 tunit0.trendcreat(ctrtag[trec0].tagname,
                          ctrtag[trec0].engunits,
                          usemagnit,
                          useratio,
                          ctrtag[trec0].maxmt,
                          ctrtag[trec0].lowscl,
                          ctrtag[trec0].upperscl);
 tunit0.cleargraph();
 i=0;
 tunit0.reset_timecountbox();
 while(!feof(temp)){
        fscanf(temp," %i %f %f %f %f ",&j,&pvbf,&cvbf,&spbf,&samtbf);
        tunit0.graphplot(pvbf,cvbf,spbf,samtbf);
        if(i != j){
              subp2.massage(" error file reading .",LIGHTRED);
              break;
        }
        i++;
 }
 fclose(temp);
          //..
 temp = fopen("TEMP22.$$$","r");
 fscanf(temp,"%f %i %i %i %f ",&tunit1.maxtime,
                                              &tunit1.lowlimt,
                                              &tunit1.hilimt,
                                              &tunit1.timecount,
                                              &tunit1.taxis);
 tunit1.trendcreat(ctrtag[trec1].tagname,
                          ctrtag[trec1].engunits,
                          usemagnit,
                          useratio,
                          ctrtag[trec1].maxmt,
                          ctrtag[trec1].lowscl,
                          ctrtag[trec1].upperscl);
 tunit1.cleargraph();
 i=0;
 tunit1.reset_timecountbox();
 while(!feof(temp)){
        fscanf(temp," %i %f %f %f %f ",&j,&pvbf,&cvbf,&spbf,&samtbf);
        tunit1.graphplot(pvbf,cvbf,spbf,samtbf);
        if(i != j){
              subp2.massage(" error file reading .",LIGHTRED);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        break;
                    }
                    i++;
                }
                fclose(temp);
}

void do22sub42(float usemagnit,float useratio,int trec0,int trec1, // 4grp
                                                        int trec2,int trec3){

 extern void outmesg(int,int,char *);

 const undertxt undload = {DARKGRAY,
                           WHITE,
                           YELLOW,
                           BLACK,
                           5,
                           " EXIT ",
                           " VAL #1 ",
                           " VAL #2 ",
                           " VAL #3 ",
                           " VAL #4 "};


 FILE *in,*temp;
 char *inname;
 str25 buff = " File : ";
 str15 filen;
 str15 tagbuf0,engbuf0,tagbuf1,engbuf1;
 str15 tagbuf2,engbuf2,tagbuf3,engbuf3;
 str15 uname = "";
 float pvbf,cvbf,spbf,samtbf;
 int  i,j;
 const namefm namefile={" Put load file name : < _          >",
                        "                      _          .gph "};


     underbar barload(undload);
     barload.setact(bartype1);   //all TRUE

     inname = getname(100,getmaxy()-70,namefile);
     strcpy(uname,inname);

     // save in temp
     temp = fopen("TEMP41.$$$","w");
     fprintf(temp,"%f %i %i %i %f ",tunit0.maxtime,
                                    tunit0.lowlimt,
                                    tunit0.hilimt,
                                    tunit0.timecount,
                                    tunit0.taxis);
     for(i = 0;i <= tunit0.timecount-1 ;i++)
          if(tunit0.findsav(i,pvbf,cvbf,spbf,samtbf))
               fprintf(temp," %i %f %f %f %f ",i,pvbf,cvbf,spbf,samtbf);
     fclose(temp);

     // save in temp
     temp = fopen("TEMP42.$$$","w");
     fprintf(temp,"%f %i %i %i %f ",tunit1.maxtime,
                                    tunit1.lowlimt,
                                    tunit1.hilimt,
                                    tunit1.timecount,
                                    tunit1.taxis);
     for(i = 0;i <= tunit1.timecount-1 ;i++)
          if(tunit1.findsav(i,pvbf,cvbf,spbf,samtbf))
               fprintf(temp," %i %f %f %f %f ",i,pvbf,cvbf,spbf,samtbf);
     fclose(temp);

     // save in temp
     temp = fopen("TEMP43.$$$","w");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
temp = fopen("TEMP43.$$$","w");
fprintf(temp,"%f %i %i %i %f ",tunit2.maxtime,
                                                tunit2.lowlimt,
                                                tunit2.hilimt,
                                                tunit2.timecount,
                                                tunit2.taxis);
for(i = 0;i <= tunit2.timecount-1 ;i++)
        if(tunit2.findsav(i,pvbf,cvbf,spbf,samtbf))
                fprintf(temp," %i %f %f %f %f ",i,pvbf,cvbf,spbf,samtbf);
fclose(temp);

// save in temp
temp = fopen("TEMP44.$$$","w");
fprintf(temp,"%f %i %i %i %f ",tunit3.maxtime,
                                                tunit3.lowlimt,
                                                tunit3.hilimt,
                                                tunit3.timecount,
                                                tunit3.taxis);
for(i = 0;i <= tunit3.timecount-1 ;i++)
        if(tunit3.findsav(i,pvbf,cvbf,spbf,samtbf))
                fprintf(temp," %i %f %f %f %f ",i,pvbf,cvbf,spbf,samtbf);
fclose(temp);

//load file
strcpy(filen,uname);
strcat(filen,".4g1");
if((in = fopen(filen,"r")) == NULL){
        subp2.massage(" file not found. ",LIGHTRED);
        fclose(in);
        return;
}
//load file
 fscanf(in,"%s %s %f %i %i %i %f",&tagbuf0,
                                                &engbuf0,
                                                &tunit0.maxtime,
                                                &tunit0.lowlimt,
                                                &tunit0.hilimt,
                                                &tunit0.timecount,
                                                &tunit0.taxis);

 tunit0.cleargraph();
 tunit0.trendcreat(tagbuf0,
                                        engbuf0,
                                        usemagnit,
                                        useratio,
                                        tunit0.maxtime,
                                        tunit0.lowlimt,
                                        tunit0.hilimt);
strcat(buff,filen);
outmesg(tunit0.posx+3,tunit0.posy+3,buff);

//plot load
i = 0;
tunit0.reset_timecountbox();
while(!feof(in)){
        fscanf(in," %i %f %f %f %f ",&j,&pvbf,&cvbf,&spbf,&samtbf);
        tunit0.graphplot(pvbf,cvbf,spbf,samtbf);
        if(i != j){
                subp2.massage(" error file reading .",LIGHTRED);
                break;
        }
        i++;
}
fclose(in);

//..
strcpy(filen,uname);
strcat(filen,".4g2");
if((in = fopen(filen,"r")) == NULL){
        subp2.massage(" file not found. ",LIGHTRED);
        fclose(in);
        return;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
//load file
fscanf(in,"%s %s %f %i %i %i %f",&tagbuf1,
                                                &engbuf1,
                                                &tunit1.maxtime,
                                                &tunit1.lowlimt,
                                                &tunit1.hilimt,
                                                &tunit1.timecount,
                                                &tunit1.taxis);
tunit1.cleargraph();
tunit1.trendcreat(tagbuf1,
                                        engbuf1,
                                        usemagnit,
                                        useratio,
                                        tunit1.maxtime,
                                        tunit1.lowlimt,
                                        tunit1.hilimt);
strcpy(buff," File : ");
strcat(buff,filen);
outmesg(tunit1.posx+3,tunit1.posy+3,buff);

 //plot load
i = 0;
tunit1.reset_timecountbox();
while(!feof(in)){
      fscanf(in," %i %f %f %f %f ",&j,&pvbf,&cvbf,&spbf,&samtbf);
      tunit1.graphplot(pvbf,cvbf,spbf,samtbf);
      if(i != j){
            subp2.massage(" error file reading .",LIGHTRED);
            break;
      }
      i++;
}
fclose(in);

//..
strcpy(filen,uname);
strcat(filen,".4g3");
if((in = fopen(filen,"r")) == NULL){
            subp2.massage(" file not found. ",LIGHTRED);
            fclose(in);
            return;
}
                  //load file
fscanf(in,"%s %s %f %i %i %i %f",&tagbuf2,
                                                &engbuf2,
                                                &tunit2.maxtime,
                                                &tunit2.lowlimt,
                                                &tunit2.hilimt,
                                                &tunit2.timecount,
                                                &tunit2.taxis);
tunit2.cleargraph();
tunit2.trendcreat(tagbuf2,
                                        engbuf2,
                                        usemagnit,
                                        useratio,
                                        tunit2.maxtime,
                                        tunit2.lowlimt,
                                        tunit2.hilimt);
strcpy(buff," File : ");
strcat(buff,filen);
outmesg(tunit2.posx+3,tunit2.posy+3,buff);

//plot load
i = 0;
tunit2.reset_timecountbox();
while(!feof(in)){
      fscanf(in," %i %f %f %f %f ",&j,&pvbf,&cvbf,&spbf,&samtbf);
      tunit2.graphplot(pvbf,cvbf,spbf,samtbf);
      if(i != j){
            subp2.massage(" error file reading .",LIGHTRED);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                break;
        }
        i++;
}
fclose(in);

//..
strcpy(filen,uname);
strcat(filen,".4g4");
if((in = fopen(filen,"r")) == NULL){
        subp2.massage(" file not found. ",LIGHTRED);
        fclose(in);
        return;
}
                //load file
fscanf(in,"%s %s %f %i %i %i %f",&tagbuf3,
                                        &engbuf3,
                                        &tunit3.maxtime,
                                        &tunit3.lowlimt,
                                        &tunit3.hilimt,
                                        &tunit3.timecount,
                                        &tunit3.taxis);
tunit3.cleargraph();
tunit3.trendcreat(tagbuf3,
                                        engbuf3,
                                        usemagnit,
                                        useratio,
                                        tunit3.maxtime,
                                        tunit3.lowlimt,
                                        tunit3.hilimt);
strcpy(buff," File : ");
strcat(buff,filen);
outmesg(tunit3.posx+3,tunit3.posy+3,buff);

//plot load
i = 0;
tunit3.reset_timecountbox();
while(!feof(in)){
        fscanf(in," %i %f %f %f %f ",&j,&pvbf,&cvbf, &spbf,&samtbf);
        tunit3.graphplot(pvbf,cvbf,spbf,samtbf);
        if(i != j){
                subp2.massage(" error file reading .",LIGHTRED);
                break;
        }
        i++;
}
fclose(in);

//display value
do{
        barload.show();
        barload.baroff();
        barload.baron();
        do{
                barload.select();
        }while(barload.exith == FALSE);
        switch(barload.statusund){
                case 0:break;
                case 1:tunit0.value_box(10,getmaxy()-70,tagbuf0,engbuf0);
                        break;
                case 2:tunit1.value_box(10,getmaxy()-70,tagbuf1,engbuf1);
                        break;
                case 3:tunit2.value_box(10,getmaxy()-70,tagbuf2,engbuf2);
                        break;
                case 4:tunit3.value_box(10,getmaxy()-70,tagbuf3,engbuf3);
                        break;
        }
}while(barload.statusund != 0);
//reteive old
do22sub0();
```

เอกสารนี้มีการที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
temp = fopen("TEMP41.$$$","r");
fscanf(temp,"%f %i %i %i %f ",&tunit0.maxtime,
                                        &tunit0.lowlimt,
                                        &tunit0.hilimt,
                                        &tunit0.timecount,
                                        &tunit0.taxis);
tunit0.trendcreat(ctrtag[trec0].tagname,
                                    ctrtag[trec0].engunits,
                                    usemagnit,
                                    useratio,
                                    ctrtag[trec0].maxmt,
                                    ctrtag[trec0].lowscl,
                                    ctrtag[trec0].upperscl);
tunit0.cleargraph();
i=0;
tunit0.reset_timecountbox();
while(!feof(temp)){
        fscanf(temp," %i %f %f %f %f ",&j,&pvbf,&cvbf,&spbf,&samtbf);
        tunit0.graphplot(pvbf,cvbf,spbf,samtbf);
        if(i != j){
                subp2.massage(" error file reading .",LIGHTRED);
                break;
        }
        i++;
}
fclose(temp);

//..
temp = fopen("TEMP42.$$$","r");
fscanf(temp,"%f %i %i %i %f ",&tunit1.maxtime,
                                        &tunit1.lowlimt,
                                        &tunit1.hilimt,
                                        &tunit1.timecount,
                                        &tunit1.taxis);
tunit1.trendcreat(ctrtag[trec1].tagname,
                                        ctrtag[trec1].engunits,
                                        usemagnit,
                                        useratio,
                                        ctrtag[trec1].maxmt,
                                        ctrtag[trec1].lowscl,
                                        ctrtag[trec1].upperscl);
tunit1.cleargraph();
i=0;
tunit1.reset_timecountbox();
while(!feof(temp)){
        fscanf(temp," %i %f %f %f %f ",&j,&pvbf,&cvbf,&spbf,&samtbf);
        tunit1.graphplot(pvbf,cvbf,spbf,samtbf);
        if(i != j){
                subp2.massage(" error file reading .",LIGHTRED);
                break;
        }
        i++;
}
fclose(temp);

//..
temp = fopen("TEMP43.$$$","r");
fscanf(temp,"%f %i %i %i %f ",&tunit2.maxtime,
                                        &tunit2.lowlimt,
                                        &tunit2.hilimt,
                                        &tunit2.timecount,
                                        &tunit2.taxis);
tunit2.trendcreat(ctrtag[trec2].tagname,
                                        ctrtag[trec2].engunits,
                                        usemagnit,
                                        useratio,
                                        ctrtag[trec2].maxmt,
                                        ctrtag[trec2].lowscl,
                                        ctrtag[trec2].upperscl);
tunit2.cleargraph();
i=0;
```

เอกสารนี้ใช้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            tunit2.reset_timecountbox();
            while(!feof(temp)){
                  fscanf(temp," %i %f %f %f %f ",&j,&pvbf,&cvbf,&spbf,&samtbf);
                  tunit2.graphplot(pvbf,cvbf,spbf,samtbf);
                  if(i != j){
                        subp2.massage(" error file reading .",LIGHTRED);
                        break;
                  }
                  i++;
      }
            fclose(temp);

            //..
            temp = fopen("TEMP44.$$$","r");
            fscanf(temp,"%f %i %i %i %f ",&tunit3.maxtime,
                                                &tunit3.lowlimt,
                                                &tunit3.hilimt,
                                                &tunit3.timecount,
                                                &tunit3.taxis);
            tunit3.trendcreat(ctrtag[trec3].tagname,
                                                ctrtag[trec3].engunits,
                                                usemagnit,
                                                useratio,
                                                ctrtag[trec3].maxmt,
                                                ctrtag[trec3].lowscl,
                                                ctrtag[trec3].upperscl);
            tunit3.cleargraph();
            i=0;
            tunit3.reset_timecountbox();
            while(!feof(temp)){
                  fscanf(temp," %i %f %f %f %f ",&j,&pvbf,&cvbf,&spbf,&samtbf);
                  tunit3.graphplot(pvbf,cvbf,spbf,samtbf);
                  if(i != j){
                        subp2.massage(" error file reading .",LIGHTRED);
                        break;
                  }
                  i++;
            };
            fclose(temp);
}

void do22sub5(int trec0,int trec1,int trec2,int trec3){  // value box

      const undertxt undload = {DARKGRAY,
                                WHITE,
                                YELLOW,
                                BLACK,
                                3,
                                " EXIT ",
                                " VAL #1 ",
                                " VAL #2 "};

      const undertxt undload1 = {DARKGRAY,
                                WHITE,
                                YELLOW,
                                BLACK,
                                5,
                                " EXIT ",
                                " VAL #1 ",
                                " VAL #2 ",
                                " VAL #3 ",
                                " VAL #4 "};

      //display value box

      switch(statustgraph){
            case 0:tunit0.value_box(10,getmaxy()-70,
                                                ctrtag[trec0].tagname,
                                                ctrtag[trec0].engunits);
                                    break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        case 1: underbar barload(undload);
                    barload.setact(bartype1);   //all TRUE
                do{

                    barload.show();
                    barload.baroff();
                    barload.baron();
                    do{
                        barload.select();
                    }while(barload.exith == FALSE);
                    switch(barload.statusund){
                        case 0:break;
                        case 1:tunit0.value_box(10,getmaxy()-70,
                                                        ctrtag[trec0].tagname,
                                                    ctrtag[trec0].engunits);
                            break;
                        case 2:tunit1.value_box(10,getmaxy()-70,
                                                        ctrtag[trec1].tagname,
                                                        ctrtag[trec1].engunits);
                            break;
                    }
                 }while(barload.statusund != 0);
                break;

    case 2:     underbar barload1(undload1);
                    barload1.setact(bartype1);   //all TRUE
                do{

                    barload1.show();
                    barload1.baroff();
                    barload1.baron();
                    do{
                        barload1.select();
                    }while(barload1.exith == FALSE);
                    switch(barload1.statusund){
                        case 0: break;
                        case 1: tunit0.value_box(10,getmaxy()-70,
                                                        ctrtag[trec0].tagname,
                                                    ctrtag[trec0].engunits);
                                break;
                        case 2:     tunit1.value_box(10,getmaxy()-70,
                                                        ctrtag[trec1].tagname,
                                                        ctrtag[trec1].engunits);
                                break;
                        case 3:     tunit2.value_box(10,getmaxy()-70,
                                                        ctrtag[trec2].tagname,
                                                        ctrtag[trec2].engunits);
                                break;
                        case 4:     tunit3.value_box(10,getmaxy()-70,
                                                        ctrtag[trec3].tagname,
                                                        ctrtag[trec3].engunits);
                                 break;
                    }
                }while(barload1.statusund != 0);
                break;
        }

