ปีการศึกษา 2531 ภาคการเรียนที่2

การจำลองการควบคุมกระบวนการ

(Process control simulation)

โดย

กนกเวทย์ ตั้งพิมลรัตน์

ชยากร เลี้ยงรื่นรมย์

ชัชพล ประสพโชค

อาจารย์ที่ปรึกษา

อ. สุเธียร เกียรติสุนทร

023099 -4 ส.ค. 2532

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจำลองการควบคุมกระบวนการ

กนกเวทย์ ตั้งพิมลรัตน์
ชยากร เลี้ยงรื่นรมย์
ชัชพล ประสพโชค
อาจารย์สุเธียร เกียรติสุนทร อาจารย์ที่ปรึกษา
ปีการศึกษา 2531

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ได้เรียบเรียงขึ้น เพื่ออธิบายถึงโครงงานที่ได้พัฒนาขึ้นมาเพื่อใช้ในการจำลองการควบคุมกระบวนการแบบง่ายๆ โดยเป็นกระบวนการพื้นฐานทั่วๆไปที่มีการควบคุมระดับของของเหลว ที่มีการต่อกันแบบนอนอินเตอร์แอกติ้ง (Non-interacting) โดยลักษณะการจำลองจะทำการจำลองกระบวนการเป็นซอฟแวร์ ซึ่งแทนสมการคุณลักษณะของกระบวนการนั้นๆบนเครื่องคอมพิวเตอร์ ไอบีเอ็ม พีซี ทั่วๆไป โดยจะรับสัญญาณอินพุทที่ส่งมาจากตัวควบคุมภายนอก(PID-controller) ผ่านวงจรปรับระดับสัญญาณโวลเตจ และ วงจรแปลงสัญญาณอนาลอคเป็นดิจิตอล เพื่อนำค่ามาทำการประมวลผล ซึ่งผลลัพธ์ที่ได้จะส่งผ่านวงจรแปลงสัญญาณดิจิตอลเป็นอนาลอค , วงจรปรับระดับสัญญาณโวลเตจ ไปยังตัวควบคุมภายนอกอีกครั้งในลักษณะลูปปิด เพื่อแสดงสภาวะของกระบวนการที่เวลานั้นๆ

สำหรับโปรแกรมที่ใช้งานเขียนด้วย ภาษาเบสิค(Quick basic Vers. 4) โดยโปรแกรมสามารถเก็บค่าพารามิเตอร์และรูปแบบของกระบวนการไว้ในดิสเก็ตต์ ซึ่งสามารถนำกลับมาใช้ใหม่ได้อีก นอกจากนั้นการทำงานยังมีการแสดงภาพของกระบวนการในโหมดกราฟิค ที่ใช้การ์ดเฮอร์คิวลิส สำหรับขอบเขตในการจำลองจะทำการจำลองเพียงกระบวนการแบบ นอนอินเตอร์แอกติ้ง ที่มี 1 อินพุท 1 เอาท์พุทเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
1.1 การจำลองกระบวนการ	1
1.2 รูปแบบ	2
1.3 การวิเคราะห์หาทรานสเฟอร์ฟังค์ชั่น และ ผลตอบสนองทางเวลา	4
1.4 การคำนวณสมการไดนามิคดีฟเฟอเรนติเอท	7
บทที่ 2 การเชื่อมต่อทางด้านฮาร์ดแวร์	12
2.1 วงจรปรับระดับโวลเตจ	13
2.2 วงจรแปลงกระแสเป็นโวลเตจ	14
2.3 วงจรแปลงโวลเตจเป็นกระแส	15
2.4 วงจรแปลงสัญญาณอนาลอคเป็นดิจิตอลและแปลงดิจิตอลเป็นอนาลอค	16
2.5 ตัวควบคุม	17
บทที่ 3 ซอฟแวร์	19
บทที่ 4 ผลการทดลอง	31
บทที่ 5 บทสรุป และ วิจารณ์	35
สรุปการทำปริญญานิพนธ์	35
แนวทางในการปรับปรุง	36
ภาคผนวก	37
ภาคผนวก ก. โปรแกรมการทำงาน	
ภาคผนวก ข. หลักการในการปรับค่าตัวควบคุม	
ภาคผนวก ค. คุณลักษณะของตัวควบคุมภายนอกที่ใช้ในการทดลอง	
กิตติกรรมประกาศ	
หนังสืออ้างอิง	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROCESS CONTROL SIMULATION

KANOKVATE TUNGPIMOLRUT

CHAYAKORN LEANGRUENROM

CHATCHAPHOL PRASOPCHOKE

SUTHEAN KAITSUNTHORN ADVISOR

1988

ABSTRACT

This thesis describes about project that uses in Process Control Simulation. The simulated Process is fundamental and general process that used in industry . Expecially , Level-control that connect in Non-interacting style. The process will be simulated as a software process on computer IBM PC. Simulated software comprises of the relation ,parameter and the characteristic equation of process . Input signal , from outter Controller , will pass V/V converter circuit and Analog to Digital converter for processing by Computer. The result will be sent through Digital to Analog converter and V/V converter to outter Controller again in closed-loop style,and display the status of process at that time too .