}
char *do22sub60(void){

  char *inname;
  str15 uname,filen;

  const namefm namefile={" Put your file name : < _              >",
                                "                  _          .rec "};


      inname = getname(100,getmaxy()-70,namefile);
      strcpy(uname,inname);
      strcpy(filen,uname);
      switch(statustgraph){
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        case 0: strcat(filen,".1rc");break;
        case 1: strcat(filen,".2rc");break;
        case 2: strcat(filen,".4rc");break;
      }
      return(filen);
}

//record
void do22sub6(FILE *recfile,int trec0,int trec1,int trec2,int trec3,
                                                        boolean &recordfg){

      extern void outmesg(int,int,char *);

      recordfg = TRUE;
      switch(statustgraph){
            case 0:
                  fprintf(recfile,"%s %s %f %i %i",
                                                ctrtag[trec0].tagname,
                                                ctrtag[trec0].engunits,
                                                tunit0.maxtime,
                                                tunit0.lowlimt,
                                                tunit0.hilimt);
                  outmesg(tunit0.posx+3,tunit0.posy+3," record ");
                  break;
            case 1:
                  fprintf(recfile,"%s %s %f %i %i",
                                                ctrtag[trec0].tagname,
                                                ctrtag[trec0].engunits,
                                                tunit0.maxtime,
                                                tunit0.lowlimt,
                                                tunit0.hilimt);

                  fprintf(recfile,"%s %s %f %i %i",
                                                ctrtag[trec1].tagname,
                                                ctrtag[trec1].engunits,
                                                tunit1.maxtime,
                                                tunit1.lowlimt,
                                                tunit1.hilimt);
                  outmesg(tunit0.posx+3,tunit0.posy+3," record ");
                  outmesg(tunit1.posx+3,tunit1.posy+3," record ");
                  break;
            case 2:
                  fprintf(recfile,"%s %s %f %i %i",
                                                ctrtag[trec0].tagname,
                                                ctrtag[trec0].engunits,
                                                tunit0.maxtime,
                                                tunit0.lowlimt,
                                                tunit0.hilimt);
                  fprintf(recfile,"%s %s %f %i %i",
                                                ctrtag[trec1].tagname,
                                                ctrtag[trec1].engunits,
                                                tunit1.maxtime,
                                                tunit1.lowlimt,
                                                tunit1.hilimt);
                  fprintf(recfile,"%s %s %f %i %i",
                                                ctrtag[trec2].tagname,
                                                ctrtag[trec2].engunits,
                                                tunit2.maxtime,
                                                tunit2.lowlimt,
                                                tunit2.hilimt);
                  fprintf(recfile,"%s %s %f %i %i",
                                                ctrtag[trec3].tagname,
                                                ctrtag[trec3].engunits,
                                                tunit3.maxtime,
                                                tunit3.lowlimt,
                                                tunit3.hilimt);
                  outmesg(tunit0.posx+3,tunit0.posy+3," record ");
                  outmesg(tunit1.posx+3,tunit1.posy+3," record ");
                  outmesg(tunit2.posx+3,tunit2.posy+3," record ");
                  outmesg(tunit3.posx+3,tunit3.posy+3," record ");
                  break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  }
}


// play
void do22sub7(FILE *recfile,float usemagnit, float useratio,long &curpos){

        extern void outmesg(int,int,char *);
        str15 tagn,engu;
        float maxtb;
        int lowlb,hilb;

        switch(statustgraph){
        case 0:fscanf(recfile,"%s %s %f %i %i",&tagn,&engu,
                                                &maxtb, &lowlb, &hilb);
                tunit0.reset_taxis();
                tunit0.cleargraph();
                tunit0.trendcreat(tagn,engu, usemagnit, useratio,
                                        maxtb,lowlb,hilb);
                outmesg(tunit0.posx+3,tunit0.posy+3," play ");
                break;
         case 1:fscanf(recfile,"%s %s %f %i %i",&tagn,&engu,
                                                &maxtb, &lowlb, &hilb);
                tunit0.reset_taxis();
                tunit0.cleargraph();
                tunit0.trendcreat(tagn,engu,    usemagnit, useratio,
                                        maxtb,lowlb,hilb);
                outmesg(tunit0.posx+3,tunit0.posy+3," play ");
                //..
                fscanf(recfile,"%s %s %f %i %i",&tagn,&engu,
                                                &maxtb, &lowlb, &hilb);
                tunit1.reset_taxis();
                tunit1.cleargraph();
                tunit1.trendcreat(tagn,engu,    usemagnit, useratio,
                                        maxtb,lowlb,hilb);
                outmesg(tunit1.posx+3,tunit1.posy+3," play ");
                break;

        case 2:fscanf(recfile,"%s %s %f %i %i",&tagn,&engu,
                                                &maxtb, &lowlb, &hilb);
                tunit0.reset_taxis();
                tunit0.cleargraph();
                tunit0.trendcreat(tagn,engu, usemagnit, useratio,
                                        maxtb,lowlb,hilb);
                outmesg(tunit0.posx+3,tunit0.posy+3," play ");
                //..
                fscanf(recfile,"%s %s %f %i %i",&tagn,&engu,
                                                &maxtb, &lowlb, &hilb);
                tunit1.reset_taxis();
                tunit1.cleargraph();
                tunit1.trendcreat(tagn,engu,    usemagnit, useratio,
                                        maxtb,lowlb,hilb);
                outmesg(tunit1.posx+3,tunit1.posy+3," play ");
                //..
                fscanf(recfile,"%s %s %f %i %i",&tagn,&engu,
                                                &maxtb, &lowlb, &hilb);
                tunit2.reset_taxis();
                tunit2.cleargraph();
                tunit2.trendcreat(tagn,engu,    usemagnit, useratio,
                                        maxtb,lowlb,hilb);
                outmesg(tunit2.posx+3,tunit2.posy+3," play ");
                //..
                fscanf(recfile,"%s %s %f %i %i",&tagn,&engu,
                                                &maxtb, &lowlb, &hilb);
                tunit3.reset_taxis();
                tunit3.cleargraph();
                tunit3.trendcreat(tagn,engu,    usemagnit,useratio,
                                        maxtb,lowlb,hilb);
                outmesg(tunit3.posx+3,tunit3.posy+3," play ");
                break;
        }
```

กสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        curpos = ftell(recfile);
}
//=====================
//= BAR graph
//=====================
#define BARCL DARKGRAY

const undertxt undbarg = {DARKGRAY,
                                        WHITE,
                                        YELLOW,
                                        BLACK,
                                        3,
                                        " EXIT   ",
                                        " START  ",
                                        " STOP   "};

//=================
// BAR display
//=================
void do23(void)
{
        extern void  getserialdata(float &,float &,float &,float &,int,boolean &);

        float pvbar,cvbar,spbar,smbar;
        boolean okflag;  //flag for serial data

 acttf stpgrpfg = {FALSE,FALSE,TRUE,FALSE,FALSE,FALSE,FALSE,FALSE,FALSE,FALSE};
 acttf rungrpfg = {TRUE,TRUE,FALSE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE};

        FILE *inf;

        initserial();
        setviewport(1,20,getmaxx(),getmaxy()-20,1);
        umouse.show(FALSE);
        clearviewport();
        umouse.show(TRUE);
        setviewport(0,0,getmaxx(),getmaxy(),1);
        if((inf = fopen(gfilename,"rb")) == NULL){
                beep();
                subp2.massage(" TAG file is not open.",YELLOW);
                fclose(inf);
        }
 else {
         inf = fopen(gfilename,"rb");
         for(int i = 1;i<MAXCTL;i++)
                        fread(&ctrtag[i],sizeof(taglist),1,inf);
         fclose(inf);
         bartrend  bg1(1  ,20 ,1,BARCL,WHITE);
         bartrend  bg2(160,20 ,1,BARCL,WHITE);
         bartrend  bg3(320,20 ,1,BARCL,WHITE);
         bartrend  bg4(480,20 ,1,BARCL,WHITE);
         bartrend  bg5(1  ,165,1,BARCL,WHITE);
         bartrend  bg6(160,165,1,BARCL,WHITE);
         bartrend  bg7(320,165,1,BARCL,WHITE);
         bartrend  bg8(480,165,1,BARCL,WHITE);
         bartrend  bg9(1  ,310,1,BARCL,WHITE);
         bartrend bg10(160,310,1,BARCL,WHITE);
         bartrend bg11(320,310,1,BARCL,WHITE);
         bartrend bg12(480,310,1,BARCL,WHITE);
         bg1.barcreat(ctrtag[1].tagname,ctrtag[1].engunits,
                                ctrtag[1].lowscl,ctrtag[1].upperscl);
         bg2.barcreat(ctrtag[2].tagname,ctrtag[2].engunits,
                                ctrtag[2].lowscl,ctrtag[2].upperscl);
         bg3.barcreat(ctrtag[3].tagname,ctrtag[3].engunits,
                                ctrtag[3].lowscl,ctrtag[3].upperscl);
         bg4.barcreat(ctrtag[4].tagname,ctrtag[4].engunits,
                                ctrtag[4].lowscl,ctrtag[4].upperscl);
         bg5.barcreat(ctrtag[5].tagname,ctrtag[5].engunits,
                                ctrtag[5].lowscl,ctrtag[5].upperscl);
         bg6.barcreat(ctrtag[6].tagname,ctrtag[6].engunits,
                                ctrtag[6].lowscl,ctrtag[6].upperscl);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
bg7.barcreat(ctrtag[7].tagname,ctrtag[7].engunits,
                    ctrtag[7].lowscl,ctrtag[7].upperscl);
bg8.barcreat(ctrtag[8].tagname,ctrtag[8].engunits,
                    ctrtag[8].lowscl,ctrtag[8].upperscl);
bg9.barcreat(ctrtag[9].tagname,ctrtag[9].engunits,
                    ctrtag[9].lowscl,ctrtag[9].upperscl);
bg10.barcreat(ctrtag[10].tagname,ctrtag[10].engunits,
                    ctrtag[10].lowscl,ctrtag[10].upperscl);
bg11.barcreat(ctrtag[11].tagname,ctrtag[11].engunits,
                    ctrtag[11].lowscl,ctrtag[11].upperscl);
bg12.barcreat(ctrtag[12].tagname,ctrtag[12].engunits,
                    ctrtag[12].lowscl,ctrtag[12].upperscl);
underbar barbarg(undbarg);
barbarg.setact(rungrpfg);   //all TRUE
barbarg.show();
boolean exitdo = FALSE;
do{
      barbarg.baroff();
      barbarg.baron();
      do{
            barbarg.select();
      }while(barbarg.exith == FALSE);
      switch(barbarg.statusund){
      case 0: exitdo = TRUE;break;
      case 1:barbarg.setact(stpgrpfg);
                  barbarg.show();
                  barbarg.baroff();barbarg.baron();
                  do {
                    getserialdata(pvbar,cvbar,spbar,smbar,1,okflag);
                    if(okflag) bg1.plotbar(pvbar,cvbar,spbar);
                    getserialdata(pvbar,cvbar,spbar,smbar,2,okflag);
                    if(okflag) bg2.plotbar(pvbar,cvbar,spbar);
                    getserialdata(pvbar,cvbar,spbar,smbar,3,okflag);
                    if(okflag) bg3.plotbar(pvbar,cvbar,spbar);
                    getserialdata(pvbar,cvbar,spbar,smbar,4,okflag);
                    if(okflag) bg4.plotbar(pvbar,cvbar,spbar);
                    getserialdata(pvbar,cvbar,spbar,smbar,5,okflag);
                    if(okflag) bg5.plotbar(pvbar,cvbar,spbar);
                    getserialdata(pvbar,cvbar,spbar,smbar,6,okflag);
                    if(okflag) bg6.plotbar(pvbar,cvbar,spbar);
                    getserialdata(pvbar,cvbar,spbar,smbar,7,okflag);
                    if(okflag) bg7.plotbar(pvbar,cvbar,spbar);
                    getserialdata(pvbar,cvbar,spbar,smbar,8,okflag);
                    if(okflag) bg8.plotbar(pvbar,cvbar,spbar);
                    getserialdata(pvbar,cvbar,spbar,smbar,9,okflag);
                    if(okflag) bg9.plotbar(pvbar,cvbar,spbar);
                    getserialdata(pvbar,cvbar,spbar,smbar,10,okflag);
                    if(okflag) bg10.plotbar(pvbar,cvbar,spbar);
                    getserialdata(pvbar,cvbar,spbar,smbar,11,okflag);
                    if(okflag) bg11.plotbar(pvbar,cvbar,spbar);
                    getserialdata(pvbar,cvbar,spbar,smbar,12,okflag);
                    if(okflag) bg12.plotbar(pvbar,cvbar,spbar);

                    int delayloop = (int)(time_samp*20);
                    for(int i=0;i<=delayloop;i++){
                          barbarg.select();
                          if(barbarg.statusund == 2&&barbarg.undactive ==TRUE)
                                break;
                          delay(50); //sec - msec
                    }
                    if(barbarg.statusund == 2&&barbarg.undactive ==TRUE)break;
                  }while(TRUE);
                  barbarg.setact(rungrpfg);
                  barbarg.show();
                  barbarg.baroff();barbarg.baron();
                  break;
      case 2:break;
      } //swicth
}while(exitdo == FALSE);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void do24(void){} //reserve
void do25(void){}


//---------- < OPTION >---------

//=================
// save config
//=================


typedef struct cfgstrct {
            int statusport;
            int statusbaud;
            int statustgraph;
            float time_samp;
            int statusvar;
            boolean realmodefg;
            str15 gfilename;
            float trend_cal[3];
            };

void do30(void)
{
      cfgstrct cfglist;
      FILE *out;
      str25 namefile = "plant.cfg";

      blankscr();
      setcolor(YELLOW);
      outtextxy(100,100," WAIT!  for saving config in 'PLANT.CFG' file.");
      cfglist.statusport = statusport;
      cfglist.statusbaud = statusbaud;
      cfglist.statustgraph = statustgraph;
      cfglist.time_samp = time_samp;
      cfglist.statusvar = statusvar;
      cfglist.realmodefg = realmodefg;
      strcpy(cfglist.gfilename,gfilename);
      for(int i = 0;i< 3;i++)
            cfglist.trend_cal[i] = trend_cal[i];
      out = fopen(namefile,"w+b");
      fwrite(&cfglist,sizeof(cfgstrct),1,out);
      fclose(out);
      blankscr();
      subp3.massage(" SAVE CONFIG finished. ",LIGHTRED);
}

//=================
// list config on CRT
//=================
typedef char  str70[70];
void do31(void)
{

      str15 baud[7] = {  "  150  BAUD ",
                         "  300  BAUD ",
                         "  600  BAUD ",
                         " 1200  BAUD ",
                         " 2400  BAUD ",
                         " 4800  BAUD ",
                         " 9600  BAUD "};

      str15 portcom[4] ={" COM 1:(3F8) ",
                         " COM 2:(2F8) ",
                         "   COM 3:  ",
                         "   COM 4:  "};

      str15 tust[4]= {  "  1-graph  ",
                        "  2-graph  ",
                        "  4-graph  ",
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        " 6-graph  "};

str15 varline[5]= {"    PV    ",
                        "  PV & SP  ",
                        "PV & SP & CV",
                        "    CV    ",
                        "  CV & SP  "};

        char *name;
        str70  buff1;
        str25  buff2;
        int dec,sign;

        blankscr();
        umouse.show(FALSE);
        box_creatl(30,80,getmaxx()-30,300,DARKGRAY,WHITE);
        setcolor(WHITE);

        outtextxy(60,100," < CONFIGULATION >");

        setcolor(YELLOW);
 // show baud rate
        strcpy(buff1," Baud rate              = ");
        strcat(buff1,baud[statusbaud]);
        outtextxy(60,120,buff1);

        // show com port
        strcpy(buff1," Comunication port      = ");
        strcat(buff1,portcom[statusport]);
        outtextxy(60,140,buff1);

        // NO. of TREND on CRT
        strcpy(buff1," NO. of TREND graph on CRT  = ");
        strcat(buff1,tust[statustgraph]);
        outtextxy(60,160,buff1);

        // variable on TREND
        strcpy(buff1," Variable line of TREND graph = ");
        strcat(buff1,varline[statusvar]);
        outtextxy(60,180,buff1);

        // show sampling delay
        sprintf(buff2," %6.3f sec. ",time_samp);
        strcpy(buff1," Sampling delay time.    = ");
        strcat(buff1,buff2);
        outtextxy(60,200,buff1);

        // working mode
        strcpy(buff1," Program operating type    = ");
        if(realmodefg == FALSE) strcat(buff1," Ramdomize ");
        else if(realmodefg == TRUE) strcat(buff1," Real time ");
        else strcat(buff1," Error operation ");
        outtextxy(60,220,buff1);

        // tag file
        strcpy(buff1," TAG file name          = ");
        strcat(buff1,gfilename);
        outtextxy(60,240,buff1);

        // trend calibation
        strcpy(buff1," Trend calibate (sec). = ");
        for(int i = 0 ;i<3;i++){
                sprintf(buff2,"%6.3f,",trend_cal[i]);
                strcat(buff1,buff2);
        }
        outtextxy(60,260,buff1);

        // out menu
        setcolor(RED);
        outtextxy(60,280 ," RETURN to MENU ");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            umouse.show(TRUE);

}

void do32(void){} //-------

void do33(void){   //RAMDOM MODE
            blankscr();
            realmodefg = FALSE;
            subp3.massage(" Randomized mode set OK. ",LIGHTGRAY);

} // reserve

vòid do34(void){   //REAL MODE
            blankscr();
            realmodefg = TRUE;
            subp3.massage(" Real time mode set OK. ",LIGHTGRAY);

} // reserve

void do35(void){} //-------


void do36(void){

const subm  calbst = {LIGHTGRAY,
                      YELLOW,
                      DARKGRAY,
                      WHITE,
                      BLACK,
                      100,
                      300,
                      17,
                      3,
                      " <PORT SETTING> ",
                      YELLOW,
                      "  for  1 TREND  ",
                      "  for  2 TRENDs ",
                      "  for  4 TRENDs ",
                      "  ",
                      "  ",
                      "  ",
                      "  ",
                      "  ",
                      "  "};

    sub calbsub(calbst);    // decare  submenu variable of baud
    char *name;
    name  = " Put calbrate time.<sec>:";
    double oldtcal;

    blankscr();
    box_creatl(30,30,getmaxx()-30,220,DARKGRAY,YELLOW);
    setcolor(WHITE);
    outtextxy(100,60, "Trend Graph Calibating ");
    outtextxy(100,80, "  - It's option that need to plot graph trend.");
    outtextxy(100,100,"    Therefore the graph nomally is ploted slower");
    outtextxy(100,120,"    than real clock . So you that calibate to  ");
    outtextxy(100,140,"    coresponding with real clock.    ");
    outtextxy(100,160,"  - Set such as 386sx = 0.10 sec.   ");
    outtextxy(100,180,"              486dx = 0.03 sec.   ");

    calbsub.subact(bartype1);
    calbsub.showsub();
    calbsub.select();
    calbsub.subclose();

    oldtcal  = (double)trend_cal[calbsub.statusrow];
    trend_cal[calbsub.statusrow] = getfloat(120,300,oldtcal,name);
    subp3.massage(" Set calibate time OK! ",LIGHTRED);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//=========================
// get data from serial =
//=========================
// use routine InitialSerial in do23 and do22

int  pvran[13] = {0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,
                  0x80,0x80,0x80,0x80,0x80,0x80,0x80};
int      cvran[13] = {0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,
                  0x80,0x80,0x80,0x80,0x80,0x80,0x80};
int      spran[13] = {0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,
                  0x80,0x80,0x80,0x80,0x80,0x80,0x80};
                        //use in RAMDOM mode "void getserialdata(..)"
void  getserialdata(float &pv,float &cv,float &sp,float &tm,
                    int noctl,boolean &ok){
static long int  MAXS = 10000;
          int buff;
          int pvi,cvi,spi;
          int scount;
          struct time sttime;
          struct time sptime;
          gettime(&sttime);
          delay(2);
          //keep flag
          switch(statusvar){
               case 0: pvflag = TRUE;cvflag = FALSE;spflag = FALSE;break;
               case 1: pvflag = TRUE;cvflag = FALSE;spflag = TRUE ;break;
               case 2: pvflag = TRUE;cvflag = TRUE ;spflag = TRUE ;break;
               case 3: pvflag = FALSE;cvflag = TRUE;spflag = FALSE;break;
               case 4: pvflag = FALSE;cvflag = TRUE;spflag = TRUE ;break;
          };
          if(realmodefg) {
               // serial Link with controller
               ok = TRUE;
               if(dataready())
                    buff = serialrecv();  //over data not use
               scount = 0;
//             do{scount++;}while(!thold()&&(scount <= MAXS));
//             if(scount< MAXS)  serialsend(65);      //send "a"

          // for  header of frame
               do{scount++;}while(!thold()&&(scount <= MAXS));
               if(scount< MAXS)  serialsend(0xAA);  //send "10101010" for header

          // No. of call controller  00..11  if show 1..12
               do{scount++;}while(!thold()&&(scount <= MAXS));
               if(scount< MAXS)  serialsend(noctl-1);  //send no. of controller

          // recive PV
               scount = 0;
               do{scount++;}while(!dataready()&&(scount <= MAXS));
               if(scount< MAXS)  pvi = serialrecv();
               else pvflag = FALSE;

               if(scount >= MAXS) ok = FALSE;

               scount = 0;
               do{scount++;}while(dataready()&&(scount <= MAXS));
               if(scount >= MAXS) ok = FALSE;  // chk ready = FALSE

          // recive CV
               scount = 0;
               do{scount++;}while(!dataready()&&(scount <= MAXS));
               if(scount< MAXS)  cvi = serialrecv();
               else cvflag = FALSE;

               if(scount >= MAXS) ok = FALSE;

               scount = 0;
               do{scount++;}while(dataready()&&(scount <= MAXS));
               if(scount >= MAXS) ok = FALSE;  // chk ready = FALSE
          // recive SP
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                scount = 0;
                do{scount++;}while(!dataready()&&(scount <= MAXS));
                if(scount< MAXS)  spi = serialrecv();
                else spflag = FALSE;

                if(scount >= MAXS) ok = FALSE;

                scount = 0;
                do{scount++;}while(dataready()&&(scount <= MAXS));
                if(scount >= MAXS) ok = FALSE;   // chk ready = FALSE
        }// if
        else {
                randomize();
                int plus = random(3);
                if(plus==1)       pvi = pvran[noctl] + random(5);
                else if(plus==0)      pvi = pvran[noctl] - random(5);
                else                  pvi = pvran[noctl];

                plus = random(3);
                if(plus==1)               cvi = cvran[noctl] + random(5);
                else if(plus==0)      cvi = cvran[noctl] - random(5);
                else              cvi = cvran[noctl];

                plus = random(3);
                if(plus==1)       spi = spran[noctl] + random(5);
                else if(plus==0)   spi = spran[noctl] - random(5);
                else              spi = spran[noctl];

                if(pvi > 0xFF) pvi = 0xFF;
                if(cvi > 0xFF) cvi = 0xFF;
                if(spi > 0xFF) spi = 0xFF;
                if(pvi < 0) pvi = 0;
                if(cvi < 0) cvi = 0;
                if(spi < 0) spi = 0;
                ok = TRUE;
                pvran[noctl] = pvi;
                cvran[noctl] = cvi;
                spran[noctl] = spi;
                delay(50);
        };
        // end serial link

        // value adjusting for comfortable in individual controllor
        float differ = ctrtag[noctl].upperscl - ctrtag[noctl].lowscl;
        pv = ctrtag[noctl].lowscl+differ*(pvi/255.0);
        cv = ctrtag[noctl].lowscl+differ*(cvi/255.0);
        sp = ctrtag[noctl].lowscl+differ*(spi/255.0);
        gettime(&sptime);
        float buft1 = (sttime.ti_min*60)+(sttime.ti_sec)
                                +(sttime.ti_hund/100.0);
        float buft2 = (sptime.ti_min*60)+(sptime.ti_sec)
                                +(sptime.ti_hund/100.0);
        tm = buft2 - buft1;
}

//=============================
// Clear SCR without  main menu
//=============================
void blankscr(void)
{
        setviewport(0,22,getmaxx(),getmaxy(),1);
        umouse.show(FALSE);
        clearviewport();
        umouse.show(TRUE);
        setviewport(0,0,getmaxx(),getmaxy(),1);
}
//..
void  outmesg(int left,int top,char *buff) {
        setcolor(YELLOW);
        settextjustify(LEFT_TEXT,TOP_TEXT);
        moveto(left,top);
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        for(int i = 0;i<= strlen(buff);i++) outtext("Û");
        setcolor(BLACK);
        outtextxy(left,top, buff);
}

/*********************/
/* end dopull.cpp  */
/*********************/

// tagset.cpp
/*****************************************************/
/* program routines for TAG setting of controller  */
/*****************************************************/

#include <stdio.h>
#include <conio.h>
#include <ctype.h>
#include <dos.h>

#define MAXCTL  13
            // define  table coordinate constant
#define UPOS    5
#define TAGPOS  30
#define SAMPOS  100
#define DISPOS  150
#define ENGPOS  350
#define UPPOS   420
#define LWPOS   490
#define ENDPOS  560
            // define line of table
#define YHEAD   70
#define YEND    300

typedef char  str15[15];
typedef char  str25[25];
typedef char  str40[40];
typedef struct  taglist{
            str15  tagname;
            str25  discript;
            str15  engunits;
            float   maxmt;     // time base
            int   upperscl;  // upper scale
            int   lowscl;    // lower scale
};
struct taglist ctrtag[MAXCTL];  //use in do##();

//=========================
//  tag-setting routine
//=========================
void  beep(void)
{
      sound(1000);
      delay(100);
      nosound();
}

void  tagmenu(void)
{
            #define HTEXT   40
            setviewport(0,20,getmaxx(),getmaxy()-20,1);
            clearviewport();
            setviewport(0,0,getmaxx(),getmaxy(),1);
            setcolor(WHITE);
            rectangle(1,25,getmaxx()-1,YEND);
            rectangle(1,25,getmaxx()-1,YHEAD+10);
            setcolor(YELLOW);
            settextstyle(DEFAULT_FONT,HORIZ_DIR,2);
            settextjustify(CENTER_TEXT,CENTER_TEXT);
            int midx = getmaxx()/2;
            outtextxy(midx,HTEXT," TAG table ");
            settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            settextjustify(LEFT_TEXT,CENTER_TEXT);
            setcolor(WHITE);
            outtextxy(UPOS,YHEAD ," #");
            outtextxy(TAGPOS,YHEAD," Tag");
            outtextxy(SAMPOS,YHEAD," T base");
            outtextxy(DISPOS,YHEAD,"  Discript ");
            outtextxy(ENGPOS,YHEAD," ENG/Units");
            outtextxy(UPPOS,YHEAD,"  Upper");
            outtextxy(LWPOS,YHEAD,"  Lower");
}

//=============================
//  Edit tag routine
//=============================
int getunit(int x,int y)
{
      char *name;
      str15 oldname;
      str15 strbuff;
      str15 inname = "";
      int keyin;
      name  = "NO. control unit:";
      setcolor(WHITE);
      setfillstyle(SOLID_FILL,YELLOW);
      setviewport(0,0,getmaxx(),getmaxy(),1);
      settextjustify(LEFT_TEXT,CENTER_TEXT);
      outtextxy(x, y ,name);
      int vx1 = x + strlen(name)*8;
      int vx2 = vx1 + 4*8;
      setcolor(RED);
      setviewport(vx1,y-4,vx2,y+4,1);
      bar(0,0,100,30);
      while((keyin = getch()) != 13) {  // press ENTER
            if(( isdigit(keyin) != 0) &&  (strlen(inname) < 2)){
                  strcpy(oldname,inname);
                  sprintf(strbuff,"%c",keyin);
                  strcat(inname,strbuff);
                  if(atoi(inname) < MAXCTL && atoi(inname) > 0){
                        bar(0,0,100,30);
                        outtextxy(8,5,inname);
                  }
                  else { beep() ;strcpy(inname,oldname); }
            }
            else if(keyin == 8) {   //press BACKSPACE
                  unsigned strwide = strlen(inname)-1;
                  inname[strwide] = '\0';
                  bar(0,0,100,50);
                  outtextxy(8,5,inname);
            }
            else beep();
      }
//    if(atoi(inname) == 0)
//          return(1);
      return(atoi(inname));
}

char *gettag(int x,int y)
{
      char *name;
      str15 strbuff;
      str15 inname = "";
      int keyin;
      name  = "Tag name:";
      setcolor(WHITE);
      setfillstyle(SOLID_FILL,YELLOW);
      setviewport(0,0,getmaxx(),getmaxy(),1);
      settextjustify(LEFT_TEXT,CENTER_TEXT);
      outtextxy(x, y ,name);
      int vx1 = x + strlen(name)*8;
      int vx2 = vx1 + 10*8;
      setcolor(RED);
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        setviewport(vx1,y-5,vx2,y+5,1);
        bar(0,0,100,30);
        while((keyin = getch()) != 13) {  // press ENTER
            if(( isalnum(keyin) != 0) &&  (strlen(inname) < 8)){
                sprintf(strbuff,"%c",keyin);
                strcat(inname,strbuff);
                bar(0,0,100,30);
                outtextxy(8,5,inname);
            }
            else if(keyin == 8) {   //press BACKSPACE
                unsigned strwide = strlen(inname)-1;
                inname[strwide] = '\0';
                bar(0,0,100,30);
                outtextxy(8,5,inname);
            }
        }
        return(inname);

}

//=====================
// get discription
//=====================
char *getdiscp(int x,int y)
{
        char *name;
        str25 strbuff;
        char inname[25];
        strcpy(inname ,"");
        int keyin;
        name  = "discription:";
        setcolor(WHITE);
        setfillstyle(SOLID_FILL,YELLOW);
        setviewport(0,0,getmaxx(),getmaxy(),1);
        settextjustify(LEFT_TEXT,CENTER_TEXT);
        outtextxy(x, y ,name);
        int vx1 = x + strlen(name)*8;
        int vx2 = vx1 + 27*8;
        setcolor(RED);
        setviewport(vx1,y-5,vx2,y+5,1);
        bar(0,0,200,30);
        while((keyin = getch()) != 13) {  // press ENTER
            if(keyin == 8) {   //press BACKSPACE
                unsigned strwide = strlen(inname)-1;
                inname[strwide] = '\0';
                bar(0,0,200,30);
                outtextxy(8,5,inname);
            }
            else if( strlen(inname) < 23){
                sprintf(strbuff,"%c",keyin);
                strcat(inname,strbuff);
                bar(0,0,200,30);
                outtextxy(8,5,inname);
            }
        }
        return(inname);
}

//=====================
// get engineering unit
//=====================
char *geteng(int x,int y)
{
        char *name;
        str25 strbuff;
        str15 inname = "";
        int keyin;
        name  = "ENG/Units:";
        setcolor(WHITE);
        setfillstyle(SOLID_FILL,YELLOW);
        setviewport(0,0,getmaxx(),getmaxy(),1);
```

เอกสารนี้ใช้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        settextjustify(LEFT_TEXT,CENTER_TEXT);
        outtextxy(x, y ,name);
        int vx1 = x + strlen(name)*8;
        int vx2 = vx1 + 10*8;
        setcolor(RED);
        setviewport(vx1,y-5,vx2,y+5,1);
        bar(0,0,100,30);
        while((keyin = getch()) != 13) {  // press ENTER
                if(keyin == 8) {   //press BACKSPACE
                        unsigned strwide = strlen(inname)-1;
                        inname[strwide] = '\0';
                        bar(0,0,100,30);
                        outtextxy(8,5,inname);
                }
                else if(strlen(inname) < 8){
                        sprintf(strbuff,"%c",keyin);
                        strcat(inname,strbuff);
                        bar(0,0,100,30);
                        outtextxy(8,5,inname);
                }
        }
        return(inname);
}

//=========================
// get upper magnitude
//=========================
int getupper(int x,int y)
{
        char *name;
        str15 strbuff;
        str15 inname = "";
        int keyin;
        name  = "Upper scale:";
        setcolor(WHITE);
        setfillstyle(SOLID_FILL,YELLOW);
        setviewport(0,0,getmaxx(),getmaxy(),1);
        settextjustify(LEFT_TEXT,CENTER_TEXT);
        outtextxy(x, y ,name);
        int vx1 = x + strlen(name)*8;
        int vx2 = vx1 + 8*8;
        setcolor(RED);
        setviewport(vx1,y-5,vx2,y+5,1);
        bar(0,0,100,30);
        while((keyin = getch()) != 13) {  // press ENTER
                if(( isdigit(keyin) != 0) &&  (strlen(inname) < 5)){
                        sprintf(strbuff,"%c",keyin);
                        strcat(inname,strbuff);
                        bar(0,0,100,30);
                        outtextxy(8,5,inname);
                }
                else if(keyin == 8) {   //press BACKSPACE
                        unsigned strwide = strlen(inname)-1;
                        inname[strwide] = '\0';
                        bar(0,0,100,30);
                        outtextxy(8,5,inname);
                }
                else beep();
        }
        return(atoi(inname));
}

//=========================
// get lower magnitude
//=========================
int getlower(int x,int y)
{
        char *name;
        str15 strbuff;
        str15 inname = "";
        int keyin;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        name  = "Lower scale:";
        setcolor(WHITE);
        setfillstyle(SOLID_FILL,YELLOW);
        setviewport(0,0,getmaxx(),getmaxy(),1);
        settextjustify(LEFT_TEXT,CENTER_TEXT);
        outtextxy(x, y ,name);
        int vx1 = x + strlen(name)*8;
        int vx2 = vx1 + 8*8;
        setcolor(RED);
        setviewport(vx1,y-5,vx2,y+5,1);
        bar(0,0,100,30);
        while((keyin = getch()) != 13) {  // press ENTER
                if(( isdigit(keyin) != 0) &&  (strlen(inname) < 5)){
                        sprintf(strbuff,"%c",keyin);
                        strcat(inname,strbuff);
                        bar(0,0,100,30);
                        outtextxy(8,5,inname);
                }
                else if(keyin == 8) {   //press BACKSPACE
                        unsigned strwide = strlen(inname)-1;
                        inname[strwide] = '\0';
                        bar(0,0,100,30);
                        outtextxy(8,5,inname);
                }
                else beep();
        }
        return(atoi(inname));
}

//=============================
// get maximum time of X-axis
//=============================
float getmaxt(int x,int y)
{
        char *name;
        str15 strbuff;
        str15 inname = "";
        int keyin;
        name  = "Time base<min>:";
        setcolor(WHITE);
        setfillstyle(SOLID_FILL,YELLOW);
        setviewport(0,0,getmaxx(),getmaxy(),1);
        settextjustify(LEFT_TEXT,CENTER_TEXT);
        outtextxy(x, y ,name);
        int vx1 = x + strlen(name)*8;
        int vx2 = vx1 + 6*8;
        setcolor(RED);
        setviewport(vx1,y-5,vx2,y+5,1);
        bar(0,0,100,30);
        while((keyin = getch()) != 13) {  // press ENTER
                if(keyin == 8) {   //press BACKSPACE
                        unsigned strwide = strlen(inname)-1;
                        inname[strwide] = '\0';
                        bar(0,0,100,50);
                        outtextxy(8,5,inname);
                }
                else if((isalpha(keyin) == 0) && strlen(inname) < 4){
                        sprintf(strbuff,"%c",keyin);
                        strcat(inname,strbuff);
                        bar(0,0,100,30);
                        outtextxy(8,5,inname);
                }
                else beep();
        }
        return(atof(inname));
}

//=============================
//  Put data in TAG table routine
//=============================
void inittag(void)
```

ไฟล์นี้จัดทำขึ้นเพื่อเก็บไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
        for(int i = 0;i<MAXCTL;i++) {
            strcpy(ctrtag[i].tagname,"?");
            strcpy(ctrtag[i].discript,"?");
            strcpy(ctrtag[i].engunits,"?");
            ctrtag[i].maxmt = 1;
            ctrtag[i].upperscl = 100;
            ctrtag[i].lowscl = 0;
        }
}

//=========================
// show TAG table on CRT
//=========================
void puttagtbl(void)
{
        #define YTEXT   75
        str25 buff;
        setviewport(2,YHEAD+12,getmaxx()-3,YEND-2,1);
        clearviewport();
        setviewport(0,0,getmaxx(),getmaxy(),1);
        setcolor(WHITE);
        line(TAGPOS,YHEAD+10,TAGPOS,YEND);
        line(SAMPOS,YHEAD+10,SAMPOS,YEND);
        line(DISPOS,YHEAD+10,DISPOS,YEND);
        line(ENGPOS,YHEAD+10,ENGPOS,YEND);
        line(UPPOS,YHEAD+10,UPPOS,YEND);
        line(LWPOS,YHEAD+10,LWPOS,YEND);
        line(ENDPOS,YHEAD+10,ENDPOS,YEND);
        setcolor(LIGHTRED);
        for(int i = 1;i<MAXCTL;i++) {
            sprintf(buff,"%2d",i);
            outtextxy(UPOS+4  ,YTEXT + (i*16),buff);
            outtextxy(TAGPOS+4,YTEXT + (i*16),ctrtag[i].tagname);
            outtextxy(DISPOS+4,YTEXT + (i*16),ctrtag[i].discript);
            outtextxy(ENGPOS+4,YTEXT + (i*16),ctrtag[i].engunits);
            sprintf(buff,"%6.2f",ctrtag[i].maxmt);
            outtextxy(SAMPOS+2,YTEXT + (i*16),buff);
            sprintf(buff,"%6i",ctrtag[i].upperscl);
            outtextxy(UPPOS+4,YTEXT + (i*16),buff);
            sprintf(buff,"%6i",ctrtag[i].lowscl);
            outtextxy(LWPOS+4,YTEXT + (i*16),buff);
        }
}

//=========================
// TAG file edition
//=========================
void edittag(char *filename)
{
        windobj helpm;
        int ch;
        str40 hdedit = "Editting TAG file : ";
        str25 name0 = "NO. control unit: ÛÛÛ";
        str25 name1 = "Tag name: ÛÛÛÛÛÛÛÛÛ";
        str40 name2 = "discription: ÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛÛ";
        str25 name3 = "ENG/Units: ÛÛÛÛÛÛÛÛÛ";
        str25 name4 = "Upper scale: ÛÛÛÛÛÛ";
        str25 name5 = "Lower scale: ÛÛÛÛÛÛ";
        str25 name6 = "Time base<min>: ÛÛÛÛÛ";
        setviewport(2,310,getmaxx()-2,330,1);
        clearviewport();
        setviewport(0,0,getmaxx(),getmaxy(),1);
        // show header file in tag
        strcat(hdedit,filename);
        settextjustify(CENTER_TEXT,CENTER_TEXT);
        setcolor(LIGHTGREEN);
        outtextxy(getmaxx()/2,320,hdedit);
        // make tag editor
        settextjustify(LEFT_TEXT,CENTER_TEXT);
        setcolor(WHITE);
```

เอกสารนี้ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        rectangle(1,309,getmaxx(),getmaxy()-49);
        unsigned statusunits;
        do {
              setviewport(2,340,getmaxx()-2,getmaxy()-50,1);
              clearviewport();
              setviewport(0,0,getmaxx(),getmaxy(),1);
              // help messg
              setcolor(YELLOW);
              outtextxy(20,410,
              "** HELP: Please put DATA in the YELLOW box by keyboard.");
              // tag editor;
              setcolor(WHITE);
              settextjustify(LEFT_TEXT,CENTER_TEXT);
              outtextxy(20,350,name0);
              outtextxy(20,370,name1);
              outtextxy(20,390,name2);
              outtextxy(200,350,name3);
              outtextxy(200,370,name4);
              outtextxy(400,350,name5);
              outtextxy(400,370,name6);
              statusunits = getunit(20,350);
              // update controller status
              strcpy(ctrtag[statusunits].tagname,gettag(20,370));
              strcpy(ctrtag[statusunits].discript,getdiscp(20,390));
              strcpy(ctrtag[statusunits].engunits,geteng(200,350));
              ctrtag[statusunits].upperscl =  getupper(200,370);
              ctrtag[statusunits].lowscl   =  getlower(400,350);
              ctrtag[statusunits].maxmt    =  getmaxt(400,370);
              // put data in table
              puttagtbl();
              // call modify
              setviewport(0,0,getmaxx(),getmaxy(),1);
              helpm.window_open(200,350,400,400);
              helpm.window_creatsh(200,350,400,400,CYAN,"",BLACK,BLACK);
              setcolor(WHITE);
              outtextxy(216,366,"Modify another(Y/N)");
              ch = getch();
              helpm.window_close();
        }while(ch == 'y'|| ch == 'Y');  // y or Y
}

//=============================
//  main routine of tag-setting
//=============================
typedef struct namefm{
        char *namehead;
        char *nametail;
};

char *getname(int x,int y,namefm namein)
{
        windobj winname;
        str25  filename = "";
        str25  strbuff;
        int   keyin;
        int x2 = x + strlen(namein.namehead)*9;
        int y2 = y + 60;
        umouse.show(FALSE);
        winname.window_open(x,y,x2,y2);
        winname.window_creatsh(x,y,x2,y2,DARKGRAY, "< file >",YELLOW,LIGHTGRAY);
        setviewport(x,y,x2,y2,1);
        setcolor(WHITE);
        settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
        settextjustify(LEFT_TEXT,RIGHT_TEXT);
        setfillstyle(SOLID_FILL,YELLOW);
        bar(192,15,330,35);
        outtextxy(16,20,namein.namehead);
        setcolor(WHITE);
        outtextxy(16,20,namein.nametail);
        setcolor(DARKGRAY);
        while((keyin = getch()) != 13) {  // press ENTER
```

เอกสารนี้จัดทำขึ้นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if(keyin == 8) {   //press BACKSPACE
                unsigned strwide = strlen(filename)-1;
                filename[strwide] = '\0';
                bar(192,15,330,35);
                outtextxy(200,20,filename);
            }
            else if(strlen(filename) < 17){
                sprintf(strbuff,"%c",keyin);
                strcat(filename,strbuff);
                bar(192,15,330,35);
                outtextxy(200,20,filename);
            }
            else beep();
        }
        winname.window_close();
        umouse.show(TRUE);
        return(filename);
}

void newtagf(void){

  const namefm namefile={" Put TAG file name : <_                >",
                         "                    _          .tag "};

        windobj mesgwin;
        int  ch;
        char *fname;
        char *extnm = "";
        str25 strbuff;
        FILE *out;

        strcpy(extnm,".tag");
        do{
            fname = getname(60,200,namefile);
        }while(strcmp(fname ,"") == 0);
        strcpy(strbuff,fname);
        strcat(strbuff,extnm);
        out = fopen(strbuff,"w+b");
        // show tag & initialize
        inittag();
        tagmenu();
        puttagtbl();
        edittag(strbuff);
        // ack for saving new file
        mesgwin.window_open(250,350,450,400);
        mesgwin.window_creatsh(250,350,450,400,RED,"",YELLOW,BLACK);
        setcolor(WHITE);
        outtextxy(266,366,"Save edit file(Y/N)?");
        ch = getch();
        mesgwin.window_close();
        if(ch == 'y'||ch == 'Y'){
            for(int i = 1;i<MAXCTL;i++)
                fwrite(&ctrtag[i],sizeof(taglist),1,out);
        }
        fclose(out);
}

void loadtagf()
{
  const namefm namefile={" Put TAG file name : <_                >",
                         "                    _          .tag "};

        windobj mesgwin;
        int  ch;
        char *fname;
        str15  extnm = "";
        str25 strbuff= "";
        FILE *out;
        strcpy(extnm,".tag");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            fname = getname(60,200,namefile);
            strcpy(strbuff,fname);
            strcat(strbuff,extnm);
            if((out = fopen(strbuff,"r+b")) == NULL){
            // mesg for file exist
                  mesgwin.window_open(250,350,500,400);
                  mesgwin.window_creatsh(250,350,500,400,RED,"",YELLOW,BLACK);
                  setcolor(WHITE);
                  outtextxy(266,366," File not exist.<ENTER>");
                  ch = getch();
                  mesgwin.window_close();
                  fclose(out);
            }
            else {
                  for(int i = 1;i<MAXCTL;i++)
                        fread(&ctrtag[i],sizeof(taglist),1,out);
                  fclose(out);
            // show tag & initialize
                  tagmenu();
                  puttagtbl();
                  edittag(strbuff);
            // ack for saving new file
                  mesgwin.window_open(250,350,450,400);
                  mesgwin.window_creatsh(250,350,450,400,RED,"",YELLOW,BLACK);
                  setcolor(WHITE);
                  outtextxy(266,366,"Save edit file(Y/N)?");
                  ch = getch();
                  mesgwin.window_close();
                  if(ch == 'y'||ch == 'Y'){
                        out = fopen(strbuff,"wb");
                        for(int i = 1;i<MAXCTL;i++)
                              fwrite(&ctrtag[i],sizeof(taglist),1,out);
                        fclose(out);
                  }
               }
}

void tagwork(void)
{
      extern void blankscr();
const undertxt  tagund = { DARKGRAY,
                           WHITE,
                           YELLOW,
                           DARKGRAY,
                           3,
                           " New file  ",
                           " Load file ",
                           " Exit      "};

      boolean  bout = FALSE;
      underbar tagunder(tagund);
      tagunder.setact(bartype1);
      blankscr();
      umouse.show(FALSE);
      setcolor(YELLOW);
      outtextxy(150,66,"** TAG editing of controller **");
      outtextxy(10,98,
"# If there isn't any TAG files in directory.You can choose new files.");
      outtextxy(10,114,
"# If there is some TAG files in directory.The Load files choice allow you");
      outtextxy(10,130,
"   editting  on information in TAG file. and you can save respectively.");
      tagunder.show();
      do {
            tagunder.baroff();
            tagunder.baron();
            do {
                  tagunder.select();
            }while(tagunder.exith == FALSE);
            if(tagunder.undactive == TRUE){
                  switch(tagunder.statusund){
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        case 0:newtagf();break;
                        case 1:loadtagf();break;
                        case 2:bout = TRUE; break;
                    }
                tagunder.exith = FALSE;
                }
            }while(bout == FALSE);
            umouse.show(TRUE);
    }

    //-------- use in SET SAMPLING menu---------
    //=============================
    // get sampling time delay
    //=============================
    float getfloat(int x,int y,double oldfloat,char *name)
    {
            extern void box_creatl(int,int,int,int,int,int);

            int dec,sign;
            str15 strbuff,strresv;
            str15 inname = "";
            int keyin;

            box_creatl(x-20,y-20,x+280,y+20,DARKGRAY,YELLOW);
            setcolor(LIGHTGREEN);
            setfillstyle(SOLID_FILL,YELLOW);
            setviewport(0,0,getmaxx(),getmaxy(),1);
            settextjustify(LEFT_TEXT,CENTER_TEXT);
            outtextxy(x, y ,name);
            int vx1 = x + strlen(name)*8;
            int vx2 = vx1 + 9*8;
            setcolor(RED);
            setviewport(vx1,y-5,vx2,y+5,1);
            bar(0,0,250,100);
            // convert to STRING
            sprintf(strbuff,"%6.3f",oldfloat);
            setcolor(RED);
            outtextxy(8,5,strbuff);

            while((keyin = getch()) != 13) {  // press ENTER
                if(keyin == 8) {   //press BACKSPACE
                        unsigned strwide = strlen(inname)-1;
                        inname[strwide] = '\0';
                        bar(0,0,250,100);
                        outtextxy(8,5,inname);
                }
                else if((isalpha(keyin) == 0) && strlen(inname) < 7){
                        sprintf(strbuff,"%c",keyin);
                        strcat(inname,strbuff);
                        bar(0,0,250,100);
                        outtextxy(8,5,inname);
                }
                else beep();
            }
            setviewport(x-20,y-20,x+280,y+20,1);
            clearviewport();
            setviewport(0,0,getmaxx(),getmaxy(),1);

            return(atof(inname));
    }
    //=============
    // end TAGSET.CPP=
    //=============


    // trendrec.cpp
    /*************************************/
    /*  This routine of trendrecorder  */
    /*************************************/
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
    #include <graphics.h>
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
```

```
#include <math.h>
#include "underbar.cpp"


//=================================
//  Ask No. of TREND routine
//=================================

int statustgraph;      // set many of graph trend  variable

//=================================
//  Trend recorder routine.
//=================================

#define REPEATG  0.25  //define graph plot repeating
#define MINMAG   200   //minimum of x-scale
#define GRPPERT  0.7   //graph magnitude per boudrary of trendrec

int statusvar;         //show variable
boolean  pvflag = TRUE;
boolean  cvflag = TRUE;
boolean  spflag = TRUE;  // USE in Dopull.cpp ,DO14

typedef char str15[15];

class  datalink{
        public:
                int pvdata;
                int cvdata;
                int spdata;
                boolean pvflg;
                boolean cvflg;
                boolean spflg;
                datalink *_back;

};


class savdata{
        public:
                float pvf;
                float cvf;
                float spf;
                float samtf;
                savdata *_next;
};

int    getpix[200];   //keep point of putpixel

class trend {
       public:
              datalink  *tail;
              datalink  *nowcur;
              savdata   *savhead;
              int    posx,posy;  // coordinate of upperleft
              float  magnitude;  // magnitude X1 X2 X3 X4 X5
              int    trendcl;
              int    grphcl;
              int    txtcl;
              float  ratio;      //ratio of Y/X scale  use 0.7
              float  maxtime;    //time value maximum in scale in minute
              int    lowlimt,hilimt;
              int    xcount;     // count for one dot plot
              int    oldmaxcount; // keep xcount when the end graph
              int    timecount;   // keep no. of samptime
              float  oldsmtime;   // keep previous samtime
              float  taxis;       // keep axis time
              int    gphposx1,gphposy1;
              int    gphposx2,gphposy2;
              double  sumdttm,sumsamtm; //uses keep sumdot/time and sumsamtm;
              int   opospv,oposcv,opossp; //keep old plot
```

เอกสารนี้ใช้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        trend(int x,int y,int tcl,int gcl,int xcl):
                posx(x),posy(y),
                trendcl(tcl),grphcl(gcl),txtcl(xcl)
                { xcount = oldmaxcount = timecount = sumdttm = sumsamtm = 0;
                    taxis = 0.0;
                    tail = new datalink; tail = NULL;
                    nowcur = new datalink; nowcur = NULL;
                    savhead  = new savdata;  savhead  = NULL;};
        ~trend(){delete tail;delete nowcur ;delete savhead;};

        void pushdata(int pvd, int cvd, int spd,
                      boolean pvfg, boolean cvfg, boolean spfg);
        void seekdata(int &pvd,int &cvd, int &spd,
                      boolean &pvfg, boolean &cvfg, boolean &spfg);
        void deltodata(int count);
        void freedatamem(void);

        void addsav(float pvs,float cvs,float sps,float samts);
        boolean  findsav(int count,float &pvs,float &cvs,
                                              float &sps,float &samts);
        void delsavhead(void);
        void delsavall(void);

        void reset_taxis(void);
        void reset_timecountbox(void);
        void cleargraph(void);
        void trendcreat(str15 name,str15 pvname,
                        float mag,float rat,float maxt,
                        int lowl ,int hil);

      void graphplot(float pvin,float cvin,float spin,float samtm);
      void plotscale(void);
      void value_box(int x,int y,str15 tagname,str15 engunit);

};

void trend::pushdata(int pvd, int cvd, int spd,
                     boolean pvfg, boolean cvfg, boolean spfg){

    datalink *temp;
    if(!(temp = new datalink)){
        closegraph();
        printf("Allocate memory of datalink error");
        exit(1);
    }
    temp->pvdata = pvd;
    temp->cvdata = cvd;
    temp->spdata = spd;
    temp->pvflg = pvfg;
    temp->cvflg = cvfg;
    temp->spflg = spfg;

    temp->_back = tail;
    tail = nowcur = temp;

}

void trend::seekdata(int &pvd,int &cvd,int &spd,
                     boolean &pvfg, boolean &cvfg,boolean &spfg){

        datalink *temp;
        if(tail == NULL) return;
        if(nowcur == NULL) return;
        pvd  = nowcur->pvdata;
        cvd  = nowcur->cvdata;
        spd  = nowcur->spdata;
        pvfg = nowcur->pvflg;
        cvfg = nowcur->cvflg;
        spfg = nowcur->spflg;
```

เอกสารนี้จัดทำขึ้นเพื่อไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                temp = nowcur;
                nowcur = temp->_back;
        }

        void trend::deltodata(int count){
                datalink *temp,*last,*cursor;
                if(tail == NULL) return;
                for(int i = 0;i<count;i++){
                        cursor = last = tail;
                        while(cursor->_back != NULL){
                                temp = cursor;
                                last = cursor->_back;
                                cursor = last;
                        }
                        temp->_back = cursor->_back;
                        delete cursor;
                }
        }


        void trend::freedatamem(void){
                datalink *temp;
                while(tail != NULL){
                  temp = tail;
                  tail = temp->_back;
                  delete temp;
                }

        }

        void trend::addsav(float pvs,float cvs,float sps,float samts){
                savdata *temp ,*cursor,*last;

                if(!(temp = new savdata)){
                        closegraph();
                        printf("Allocate memory of savdata error");
                        exit(1);
                }
                temp->pvf  = pvs;
                temp->cvf  = cvs;
                temp->spf  = sps;
                temp->samtf = samts;
                temp->_next = NULL;

                if((cursor = savhead) == NULL){
                                savhead = temp;
                                return;
                }
                last = cursor;
                while(cursor->_next != NULL){
                        last = cursor->_next;
                        cursor = last;
                }
                cursor->_next = temp;
}

boolean trend::findsav(int count,float &pvs,float &cvs,
                                                float &sps,float &samts){
                savdata *cursor,*last;

                if(savhead == NULL)  return(FALSE);
                cursor =last= savhead;
                for(int i = 0;i<=count;i++){
                        last = cursor->_next;
                        cursor = last;
                        if(cursor == NULL) return(FALSE);
                }
                pvs = cursor->pvf;
                cvs = cursor->cvf;
                sps = cursor->spf;
                samts = cursor->samtf;
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            return(TRUE);
}

void trend::delsavhead(void){
            savdata *temp;
            if(savhead == NULL) return;
            temp = savhead;
            savhead = temp->_next;
            delete temp;
}
void trend::delsavall(void){
            savdata *temp;
            while(savhead != NULL){
                  temp = savhead;
                  savhead = temp->_next;
                  delete temp;
            }
}

void trend::reset_taxis(void){
            taxis = 0.0;
            timecount = 0;
            delsavall();
}

void trend::reset_timecountbox(void){
            timecount = 0;
            delsavall();
}
void trend::cleargraph(void) {
      extern void box_creatl(int,int,int,int,int,int);

      setviewport(0,0,getmaxx(),getmaxy(),1);
      box_creatl(gphposx1,gphposy1,gphposx2,gphposy2,grphcl,WHITE);
      setcolor(WHITE);

      // plot y
      int diflgl   = (gphposy2-gphposy1)/4;
      for(int i = 0;i < 5;i++){
            int betw = gphposy2-i*diflgl;
            setlinestyle(DOTTED_LINE,0x1,1);
            line(gphposx1, betw, gphposx2, betw);
      }

      // plot x
      diflgl   = (gphposx2-gphposx1)/4;
      for(i = 0;i < 5;i++){
            int betw = gphposx1+i*diflgl;
            setlinestyle(DOTTED_LINE,0x1,1);
            line(betw, gphposy1, betw, gphposy2);
      }

      // tell time axis
      str15 buff;
      settextjustify(RIGHT_TEXT,BOTTOM_TEXT);
      sprintf(buff,"%8.3fmin+",taxis);
      outtextxy(gphposx2-5,gphposy2-5,buff);


      oldmaxcount = xcount;
      xcount = 0;  // new initialized
      sumdttm = sumsamtm = 0;


id trend::trendcreat(str15 name,str15 pvname,float mag,float rat,float maxt,
                  int lowl ,int hil)

      #define YTGAP 4
      #define SHF  1
      extern void box_creatl(int,int,int,int,int,int);
```

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        magnitude = mag;
        ratio     = rat;
        maxtime   = maxt;
        lowlimt   = lowl;
        hilimt    = hil;

        setviewport(0,0,getmaxx(),getmaxy(),1);
        umouse.show(FALSE);
        int posxx = posx + MINMAG*magnitude;
        int posyy = posy + MINMAG*magnitude*ratio;
        box_creatl(posx,posy,posxx,posyy,trendcl,WHITE);
        int gphx1 = posx + ((MINMAG*magnitude -
                                        MINMAG*magnitude*GRPPERT)/2)*SHF;
        int gphy1 = posy + ratio*((MINMAG*magnitude -
                                        MINMAG*magnitude*GRPPERT)/2);
        int gphx2 = gphx1 + MINMAG*magnitude*GRPPERT;
        int gphy2 = gphy1 + MINMAG*magnitude*GRPPERT*ratio;
        gphposx1 = gphx1;
        gphposy1 = gphy1;
        gphposx2 = gphx2;
        gphposy2 = gphy2;
        int mid1x = posx + MINMAG*magnitude/2;
        int mid1y = posy + MINMAG*magnitude*ratio/2;
        int txt1y = posy + (YTGAP)*magnitude;
        settextjustify(CENTER_TEXT,CENTER_TEXT);
        if(magnitude <= 1) settextstyle(SMALL_FONT,HORIZ_DIR,2);
        else settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
        setcolor(WHITE);
        outtextxy(mid1x,txt1y,name);    //name of collorler
        setcolor(txtcl);
        outtextxy(gphposx2+(posxx-gphposx2)/2,gphposy2," Min"); //name of X-scale
        if(magnitude <= 1) settextstyle(SMALL_FONT,VERT_DIR,2);
        else settextstyle(DEFAULT_FONT,VERT_DIR,1);
        outtextxy(posx+(6*magnitude),mid1y,pvname); //name of Y-scale
        //..
        plotscale();
        cleargraph();
        umouse.show(TRUE);
}

void trend::graphplot(float pvin,float cvin,float spin,float samtm)
{
        #define KIND 1   // have 3 graphs is 1:pvin 2:cvin 3:spin
        #define PVCL LIGHTGREEN
        #define CVCL LIGHTCYAN
        #define SPCL LIGHTRED
        #define TGAP 16

        extern int cutfloat(float);

        float ppvin ,ccvin,sspin;

        str15 pvmes,cvmes,spmes,smmes;

        int  pv0,cv0,sp0,pv1,cv1,sp1;
        boolean  pvfg,cvfg,spfg;

        setviewport(0,0,getmaxx(),getmaxy(),1);
        float widthtime = gphposx2 - gphposx1;
        float widthampl = gphposy2 - gphposy1;
        float renge     = hilimt   - lowlimt;
        float chgratio  = widthampl/renge;  //ratio of value and graph
        float dotpertm  = (maxtime*60)/(widthtime*KIND);  //dot per secound
        // adjust samtm for each graph
        oldsmtime += samtm;
        if(oldsmtime < dotpertm) return;
        samtm = oldsmtime;
        oldsmtime = 0.0;
        sprintf(pvmes,"%5.1f",pvin);
        sprintf(cvmes,"%5.1f",cvin);
        sprintf(spmes,"%5.1f",spin);
```

เอกสารนี้จัดทำขึ้นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
sprintf(smmes,"%5.2fs",samtm);
settextjustify(LEFT_TEXT,CENTER_TEXT);
setcolor(trendcl);
outtextxy(gphposx2+(4*magnitude),gphposy1+TGAP,  "ÛÛÛÛÛÛ");
outtextxy(gphposx2+(4*magnitude),gphposy1+3*TGAP,"ÛÛÛÛÛÛ");
outtextxy(gphposx2+(4*magnitude),gphposy1+5*TGAP,"ÛÛÛÛÛÛ");
outtextxy(gphposx2+(4*magnitude),gphposy1+7*TGAP,"ÛÛÛÛÛÛ");
setcolor(PVCL);
outtextxy(gphposx2+(4*magnitude),gphposy1," PV");
outtextxy(gphposx2+(4*magnitude),gphposy1+TGAP,pvmes);
setcolor(CVCL);
outtextxy(gphposx2+(4*magnitude),gphposy1+2*TGAP," CV");
outtextxy(gphposx2+(4*magnitude),gphposy1+3*TGAP,cvmes);
setcolor(SPCL);
outtextxy(gphposx2+(4*magnitude),gphposy1+4*TGAP," SP");
outtextxy(gphposx2+(4*magnitude),gphposy1+5*TGAP,spmes);
setcolor(WHITE);
outtextxy(gphposx2+(4*magnitude),gphposy1+6*TGAP," SM");
outtextxy(gphposx2+(4*magnitude),gphposy1+7*TGAP,smmes);
ppvin = pvin; ccvin = cvin; sspin = spin;
if(pvin > hilimt)  ppvin = hilimt;
if(pvin < lowlimt) ppvin = lowlimt;
if(cvin > hilimt)  ccvin = hilimt;
if(cvin < lowlimt) ccvin = lowlimt;
if(spin > hilimt)  sspin = hilimt;
if(spin < lowlimt) sspin = lowlimt;
float fpospv = (hilimt - ppvin)*chgratio;
float fposcv = (hilimt - ccvin)*chgratio;
float fpossp = (hilimt - sspin)*chgratio;
int ipospv =  cutfloat(fpospv); //----------------
int iposcv =  cutfloat(fposcv);  // value reference at 0.0
int ipossp =  cutfloat(fpossp); //----------------
setlinestyle(SOLID_LINE,0x1,1);
do {
  if(sumdttm <= sumsamtm){
            sumdttm += dotpertm;
            if(sumdttm > maxtime*60){  // if graph is ploted full then
                  taxis += (float)(maxtime*REPEATG);
                  //..
                  int countbuff = timecount*REPEATG;
                  for(int i = 0;i<countbuff;i++)
                              delsavhead();
                  timecount -= countbuff; //reset at cleargraph

                  //..
                  cleargraph();      // plot previosly dot
                  setlinestyle(SOLID_LINE,0x1,1);
                  //..
                  seekdata(pv1,cv1,sp1,pvfg,cvfg,spfg);
                  for(i = oldmaxcount*((1-REPEATG));i > 0;i--){
                        seekdata(pv0,cv0,sp0,pvfg,cvfg,spfg);
                        // plot historic sp
                        if(spfg){
                          if(abs(sp0 - sp1) > 2){
                                setcolor(SPCL);
                                line(gphposx1 + i -1,gphposy1+ sp0,
                                      gphposx1 + i,gphposy1 + sp1);
                              }
                             else putpixel(gphposx1+i,gphposy1+sp1,SPCL);
                        }
                        // plot historic CV
                        if(cvfg){
                          if(abs(cv0 - cv1) > 2){
                                setcolor(CVCL);
                                line(gphposx1 + i -1,gphposy1+ cv0,
                                      gphposx1 + i,gphposy1+ cv1);
                          }
                          else putpixel(gphposx1+i,gphposy1+ cv1,CVCL);
                        }
                        // plot historic PV
                        if(pvfg){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        if(abs(pv0 - pv1) > 2){
                                setcolor(PVCL);
                                line(gphposx1 + i -1,gphposy1+ pv0,
                                        gphposx1 + i,gphposy1+ pv1);
                        }
                        else putpixel(gphposx1+i,gphposy1+pv1,PVCL);
                      }
                      pv1 = pv0;
                      cv1 = cv0;
                      sp1 = sp0;
                }// for
                deltodata(oldmaxcount*REPEATG);
                sumdttm = sumsamtm = (maxtime*60)*(1-REPEATG);//repeat 85%
                xcount = oldmaxcount*(1-REPEATG);  //repeat plot at 85%
        }

        // keep value
        pushdata(ipospv,iposcv,ipossp,pvflag,cvflag,spflag);

        // draw real time data

        if(xcount > 0) {

        // draw line or dot
                        // draw SP
        if(spflag){
                if(abs(opossp - ipossp) > 2){
                        setcolor(SPCL);
                        line(gphposx1+xcount-1,gphposy1+opossp,
                                gphposx1+ xcount, gphposy1+ipossp);
                }
                else putpixel(gphposx1+xcount,gphposy1+ipossp,SPCL);
        }
                        // draw CV
        if(cvflag){
                if(abs(oposcv - iposcv) > 2){
                        setcolor(CVCL);
                        line(gphposx1+xcount-1,gphposy1+oposcv,
                                gphposx1 + xcount,gphposy1+iposcv);
                }
                else putpixel(gphposx1+xcount,gphposy1+iposcv,CVCL);
        }
                        // draw PV
        if(pvflag){
                if(abs(opospv - ipospv) > 2){
                        setcolor(PVCL);
                        line(gphposx1+xcount-1,gphposy1 + opospv,
                                gphposx1 + xcount,gphposy1 + ipospv);
                }
                else putpixel(gphposx1+xcount,gphposy1+ipospv,PVCL);
        }
        }//if
        opospv = ipospv;
        oposcv = iposcv;
        opossp = ipossp;
        xcount++;
  } // if

  if(sumdttm > sumsamtm){   // if sum of dot time of graph more than
        sumsamtm += samtm;    //   sum of samtm input  then out.
        addsav(pvin,cvin,spin,samtm);  //link list
        timecount ++;
        break;
  }
 }while(TRUE);

}

void trend::plotscale(void)
{
        #define SML 2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define MDL 3
#define LGL 4

int among;
int nline;

str15 mes;
setviewport(0,0,getmaxx(),getmaxy(),1);
setcolor(WHITE);
settextjustify(RIGHT_TEXT,CENTER_TEXT);
if(magnitude <= 1) settextstyle(SMALL_FONT,HORIZ_DIR,2);
else settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
int diflgl   =  (gphposy2-gphposy1)/4;
for(int i = 0;i < 5;i++){
        int betw = gphposy2-i*diflgl;
        setlinestyle(SOLID_LINE, 0x1,1);
        if((i % 2) == 0) {
                line(gphposx1 - LGL*magnitude,betw, gphposx1,betw);
                line(gphposx2,betw, gphposx2 + LGL*magnitude,betw);
        }
        else {
                line(gphposx1 - MDL*magnitude,betw, gphposx1,betw);
                line(gphposx2,betw, gphposx2 + MDL*magnitude,betw);
        }
        switch(i) {
                case 0:sprintf(mes,"%d",lowlimt);break;
                case 1:sprintf(mes,"%d",lowlimt + (hilimt-lowlimt)/4);break;
                case 2:sprintf(mes,"%d",lowlimt + (hilimt-lowlimt)/2);break;
                case 3:sprintf(mes,"%d",lowlimt + (hilimt-lowlimt)*3/4);break;
                case 4:sprintf(mes,"%d",hilimt);break;
        }
        outtextxy(gphposx1 - (LGL+1)*magnitude,betw,mes);
} //for
for(i = 9;i >= 2;i--){
        if((diflgl % i) == 0){
                among = i;
                break;
        }
        else among = 0;
}
if(among != 0){
        nline = (gphposy2- gphposy1)/among;
        setlinestyle(SOLID_LINE,0x1,1);
        for(i = 0;i< nline;i++) {
                line(gphposx1 - SML*magnitude,gphposy2 - i*among,
                        gphposx1,gphposy2- i*among);
                line(gphposx2,gphposy2- i*among,
                        gphposx2 + SML*magnitude,gphposy2- i*among);
        }
}
//
// make X - scale
//
diflgl   =  (gphposx2-gphposx1)/4;
for(i = 0;i < 5;i++){
        int betw = gphposx1+i*diflgl;
        setlinestyle(SOLID_LINE, 0x1,1);
        if((i % 2) == 0) {  //odd line
                line(betw, gphposy1 - LGL*magnitude,betw, gphposy1);
                line(betw, gphposy2,betw, gphposy2+ LGL*magnitude);
        }
        else {          //even line
                line(betw,gphposy1 - MDL*magnitude,betw, gphposy1);
                line(betw,gphposy2,betw, gphposy2+ MDL*magnitude);
        }
        switch(i) {
                case 0:sprintf(mes,"%6.2f",0.0);break;
                case 1:sprintf(mes,"%6.2f",maxtime/4);break;
                case 2:sprintf(mes,"%6.2f",maxtime/2);break;
                case 3:sprintf(mes,"%6.2f",maxtime*3/4);break;
                case 4:sprintf(mes,"%6.2f",maxtime);break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                }
                settextjustify(CENTER_TEXT,RIGHT_TEXT);
                outtextxy(betw ,gphposy2+ (LGL+1)*magnitude,mes);
        } //for
        for(i = 9;i >= 2;i--){
                if((diflgl % i) == 0){
                        among = i;
                        break;
                }
                else among = 0;
        }
        if(among != 0){
                nline = (gphposx2 - gphposx1)/among;
                setlinestyle(SOLID_LINE,0x1,1);
                for(i = 0;i< nline;i++) {
                        line(gphposx1 + i*among, gphposy1 - SML*magnitude,
                                gphposx1 + i*among,gphposy1);
                        line(gphposx1 + i*among, gphposy2,
                                gphposx1 + i*among, gphposy2+ SML*magnitude);
                }
        }
}

void trend::value_box(int x,int y,str15 tagname,str15 engunit){

    extern void  box_creatl(int,int,int,int,int,int);
#define  HIGH  50
#define  LONG  600
#define  WIDE  16


const undertxt undbox  =  {DARKGRAY,
                                WHITE,
                                YELLOW,
                                BLACK,
                                9,
                                " EXIT ",
                                " -1000 ",
                                " -100 ",
                                " -10 ",
                                " -1 ",
                                " +1 ",
                                " +10 ",
                                " +100 ",
                                " +1000 "};


    float pvv = 0.0;
    float cvv = 0.0;
    float spv = 0.0;
    float samtv = 0.0;
    float totalsam = 0.0;
    float tdot;
    int  gdot;
    int  vcount = 0;
    boolean exitfg = FALSE;
    str15 pvvst,cvvst,spvst,smvst,totalst,countst;

    // define underbar
    underbar barbox(undbox);
    barbox.setact(bartype1);   //all TRUE
    barbox.show();

    // main bound
    box_creatl(x,y,x+LONG,y+HIGH,LIGHTGRAY,WHITE);

    // data bound
    setcolor(WHITE);outtextxy(x+20,y+10,"Tag:");
    setcolor(BLACK);outtextxy(x+60,y+10,tagname);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
setcolor(WHITE);outtextxy(x+(LONG/5)*2    ,y+10,"ENG/units:");
setcolor(BLACK);outtextxy(x+(LONG/5)*2+100 ,y+10,engunit);
setcolor(WHITE);outtextxy(x+(LONG/5)*4,y+10     ,"Count:");
setcolor(WHITE);outtextxy(x+(LONG/5)*0,y+HIGH-WIDE+3," X:");
setcolor(WHITE);outtextxy(x+(LONG/5)*1,y+HIGH-WIDE+3,"SM:");
setcolor(WHITE);outtextxy(x+(LONG/5)*2,y+HIGH-WIDE+3,"PV:");
setcolor(WHITE);outtextxy(x+(LONG/5)*3,y+HIGH-WIDE+3,"CV:");
setcolor(WHITE);outtextxy(x+(LONG/5)*4,y+HIGH-WIDE+3,"SP:");

//time per dot in x-axis(time axis)
tdot  = (maxtime*60)/((float)(gphposx2 - gphposx1));

do{
      setfillstyle(SOLID_FILL,GREEN);
      bar(x+(LONG/5)*4+50,y+5         ,x+(LONG/5)*5-8,y+WIDE+5);//count
      bar(x+(LONG/5)*0+25,y+HIGH-WIDE-3,x+(LONG/5)*1-8,y+HIGH-3);//X
      bar(x+(LONG/5)*1+25,y+HIGH-WIDE-3,x+(LONG/5)*2-8,y+HIGH-3);//SM
      bar(x+(LONG/5)*2+25,y+HIGH-WIDE-3,x+(LONG/5)*3-8,y+HIGH-3);//PV
      bar(x+(LONG/5)*3+25,y+HIGH-WIDE-3,x+(LONG/5)*4-8,y+HIGH-3);//CV
      bar(x+(LONG/5)*4+25,y+HIGH-WIDE-3,x+(LONG/5)*5-8,y+HIGH-3);//SP

      // find value
      //..
      totalsam = 0.0;
      for(int i = 0;i < vcount;i++)
            if(findsav(i,pvv,cvv,spv,samtv))
                  totalsam += samtv;

      if(!findsav(vcount,pvv,cvv,spv,samtv))
                        pvv = cvv = spv = samtv = 0.0;

      sprintf(pvvst,"%5.1f",pvv);
      sprintf(cvvst,"%5.1f",cvv);
      sprintf(spvst,"%5.1f",spv);
      sprintf(smvst,"%6.3fsec",samtv);
      sprintf(totalst,"%6.3fmin",totalsam/60.0);
      sprintf(countst,"%5d",vcount);

      // put value on screen
      setcolor(DARKGRAY);
      outtextxy(x+(LONG/5)*4+50,y+10        ,countst);
      outtextxy(x+(LONG/5)*0+35,y+HIGH-WIDE+3,totalst);
      outtextxy(x+(LONG/5)*1+35,y+HIGH-WIDE+3,smvst);
      outtextxy(x+(LONG/5)*2+35,y+HIGH-WIDE+3,pvvst);
      outtextxy(x+(LONG/5)*3+35,y+HIGH-WIDE+3,cvvst);
      outtextxy(x+(LONG/5)*4+35,y+HIGH-WIDE+3,spvst);

      barbox.baroff();
      barbox.baron();

      //put cursor
            gdot  = (int)(totalsam/tdot);
      for( i = 0;i <= (gphposy2-gphposy1)/2;i++){
            getpix[i]    =  getpixel(gphposx1+gdot,gphposy1+i*2);
            putpixel(gphposx1+gdot,gphposy1+i*2,YELLOW);
      }
      //..
      do{
            barbox.select();
      }while(barbox.exith == FALSE);

      // get previous
      for( i = 0;i <= (gphposy2-gphposy1)/2;i++){
            putpixel(gphposx1+gdot,gphposy1+i*2,getpix[i]);
      }


      switch(barbox.statusund){
       case 0:exitfg = TRUE ;
       case 1:vcount = vcount - 1000; break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                case 2:vcount = vcount - 100; break;
                case 3:vcount = vcount - 10; break;
                case 4:vcount = vcount - 1; break;

                case 5:vcount = vcount +1; break;
                case 6:vcount = vcount +10; break;
                case 7:vcount = vcount +100;  break;
                case 8:vcount = vcount + 1000;break;
                }
                if(vcount < 0) vcount = 0;
                if(vcount+1 >= timecount) vcount = timecount-2;
        }while(!exitfg);
        barbox.baroff();
        setviewport(x,y,x+LONG,y+HIGH,1);
        clearviewport();
        setviewport(0,0,getmaxx(),getmaxy(),1);

}
//=========================
// other use routine =
//=========================

int cutfloat(float ft)
{
        double integer;
        double factional;
        factional = modf((double)ft,&integer);
        if(factional < 0.5) return(floor((double)ft));
        else return(ceil((double)ft));
}

void box_creatl(int x1,int y1,int x2,int y2,int incolor,int lcl){
//  creat window have bondary
        umouse.show(FALSE);
        setfillstyle(SOLID_FILL,incolor);
        bar(x1,y1,x2,y2);
        int oldcolor = getcolor();
        setlinestyle(SOLID_LINE ,1,1);
        setcolor(lcl);
        line(x1,y1,x2,y1);
        line(x2,y1,x2,y2);
        line(x2,y2,x1,y2);
        line(x1,y2,x1,y1);
        setcolor(oldcolor);
        umouse.show(TRUE);
}
//==== end trendrec.cpp =====


// T$Sintr.cpp
/*****************************************************/
/*  timer and serial interupt program routine  */
/*****************************************************/

#include <dos.h>

#define TRUE 1
#define FALSE 0

typedef int BOOL;

const  int baudrate[8] = { 0,0x300,0x180,0x0C0,0x060,0x030,0x018,0x00C};
const  int baudhi[8]   = { 0,0x3  ,0x1  ,0x0  ,0x0  ,0x0  ,0x0  ,0x0  };
const  int baudlo[8]   = { 0,0x000,0x080,0x0C0,0x060,0x030,0x018,0x00C};
                                        //0,  150,  300,  600, 1200, 2400, 4800, 9600 baudrate
const  int comaddr[5]  = { 0x3F8,0x3F8,0x2F8,0x2F8,0x2F8};
                                        // 00    COM1, COM2, 00  , 00
int baudconst  = 2;
int comconst   = 2;
void initserial(void)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
        outport(comaddr[comconst]+3,0x080);  //Set DLAB = 1 for set Baud
        outport(comaddr[comconst],baudlo[baudconst]);
        outport(comaddr[comconst]+1,baudhi[baudconst]);
        outport(comaddr[comconst]+3,0x03); //0000,0011 No parity,8-bit,1-stop
}

BOOL dataready(void)
{
        int buf = inport(comaddr[comconst]+5);
        if((buf&0x01)==0x01) return(TRUE);   // if Data in Buf TRUE
        else return(FALSE);
}

BOOL thold(void)
{
        int buf = inport(comaddr[comconst]+5);
        if((buf&0x20) == 0x20) return(TRUE);   // if Data in Buf TRUE
        else return(FALSE);
}

int serialrecv(void)
{
        int buf = inport(comaddr[comconst]);
        return(buf);
}

void serialsend(int datasr)
{
        outport(comaddr[comconst],datasr);
}

//======= routine timer interupt=======

#define INTR 0X1C    /* The clock tick interrupt */

#ifdef __cplusplus
        #define __CPPARGS ...
#else
        #define __CPPARGS
#endif

unsigned long int samplingcount = 0;
void interrupt ( *oldfunc)(__CPPARGS);

void interrupt timerintr(__CPPARGS)
{
       samplingcount++;
}

void timerstart(void)
{
        oldfunc = getvect(INTR);        //save old interupt vecter
        setvect(INTR,timerintr);
}

void timerstop(void)
{
        setvect(INTR,oldfunc);
}
//=== end T&SINTR.CPP ===


// bargrph.cpp
/*****************************************/
/* Routine crate bar graph of controllor */
/*****************************************/

#include <graphics.h>
#define  REFSIZE   155
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define BRATIO   0.9  // Y/X

typedef char str15[15];

class bartrend{
        public:
     int posx , posy ,posx2,posy2, size;
     int hmax,hmin;
     int wincl,txtcl;
     str15  uname, hname;
          int barx1 , bary1, barx2,bary2;
          bartrend(int x,int y, int sz,int wc,int tc):
                posx(x),posy(y),size(sz),wincl(wc),txtcl(tc) {}
         ~bartrend(){}
          void barcreat(str15 unm,str15 hnm,int hmn,int hmx);
     void plotbar(float pv,float cv,float sp);
     void clearbar(void);
};

void bartrend::barcreat(str15 unm,str15 hnm,int hmn,int hmx)
{
     #define HGAP  8
     #define BPERWY 0.6
     #define BPERWX 0.3
     #define SHIFTX 0.3
     #define SHIFTY 0.1
     #define LS    4
     #define MS    3
          windobj bgrph;

     strcpy(uname,unm);
     strcpy(hname,hnm);
     hmin  = hmn;
          hmax  = hmx;

          umouse.show(FALSE);
          setviewport(0,0,getmaxx(),getmaxy(),1);
          posx2 = posx + size*REFSIZE;
          posy2 = posy + size*REFSIZE*BRATIO;
          if(posx2 > getmaxx()) posx2 = getmaxx();
          if(posy2 > getmaxy()) posy2 = getmaxy();
          int midx  = posx + size*REFSIZE/2;
          int midy  = posy + size*REFSIZE*BRATIO/2;
          bgrph.window_creatl(posx,posy,posx2,posy2,wincl,txtcl);
          barx1  =  midx - (size*REFSIZE/2)*BPERWX + size*REFSIZE*SHIFTX/2;
          barx2  =  midx + (size*REFSIZE/2)*BPERWX + size*REFSIZE*SHIFTX/2;
          bary1  =  midy - (size*REFSIZE/2)*BPERWY - size*REFSIZE*SHIFTY/2;
          bary2  =  midy + (size*REFSIZE/2)*BPERWY - size*REFSIZE*SHIFTY/2;
          bgrph.window_creatl(posx,posy,posx2,posy2,wincl,WHITE);
          settextjustify(CENTER_TEXT,CENTER_TEXT);
          setcolor(txtcl);
          settextstyle(DEFAULT_FONT,VERT_DIR,1);
          outtextxy(posx+(size*HGAP*2),midy,hname);
          settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
          outtextxy(midx,posy+(size*HGAP),uname);
          line(barx1-1,bary1,barx1-1,bary2);
          line(barx2+1,bary1,barx2+1,bary2);
          int difls = (bary2 - bary1)/2;
          char *mesg;
          for(int i = 0;i<3;i++) {
                line(barx1-1-LS,bary2 - i*difls,barx1-1,bary2 - i*difls);
                line(barx2+1+LS,bary2 - i*difls,barx2+1,bary2 - i*difls);
                switch(i) {
                     case  0: sprintf(mesg,"%d",hmin);
                                settextjustify(RIGHT_TEXT,LEFT_TEXT);
                                break;
                     case  1: sprintf(mesg,"%d",(hmax-hmin)/2);
                                settextjustify(RIGHT_TEXT,CENTER_TEXT);
                                break;
                     case  2: sprintf(mesg,"%d",hmax);
                                settextjustify(RIGHT_TEXT,RIGHT_TEXT);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        break;
                }
                outtextxy(barx1-1-LS,bary2 - i*difls,mesg);
        }
        for(i = 10;i > 2;i--)
                if(difls % i == 0) {
                        for(int j = 0;j < (bary2 - bary1)/i; j++) {
                                line(barx1-1-MS,bary2 - j*i,barx1-1,bary2 - j*i);
                                line(barx2+1+MS,bary2 - j*i,barx2+1,bary2 - j*i);
                        }
                        break;
                } // if
        clearbar();
        umouse.show(TRUE);
}

void bartrend::plotbar(float pv,float cv,float sp)
{
        #define HNUM  10
        #define HVAR  20
        #define PVCL  LIGHTGREEN
        #define CVCL  LIGHTCYAN
        #define SPCL  LIGHTRED

        umouse.show(FALSE);
        setviewport(0,0,getmaxx(),getmaxy(),1);
        str15 mesg[3],bank;
        strcpy(bank,"ÛÛÛÛÛÛÛ");        // alt 219
        int diftext = (posy2 - posy)/3;  // text is in a quarter range
        setcolor(wincl);
        settextjustify(RIGHT_TEXT,CENTER_TEXT);
        for(int i= 0;i < 3;i++)
                outtextxy(posx2-2-(i*diftext),posy2-HNUM,bank);
        sprintf(mesg[2],"%6.1f",pv);
        sprintf(mesg[1],"%6.1f",cv);
        sprintf(mesg[0],"%6.1f",sp);
        setcolor(SPCL);
        outtextxy(posx2-(0*diftext),posy2-(size*HNUM),mesg[0]);
        outtextxy(posx2-(0*diftext),posy2-(size*HVAR),"SP ");
        setcolor(CVCL);
        outtextxy(posx2-(1*diftext),posy2-(size*HNUM),mesg[1]);
        outtextxy(posx2-(1*diftext),posy2-(size*HVAR),"CV ");
        setcolor(PVCL);
        outtextxy(posx2-(2*diftext),posy2-(size*HNUM),mesg[2]);
        outtextxy(posx2-(2*diftext),posy2-(size*HVAR),"PV ");
        int ymax = hmax;  // hmax is not access for any thing
  int ymin = hmin;  // hmin is not access for any thing
  int barxx1 = barx1;
  int barxx2 = barx2;
  int baryy1 = bary1;
  int baryy2 = bary2;
  if(pv > ymax) pv = ymax;
  if(pv < ymin) pv = ymin;
  if(cv > ymax) cv = ymax;
  if(cv < ymin) cv = ymin;
  if(sp > ymax) sp = ymax;
        if(sp < ymin) sp = ymin;
        float gprs;
        if(ymax > ymin)
                gprs = (float)(baryy2 - baryy1)/(float)(ymax - ymin);
        else
                gprs = (float)(baryy2 - baryy1);
        int pvx1 = barxx1;
        int pvy1 = baryy2 - (pv - (float)ymin)*gprs;
  int pvx2 = barxx1 + (barxx2-barxx1)/3;     // there's three bars
  int pvy2 = baryy2;
  int cvx1 = pvx2+1;
        int cvy1 = baryy2 - (cv - (float)ymin)*gprs;
        int cvx2 = cvx1  + (barxx2-barxx1)/3;
        int cvy2 = baryy2;
        int spx1 = cvx2+1;
```

เอกสารนี้จัดทำขึ้นสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        int spy1 = baryy2 - (sp - (float)ymin)*gprs;
        int spx2 = spx1  + (barxx2-barxx1)/3-1;
        int spy2 = baryy2;
        clearbar();
        setfillstyle(SOLID_FILL,PVCL);
        bar(pvx1,pvy1,pvx2,pvy2);
        setfillstyle(SOLID_FILL,CVCL);
        bar(cvx1,cvy1,cvx2,cvy2);
        setfillstyle(SOLID_FILL,SPCL);
        bar(spx1,spy1,spx2,spy2);
        umouse.show(TRUE);
}

void bartrend::clearbar(void)
{
        setfillstyle(SOLID_FILL,DARKGRAY);
        bar(barx1,bary1,barx2,bary2);
}


//============== end BARGRPH.CPP ================


// submenu.cpp
/************************************/
/*  Make sub-menu program routine  */
/************************************/

#include <graphics.h>
#include <string.h>
#include <conio.h>
#include <alloc.h>
#include <stdlib.h>
#include <bios.h>
#include <dos.h>
#include "wingraph.cpp"
#include "mousefar.cpp"

#define MAXROW 15
#define TRUE  1
#define FALSE  0
#define IMPXY  16

graphicmouse umouse;
typedef char str25[25];
typedef int  choiceact[MAXROW];

typedef struct  subm {
          int   shardwcl,barcolor,gndcolor;
          int   txtcl,txtbar;
          int   posx,posy,col,row;
          str25  head;
          int   headcl;
          str25  rowtext[MAXROW];
};

class far sub {
        public:
          subm txts;
          choiceact actc;
          int statusrow;
          boolean subactive;
          boolean exits;
          windobj bar,subwindow;
          sub(subm txt):txts(txt)
                  { statusrow = 0;subactive = exits = FALSE;}
          void subact(choiceact actch);
          void showsub(void);
          void countup(void);
          void countdown(void);
          void barshow(void);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        void select(void);
        void subclose(void);
        void massage(char *ch,int mescl);
};

void sub::subact(choiceact actch)
{
        for( int i = 0;i<txts.row;i++) {
             switch (actch[i]) {
             case TRUE:  break;
               case FALSE:  break;
               default:    actch[i] = TRUE;
             }
             actc[i] = actch[i];
        }
}

void sub::countdown(void)
{
       int errcount = 0;
       do {
         errcount++;
         if (statusrow+1 >= txts.row) statusrow = 0;
         else statusrow++;
       }while(actc[statusrow] == FALSE && errcount<MAXROW);
       if (errcount<MAXROW) {
         umouse.show(FALSE);
         bar.window_close();
         barshow();
       }
}

void sub::countup(void)
{
       int errcount = 0;
       do {
             errcount++;
             if (statusrow-1 < 0) statusrow = txts.row-1;
             else statusrow--;
       }while(actc[statusrow] == FALSE && errcount <MAXROW);
       if (errcount<MAXROW) {
             umouse.show(FALSE);
             bar.window_close();
             barshow();
       }
}

void sub::barshow(void)
{
       int oldcolor = getcolor();
       umouse.show(FALSE);
       bar.window_open(txts.posx+IMPXY,
                       txts.posy+16*statusrow+IMPXY,
                       txts.posx+8*txts.col+IMPXY,
                       txts.posy+16*(statusrow+1)+IMPXY);
       bar.window_create( txts.posx+IMPXY,
                       txts.posy+16*statusrow+IMPXY,
                       txts.posx+8*txts.col+IMPXY,
                       txts.posy+16*(statusrow+1)+IMPXY,
                       txts.barcolor);
       setcolor(txts.txtbar);
       outtextxy( txts.posx+8+IMPXY,
                  txts.posy+4+16*(statusrow)+IMPXY,
                  txts.rowtext[statusrow]);
       setcolor(oldcolor);
       umouse.show(TRUE);
}

void sub::showsub(void)
{
       const int  shardowgab =  10;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        int oldcolor = getcolor();
        umouse.show(FALSE);
        settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
        subwindow.window_open(txts.posx,
                              txts.posy,
                              txts.posx+8*txts.col+2*IMPXY+shardowgab,
                                  txts.posy+16*txts.row+2*IMPXY+shardowgab);
        setcolor(txts.shardwcl);
        for(int k = 0;k<txts.col+4;k++) outtextxy(txts.posx+8+8*k,
                                  txts.posy+16*txts.row+2*IMPXY+shardowgab-9,
                              "±");
        for(k= 0;k<txts.row*2+3;k++) outtextxy(
                                          txts.posx+8*txts.col+2*IMPXY+shardowgab-10,
                                              txts.posy+8*k+9,"±");

        subwindow.window_creatch(txts.posx,
                              txts.posy,
                              txts.posx+8*txts.col+2*IMPXY,
                              txts.posy+16*txts.row+2*IMPXY,
                              txts.grdcolor,
                              txts.head,txts.headcl);

        for (int i = 0;i<txts.row;i++) {
                    setcolor(txts.txtcl);
                    outtextxy(txts.posx+8+IMPXY,
                                  txts.posy+(i)*16+4+IMPXY,
                                  txts.rowtext[i]);
                    if (actc[i] == FALSE){
                    str25 grid ="++++++++++++++++++++++";
                    str25 useg = "";
                    strncat(useg,grid,txts.col);
                    setcolor(txts.grdcolor);
                    outtextxy(txts.posx+8+IMPXY,
                                  txts.posy+(i)*16+4+IMPXY,
                                  useg);
                    }
        }
        int errcount = 0;
        while(actc[statusrow] == FALSE && errcount< MAXROW) {
                statusrow++;
                if(statusrow >= txts.row) statusrow = 0;
                errcount++;
        }
        if(errcount < MAXROW ) barshow();
        setcolor(oldcolor);
        umouse.show(TRUE);
}

void sub::select(void)
{
        #define TXTROW 16

        #define RETURN 7181
        #define ESC    283
        #define UP     18432
        #define DOWN   20480
        #define LEFT   19200
        #define RIGHT  19712

        int oldx,oldy;
        do { umouse.show(TRUE);
          subactive = exits = FALSE;
          if(bioskey(1)!=0) {
                int ch = bioskey(0);
                switch (ch) {
                        case RETURN: subactive = TRUE;exits = TRUE; break;
                        case ESC:   subclose(); subactive = FALSE;exits = TRUE; break;
                        case LEFT:  subclose(); subactive = FALSE;exits = TRUE; break;
                        case RIGHT: subclose(); subactive = FALSE;exits = TRUE; break;
                        case UP:    countup(); break;
                        case DOWN:  countdown(); break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            }
        }
        umouse.getposition();  // mouse action chioce
        if(oldx != umouse.mbutton.xpos ||
              oldy != umouse.mbutton.ypos ) {
              if(umouse.mbutton.xpos > txts.posx+IMPXY &&
                    umouse.mbutton.xpos < txts.posx + 8*txts.col+IMPXY){
                        for(int i = 0;i<txts.row;i++)
                        if(umouse.mbutton.ypos > txts.posy+i*TXTROW+IMPXY &&
                          umouse.mbutton.ypos < txts.posy+(i+1)*TXTROW+IMPXY)
                        if(actc[i] == TRUE) {    //active or blind
                              int oldrow = statusrow;
                              statusrow = i;
                              if (statusrow >txts.row) statusrow = txts.row;
                              if(oldrow != statusrow) {  // stop blinking
                                    umouse.show(FALSE);
                                    bar.window_close();
                                    barshow();
                              }
                        }
              }
        oldx = umouse.mbutton.xpos;
        oldy = umouse.mbutton.ypos;
        } //if getposition
        umouse.querybtndn(BUTTONL);  // check press BUTTONL
        if (umouse.mbutton.opcount > 0) {
              if(umouse.mbutton.xpos > txts.posx+IMPXY  &&
                    umouse.mbutton.xpos < txts.posx + 8*txts.col+IMPXY &&
                    umouse.mbutton.ypos > txts.posy+IMPXY  &&
                    umouse.mbutton.ypos < txts.posy + 16*txts.col + IMPXY){
                        for(int i = 0;i<txts.row;i++)
                        if(umouse.mbutton.ypos > txts.posy+i*TXTROW+IMPXY  &&
                          umouse.mbutton.ypos < txts.posy+(i+1)*TXTROW+IMPXY)
                        if(actc[i] == TRUE) {    //active or blind
                              statusrow = i;
                              subactive = TRUE;
                              exits = TRUE;
                              }
                    }
                    else  {
                              subactive = FALSE;
                              exits = TRUE;
                    }
        } //if buttonDn
    umouse.mbutton.opcount =  0;
        umouse.querybtndn(BUTTONR);  // check press BUTTONR
        if (umouse.mbutton.opcount > 0) {
              subactive = FALSE;
              exits = TRUE;
        }
        }while(exits == FALSE );
}

void sub::subclose(void)
{
        umouse.show(FALSE);
        bar.window_close();
        subwindow.window_close();
        umouse.show(TRUE);
}

void sub::massage(str25 ch,int mescl)
{
         windobj masg;
         str25  string;
         itoa(statusrow,string,10);
         umouse.show(FALSE);
         int lenght = strlen(ch)*8+36;
         int hight  = 16*4;
         masg.window_open((getmaxx()-lenght)/2,(getmaxy()-hight)/2,
                                (getmaxx()-lenght)/2 + lenght,
```

เอกสารนี้จัดทำขึ้นเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                (getmaxy()- hight)/2 + hight);
        masg.window_creatsh((getmaxx()-lenght)/2,(getmaxy()-hight)/2,
                                (getmaxx()-lenght)/2 + lenght,
                                (getmaxy()- hight)/2 + hight,
                                mescl, string ,YELLOW,BLUE);
        outtextxy((getmaxx()-lenght)/2+16,(getmaxy()-hight)/2+25,ch);
        umouse.show(TRUE);
        do{
              umouse.querybtndn(BUTTONL);
              if(bioskey(1) != 0)  if(bioskey(0) == 7181 ) break; //PRESS RETURN
        }while(umouse.mbutton.opcount<=0);
        umouse.show(FALSE);
        masg.window_close();
        umouse.show(TRUE);

}
/******* End of SUBMENU.CPP **********/


// underbar.cpp
/*****************************************/
/*  Make the under bar CRT menu routine */
/*****************************************/

#include <graphics.h>
#include <string.h>
#include <bios.h>
#include <dos.h>

#define  MAXUND  11
#define  HIGHUND 16
#define  YYGAP  4
#define  TRUE   1
#define  FALSE  0

typedef  char strg15[15];  /* head of pulldownmenu */
typedef  int  boolean;
typedef  int  acttf[MAXUND];

//=====================
// define activate
//=====================
acttf bartype1 = { TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE };
acttf bartype2 = { TRUE,TRUE,FALSE,TRUE,TRUE,TRUE,FALSE,TRUE,TRUE,TRUE };
acttf bartype3 = { FALSE,TRUE,FALSE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,FALSE };

typedef struct  undertxt {
        int headcl;
        int txthcl;
        int barcl;
        int bartxtcl;
        int nohead;
        strg15  txth[MAXUND];
};

const undertxt untxt = {
                                LIGHTGRAY,
                                RED,
                                YELLOW,
                                BLACK,
                                10,
                                " NO1 ",
                                " NO2 ",
                                " NO3 ",
                                " NO4 ",
                                " NO5 ",
                                " NO6 ",
                                " NO7 ",
                                " NO8 ",
                                " NO9 ",
                                " NO10 " };
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
class far underbar{
  public:
        windobj  barund;
        int      statusund;
        boolean  undactive;
        boolean  exith;
        undertxt txts;
        acttf    tf;        // for active or unactive bar
        underbar(undertxt txt):txts(txt)
                {statusund = 0;undactive = FALSE;exith = FALSE;};
        void setact(acttf ttf);
        void baron(void);
        void baroff(void);
        void lf(void);
        void rt(void);
        void show(void);
        void select(void);
};

//=======================
// Show highlight bar
//=======================
void underbar::baron(void) {
        umouse.show(FALSE);
        int oldlen = 0;
        setviewport(0,0,getmaxx(),getmaxy(),1);
        settextjustify(LEFT_TEXT,RIGHT_TEXT);
        for(;;){
              if(tf[statusund] == TRUE) break;
              else{ statusund ++;
                      if(statusund > txts.nohead) statusund = 0;
                    }
        }
        for(int i = 0 ;i<statusund;i++) oldlen += strlen(txts.txth[i])*8;
        barund.window_open(oldlen,getmaxy()-HIGHUND,
                                   oldlen+strlen(txts.txth[statusund])*8,
                                   getmaxy());
        barund.window_create(oldlen,getmaxy()-HIGHUND,
                    oldlen+strlen(txts.txth[statusund])*8,
                                   getmaxy(),txts.barcl);
        setcolor(txts.bartxtcl);
        outtextxy(oldlen,getmaxy() - HIGHUND + YYGAP,txts.txth[statusund]);
        umouse.show(TRUE);
}

//=======================
// remove highligh bar
//=======================
void underbar::baroff(void)
{
       umouse.show(FALSE);
       barund.window_close();
       umouse.show(TRUE);
}

//=======================
// shift left bar
//=======================
void underbar::lf(void)
{
        baroff();
        int count = 0;
        do {
        count++;
        if (statusund - 1 < 0) statusund = txts.nohead - 1;
        else statusund -= 1;
        }while(tf[statusund] == FALSE && count < 20);
        if(tf[statusund] == TRUE) baron();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//=====================
// shift right bar
//=====================
void underbar::rt(void)
{
        baroff();
        int count = 0;
        do {
        count++;
        if (statusund + 1 >= txts.nohead) statusund = 0;
        else statusund += 1 ;
        }while(tf[statusund] == FALSE && count < 20);
        if(tf[statusund] == TRUE) baron();
}

//=====================
// menu selection
//=====================
void underbar::select(void)
{
       #define RETURN 7181
       #define ESC   283
       #define LEFT   19200
       #define RIGHT  19712
         umouse.show(TRUE);
         exith = FALSE;undactive = FALSE;
         if(bioskey(1)!=0) {
               int ch = bioskey(0);
               switch (ch) {
                     case RETURN:  baroff(); exith = TRUE; undactive = TRUE; break;
                     case  ESC:  baroff(); exith = TRUE; break;
                     case  LEFT:  lf(); break;
                     case  RIGHT:  rt(); break;
               }
         }
         umouse.querybtndn(BUTTONL);  // mouse action chioce
         if(umouse.mbutton.opcount > 0 ) {
                if( umouse.mbutton.ypos > getmaxy() - HIGHUND) {
                           int lenght = 0;
                           for(int i = 0;i<txts.nohead;i++) {
                             lenght += strlen(txts.txth[i])*8;
                             if(umouse.mbutton.xpos < lenght) {
                                        if(tf[i] == TRUE) {
                                                statusund = i;
                                                baroff();
                                                baron();
                                                delay(50);
                                                baroff();
                                                undactive = TRUE;
                                                exith = TRUE;
                                        }
                                   break;
                             }
                           } //for
               }
         }
}

//=====================================
// show under bar menu in lower bound of CRT
//=====================================
void underbar::show(void)
{
        umouse.show(FALSE);
        strg15 str1;
        strg15 str2 = "+++++++++++++";
        setviewport(0,0,getmaxx(),getmaxy(),1);
        settextjustify(LEFT_TEXT,RIGHT_TEXT);
        setfillstyle(SOLID_FILL,txts.headcl);
        bar(1,getmaxy()-HIGHUND,getmaxx(),getmaxy());
        setcolor(LIGHTRED);
```

เอกสารนี้จัดทำขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        setlinestyle(SOLID_LINE,1,1);
        rectangle(1,getmaxy()-HIGHUND,getmaxx(),getmaxy());
        int oldlen = 0;
        for(int i=0;i<txts.nohead;i++) {
            setcolor(txts.txthcl);
            outtextxy(oldlen,getmaxy()-12,txts.txth[i]);
            if(tf[i] == FALSE){
                strncpy(str1,str2,strlen(txts.txth[i]));
                setcolor(txts,headcl);
                outtextxy(oldlen,getmaxy()-12,str1);
            }
            oldlen += strlen(txts.txth[i])*8;
        } //for
        umouse.show(TRUE);
}

//=====================
// set active of each
//=====================
void underbar::setact(acttf ttf)
{
        for(int i = 0;i<txts.nohead;i++)
            tf[i]  = ttf[i];
        undactive = FALSE;
        exith     = FALSE;
        show();
}


// ===== end UNDERBAR.CPP =======


// wingraph.cpp
/*************************************/
/*  Pop-up window  program routine  */
/*************************************/

# include <stdio.h>
# include <stdlib.h>
# include <alloc.h>
# include <graphics.h>
# include <string.h>
# include <math.h>


class window_block {
      public:
            window_block  *_back;
            unsigned int posx,posy;
            int _X1,_Y1,_X2,_Y2;
            unsigned int sizecontent;
            int   *contentptr;
            int viewx1,viewy1,viewx2,viewy2,viewclip;
};

class  windobj:public window_block {
      public:
      window_block *lastlink;
      windobj() { lastlink = new window_block; lastlink = NULL; }
      ~windobj() { delete lastlink;}
      void window_create(int x1,int y1,int x2,int y2,int incolor);
      void window_creatl(int x1,int y1,int x2,int y2,int incolor,int lcl);
      void window_creatch(int x1,int y1,int x2,int y2,
                                    int incolor,char *head,int chcolor);
      void window_creatsh(int xx1,int yy1,int xx2,int yy2,
                                    int incolor,char *head,int chcolor,int shardwcl);
      void window_open(int x1,int y1,int x2,int y2);
      void window_close();
};
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//=========================================
// window haven't bondary creation
//=========================================
void windobj::window_create(int x1,int y1,int  x2,int y2,int incolor)
{
        setfillstyle(SOLID_FILL,incolor);
        bar(x1,y1,x2,y2);
}

//=========================================
//  creat window  have bondary
//=========================================
void windobj::window_creatl(int x1,int y1,int x2,int y2,int incolor,int lcl)
{
        setfillstyle(SOLID_FILL,incolor);
        bar(x1,y1,x2,y2);
        int  oldcolor = getcolor();
        setlinestyle(SOLID_LINE ,1,1);
        setcolor(lcl);
        rectangle(x1,y1,x2,y2);
        setcolor(oldcolor);
}

//=========================================
//   window have text head
//=========================================
void windobj::window_creatch(int x1,int y1,int x2,int y2,
                             int incolor,char *head,int chcolor)
{
        const int  high = 8;
        setfillstyle(SOLID_FILL,incolor);
        bar(x1,y1,x2,y2);
        int oldcolor = getcolor();
        setcolor(chcolor);
        settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
        settextjustify(LEFT_TEXT,RIGHT_TEXT);
        int lenstr = strlen(head)*8;
        int posc = (x2-x1-lenstr)/2;
        if (posc <= 0 ){ posc =2*high; lenstr = 0;}
        else outtextxy(x1+posc,y1+high-3,head);
        setlinestyle(SOLID_LINE ,1,1);
        line(x1+high,y1+high,x1+posc-1,y1+high);
        line(x1+posc+lenstr-2,y1+high,x2-high,y1+high);
        line(x2-high,y1+high,x2-high,y2-high);
        line(x2-high,y2-high,x1+high,y2-high);
        line(x1+high,y2-high,x1+high,y1+high);
        setcolor(oldcolor);
}

//=========================================
// window  have shawdow
//=========================================
void windobj::window_creatsh(int xx1,int yy1,int xx2,int yy2,
                             int incolor,char *head,int chcolor,int shardwcl)
{
        int x1 = xx1;
        int y1 = yy1;
        int x2 = xx2-8;
        int y2 = yy2-10;
        const int  high = 8;
        setcolor(shardwcl);   // make sharwdow
        settextstyle(DEFAULT_FONT,HORIZ_DIR,1);
        settextjustify(LEFT_TEXT,RIGHT_TEXT);
        for(int k = 0;k<ceil((x2-x1)/8);k++) outtextxy(x2-8*k,y2,"±");
        for(k= 0;k<ceil((y2-y1)/8);k++) outtextxy(x2,y2-8*k,"±");
        setfillstyle(SOLID_FILL,incolor);
        bar(x1,y1,x2,y2);
        int oldcolor = getcolor();
        setcolor(chcolor);
        int lenstr = strlen(head)*8;
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        int posc = (x2-x1-lenstr)/2;
        if (posc <= 0 ){ posc =2*high; lenstr = 0;}
        else outtextxy(x1+posc,y1+high-3,head);
        setlinestyle(SOLID_LINE ,1,1);
        line(x1+high,y1+high,x1+posc-1,y1+high);
        line(x1+posc+lenstr-2,y1+high,x2-high,y1+high);
        line(x2-high,y1+high,x2-high,y2-high);
        line(x2-high,y2-high,x1+high,y2-high);
        line(x1+high,y2-high,x1+high,y1+high);
        setcolor(oldcolor);
}

void windobj::window_open(int x1,int y1,int x2,int y2)
{
   struct viewporttype viewinfo;
        window_block  *newlink;
        if(!(newlink = new window_block)){
                closegraph();
                printf("Error in window_block allocation");
                exit(1);
        }
        unsigned     isize;
              int       *icontentptr;
        isize  = imagesize(x1,y1,x2,y2);
        if((icontentptr = (int*)malloc(isize)) == NULL) {
                closegraph();
                printf(" Saved imagesize in memory error ");
                exit(1);
        };
        getviewsettings(&viewinfo);       //getviewport
        /* alocated  memory of picture */
        newlink->sizecontent = isize;
        newlink->posx = getx();
        newlink->posy = gety();
        newlink->_X1 = x1;
        newlink->_Y1 = y1;
        newlink->_X2 = x2;
        newlink->_Y2 = y2;
        newlink->viewx1   = viewinfo.left;
        newlink->viewy1   = viewinfo.top;
        newlink->viewx2   = viewinfo.right;
        newlink->viewy2   = viewinfo.bottom;
        newlink->viewclip = viewinfo.clip;
        newlink->contentptr = icontentptr;
        getimage(x1,y1,x2,y2,icontentptr);
        if (lastlink == NULL){
         lastlink  =  newlink;
         lastlink->_back = NULL;
        }
        else {
              newlink->_back = lastlink;
              lastlink       = newlink;
        }
}

void windobj::window_close(){
        struct window_block  *current;
        if (lastlink != NULL) {
        current = lastlink;
        lastlink = current->_back;
        setviewport(current->viewx1,current->viewy1,
                                current->viewx2,current->viewy2,current->viewclip);
        putimage(current->_X1,  current->_Y1,current->contentptr,0);
        moveto(current->posx,current->posy);
        free(current->contentptr);
        free(current);
        }
}
//========= end WINGRAPH.CPP =========
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// mouseobj.cpp
/*****************************/
/*  mouse program routine  */
/*****************************/

#include <stdio.h>
#include <dos.h>
#include <graphics.h>

typedef int boolean;

      struct far position {
            int  btnstat,
                  opcount,
                  xpos,ypos;
  };
  struct gcursor {
                  int screenmask[16],
                        cursormask[16];
                  int hotx,hoty;
  };
#define   BUTTONL   0
#define   BUTTONR   1
#define   BUTTONM   2
#define   SOLFWARE  0
#define   HARDWARE  1
#define   TRUE      1
#define   FALSE     0

//=============================================
//=      five graphics cursors are predefined   =
//=          for usr with GraphicMouse          =
//=============================================
gcursor arrow = {    // default graphics cursor
                  0x1FFF, 0x0FFF, 0x07FF, 0x03FF, //screenmask
                  0x01FF, 0x00FF, 0x007F, 0x003F,
                  0x001F, 0x003F, 0x01FF, 0x01FF,
                  0xE0FF, 0xF0FF, 0xF8FF, 0xF8FF,
                  0x0000, 0x4000, 0x6000, 0x7000, //cursormask
                  0x7800, 0x7C00, 0x7E00, 0x7F00,
                  0x7F80, 0x0300, 0x4C00, 0x0600,
                  0x0600, 0x0300, 0x0300, 0x0000,
                  0x0001,  //hotx
                  0x0001}; //hoty
gcursor     check    = {    // check mask graphics cursor
                  0xFFF0, 0xFFE0, 0xFFC0, 0xFF81, //screenmask
                  0xFF03, 0x0607, 0x000F, 0x001F,
                  0x803F, 0xC07F, 0xE0FF, 0xF1FF,
                  0xFFFF, 0xFFFF, 0xFFFF, 0xFFFF,
                  0x0000, 0x0006, 0x000C, 0x0018, //cursormask
                  0x0030, 0x0060, 0x70C0, 0x3908,
                  0x1F00, 0x0E00, 0x0400, 0x0000,
                  0x0000, 0x0000, 0x0000, 0x0000,
                  0x0005,  //hotx
                  0x0010}; //hoty
gcursor     cross    ={  //circle with cross hairs
                  0xF01F, 0xE0FF, 0xC007, 0x8003, //screenmask
                  0x0441, 0x0C61, 0x0381, 0x0381,
                  0x0381, 0x0C61, 0x0441, 0x8003,
                  0xC007, 0xE00F, 0xF01F, 0xFFFF,
                  0x0000, 0x070C, 0x0920, 0x1110, //cursormask
                  0x2108, 0x4004, 0x4004, 0x783C,
                  0x4004, 0x4004, 0x2108, 0x1110,
                  0x0902, 0x07C0, 0x0000, 0x0000,
                  0x0007,  //hotx
                  0x0007}; //hoty
gcursor     glove = {      //glove or hand image cursor}
                  0xF3FF, 0xE1FF, 0xE1FF, 0xE1FF, //screenmask
                  0xE1FF, 0xE049, 0xE000, 0x8000,
                  0x0000, 0x0000, 0x07FC, 0x07F8,
                  0x9FF9, 0x8FF1, 0xC003, 0xE007, //cursormask
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                0x0C00, 0x1200, 0x1200, 0x1200,
                                0x1200, 0x13B6, 0x1249, 0x7249,
                                0x4002, 0x4002, 0x2004, 0x1FF8,
                                0x0004,  //hotx
                                0x0000}; //hoty
 gcursor   ibeam  = {
                                0xF3FF, 0xE1FF, 0xE1FF, 0xE1FF, //screenmask
                                0xE1FF, 0xE049, 0xE000, 0x8000,
                                0x0000, 0x0000, 0x07FC, 0x07F8,
                                0x9FF9, 0x8FF1, 0xC003, 0xF007,
                                0x0C30, 0x0240, 0x0180, 0x0180, //cursormask
                                0x0180, 0x0180, 0x0180, 0x0180,
                                0x0180, 0x0180, 0x0180, 0x0180,
                                0x0180, 0x0180, 0x0240, 0x0C30,
                                0x0000, //hotx
                                0x0007}; //hoty
 class genmouse {
             public:
               int xmove,ymove;
               position mbutton;
               int btncount;
               boolean status;
               boolean visible;
                   boolean testmouse(void);
                   void  setaccel(int threshold);
                   void  show(boolean option);
                   void  getposition(void);
                   void  querybtndn(int button);
                   void  querybtnup(int botton);
                   void  readmove(void);
                   void  reset(void);
                   void  setratio(int horpix,int verpix);
                   void  setlimits(int xposmin,int yposmin,int xposmax,int yposmax);
               void  setposition(int xpos,int ypos);
               };

 class  graphicmouse:public genmouse {
         public:
               void conditionalhide(int left,int top,int right,int bottom);
               void setcursor(gcursor);
               void initialize(gcursor);
 };

 class  textmouse:public genmouse {
         public:
               void initialize();
               void setcursor(int ctype,int c1,int c2);
 };

 class  graphiclightpen:public graphicmouse {
               void lightpen(boolean option);
 };

 class  textlightpen:public textmouse {
               void lightpen(boolean option);
 };

 //=============================================
 //=  used in mouse routine     =
 //=============================================
 int lower(int n1,int n2){ return((n1 < n2) ? n1 : n2);}
 int upper(int n1,int n2){ return((n1 > n2) ? n1 : n2);}

 //=============================================================
 //= implementation mothods for GeneralMouse       =
 //=============================================================
 boolean  genmouse::testmouse(void)
 {
       const   int iret=207;
       unsigned int far  *doff;
       unsigned int far  *dseg;
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                doff = (unsigned int far*)MK_FP(0000,0204);
                dseg = (unsigned int far*)MK_FP(0000,0206);
                if((dseg==0) || (doff==0)) return(FALSE);
                else return((int)MK_FP(*dseg,*doff) != iret);
}

void genmouse::reset(void)
{
                _AX = 0x00;          //reset to default condition
                geninterrupt(0x33);
                status  = _AX != 0;   //Mouse present
                btncount = _BX;       //botton count }
}

void genmouse::setaccel(int threshold)
{
                _AX = 0x13;
                _DX = threshold;
                geninterrupt( 0x33);
}

void genmouse::show(boolean option)
{
                if (option &&(!visible)) {
                        _AX   = 0x01;  // show mouse cursor
                        visible = TRUE;
                        geninterrupt( 0x33);
                }
                else if (visible && (!option)) {
                        _AX   = 0x02;   // Hide mouse cursor }
                        visible = FALSE;
                        geninterrupt( 0x33);
                }
}

void  genmouse::getposition(void)
{
                _AX = 0x03;
                geninterrupt( 0x33);
                mbutton.btnstat  = _BX;
                mbutton.xpos     = _CX;
                mbutton.ypos     = _DX;
}

void genmouse::setposition(int xco,int yco )
{
                _AX  = 0x04;
                _CX  = xco;
                _DX  = yco;
                    geninterrupt( 0x33);
}

void genmouse::setratio(int horpix,int verpix)
{
                    _AX  = 0x0F;
                    _CX  = horpix;  //horizontal mickeys/pixel
                    _DX  = verpix;  //vertical mickeys/pixel
                    geninterrupt( 0x33);
}

void genmouse::querybtndn(int button )
{
                _AX = 0x05;
                    _BX = button;     //BUTTONL ,BUTTONM,BUTTONR
                    geninterrupt( 0x33);
                    mbutton.btnstat = _AX;
                    mbutton.opcount = _BX;
                    mbutton.xpos  = _CX;
                mbutton.ypos  = _DX;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void genmouse::querybtnup(int botton)
{
                _AX =  0x06;
                _BX = botton;
                genihterrupt( 0x33);
                mbutton.btnstat = _AX;
                mbutton.opcount = _BX;
                mbutton.xpos   = _CX;
                mbutton.ypos   = _DX;
}

void genmouse::setlimits(int xposmin,int yposmin,int xposmax,int yposmax)
{
                _AX = 0x07;
                _CX = lower(xposmin,xposmax);
                _DX = upper(xposmin,xposmax);
                geninterrupt( 0x33);
                _AX = 0x08;
                _CX = lower(yposmin,yposmax);
                _DX = upper(yposmin,yposmax);
                geninterrupt( 0x33);
}

void genmouse::readmove(void)
{
                _AX  = 0x0B;
                geninterrupt( 0x33);
                xmove = _CX;
                ymove = _DX;
}

//=============================================================}
//  implementation method for GraphicsMouse          }
//=============================================================}

void graphicmouse::setcursor(gcursor gcur)
{
                _AX = 0x09;
                _BX = gcur.hotx;
                _CX = gcur.hoty;
                _DX = FP_OFF(gcur.screenmask);
                _ES = FP_SEG(gcur.screenmask);
                geninterrupt( 0x33);
}

void graphicmouse::conditionalhide(int left,int top,int right,int bottom)
{
                _AX = 0x0A;
                _CX = left;
                _DX = top;
                _SI = right;
                _DI = bottom;
                geninterrupt( 0x33);
}

void graphicmouse::initialize(gcursor newg)
{
                genmouse::reset();
                if(status) {
                    visible = FALSE;
                    setlimits(0,0,getmaxx()-10,getmaxy()-10);
                    setposition(getmaxx()/2,getmaxy()/2);
                }
                setcursor(newg);
//              genmouse::show(TRUE);
}

//=============================================================}
// implementtation method for TextMouse          }
//=============================================================}
```

สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void textmouse::initialize(void)
{
        genmouse::reset();
        if(status) {
            visible=FALSE;
            setlimits( 1, 1,80,25);
            setcursor(HARDWARE,6,7);
            setposition(0,0);
            show(TRUE);
        }
}

void textmouse::setcursor(int ctype,int c1,int c2)
{
        _AX = 0x0A;       // function 10h
        _BX = ctype;      // 0=software / 1=hardware
        _CX = c1;         //sceen mask or scan stat line
        _DX = c2;         //cursor mask or scan stop line
        geninterrupt(0x33);
}

//=============================================================
// implementation method for TextLightPen        =
//=============================================================

void textlightpen::lightpen(boolean option)
{
        if (option) _AX = 0x0D;    //turn pen on
        else _AX = 0x0E;           //turn pen off
        geninterrupt( 0x33);
}

//=============================================================}
// implementation method for GraphicLightPen      }
//=============================================================}

void graphiclightpen::lightpen(boolean option)
{
        if (option) _AX = 0x0D;    //turn pen on
        else _AX = 0x0E;           //turn pen off
        geninterrupt( 0x33);
}

//=============================
//= end. MOUSEOBJ.CPP =
//=============================
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อผู้เขียน	นาย เจตนา สิงห์สินธุ์
วัน เดือน ปี เกิด	31 มกราคม พ.ศ. 2508
สถานที่เกิด	โรงพยาบาลเด็ก กรุงเทพฯ
ระดับการศึกษา	ปริญญาตรีวิทยาศาสตรบัณฑิต สาขาฟิสิกส์
สถานศึกษาจบ	มหาวิทยาลัยรามคำแหง
ปีจบการศึกษา	พ.ศ. 2531
เข้าศึกษาต่อระดับ	ปริญญาวิศวกรรมศาสตรมหาบัณฑิต
ปีการศึกษาต่อ	พ.ศ. 2531
สถานศึกษาต่อ	สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง

ผลงานทางวิชาการ

1. "ระบบเซอร์โวควบคุมตำแหน่งด้วยเทคนิคของเฟสล็อกลูป" วารสารการประชุมวิชาการวิศวกรรมไฟฟ้าครั้งที่ 12 , พ.ศ. 2532
2. "การพัฒนาตัวควบคุม PFD เพื่อปรับปรุงการล็อคในระบบเซอร์โวควบคุมตำแหน่งให้เร็วขึ้น " วารสารการประชุมวิชาการวิศวกรรมไฟฟ้าครั้งที่ 13 , พ.ศ. 2533

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้