This program is built in Basic Programming (Quick Basic Version 4).It will store all parameter and form of the process into diskette .Expecially, displaying the proces picture must use in GRAPHICS MODE which HERCULIS CARD .Simulation in this thesis concerns with the NON-INTERACTING PROCESS that has 1 inputand 1 output only.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ และ ทฤษฎี

ในภาวะที่อุตสาหกรรมมีการขยายตัวอย่างมากในปัจจุบัน โดยเฉพาะการขยายตัวทางด้านกว้างๆ ดังนั้นกระบวนการในการผลิตต่างๆจึงต่างออกไปจากอดีตมาก โดยจะมีรูปแบบและวิธีการที่แปลกใหม่มากมาย และเป็นกระบวนการที่สลับซับซ้อนยิ่งขึ้น ดังนั้นงานในการควบคุมกระบวนการ (Process Control) ให้ได้ผลผลิตตามที่ตนต้องการ จึงมีความสำคัญมากขึ้นและจำเป็นอย่างยิ่งที่จะต้องมีพนักงาน (Operator) ที่เข้าใจและสามารถใช้ตัวควบคุมได้อย่างมีประสิทธิภาพ เพื่อมิให้กระบวนการเกิดการเสียหาย ซึ่งพนักงานนั้นต้องมีประสพการณ์ และ ทักษะที่ดีในเรื่องเกี่ยวกับกระบวนการนั้นๆ สำหรับพนักงานใหม่ จึงจำเป็นที่จะต้องมีการอบรมเสียก่อน ซึ่งโดยปกติจะอบรมกับกระบวนการที่ได้จำลองขึ้นมาก่อน ซึ่งจะมีขนาดที่เล็กกว่าและเป็นการจำลองทางด้านฮารดแวร์ เพื่อให้การศึกษาและการทดลองทำได้สะดวกขึ้น แต่อย่างไรก็ตามในการจำลองดังกล่าวก็ยังเสียค่าใช้จ่ายไม่น้อยเหมือนกัน และหากมีการเปลี่ยนแปลงกระบวนการใหม่ระบบที่จำลองขึ้นมาก็จะใช้ไม่ได้อีก ดังนั้นในโครงงานนี้จึงมีความคิดที่จะทำการจำลองกระบวนการทางด้านซอฟแวร์แทน โดยใช้ไมโครคอมพิวเตอร์ ไอบีเอ็ม พีซี ทั่วๆไป เป็นตัวประมวลผลตามสมการคุณลักษณะของกระบวนการ ซึ่งสะดวกกว่าการจำลองทางด้านฮารดแวร์ โดยที่ผู้ใช้สามารถที่จะกำหนดพารามิเตอร์ต่างๆของกระบวนการได้เอง และจากผลการทดลองเมื่อเทียบกับกระบวนการจริงๆ จะให้ผลที่เหมือนกัน มีค่าคลาดเคลื่อนน้อยมาก

1.1 การจำลองกระบวนการ

การศึกษาถึงกระบวนการผลิตในอุตสาหกรรมนั้น เป็นไปได้ยากที่เราจะทราบคุณลักษณะ และผลตอบสนองของกระบวนการนั้นได้ทุกสภาวะ ทั้งนี้เพราะในกระบวนการผลิตจริงแล้วผลตอบสนองของกระบวนการ จะขึ้นกับตัวแปรหลายๆอย่างพร้อมกัน มิใช่ตัวแปรใดตัวแปรหนึ่งโดยเฉพาะ เช่น อุณหภูมิ ความดัน อัตราการไหล และเชื้อเพลิงที่ป้อน เป็นต้น ดังนั้นในการที่จะควบคุมกระบวนการผลิต ให้ได้ผลตามที่ต้องการ หมายความว่า เราจะต้องสามารถควบคุมทุกๆองค์ประกอบได้เป็นอย่างดี ดังนั้นการจะควบคุมกระบวนการนั้น จะต้องมีพื้นฐานความเข้าใจในทุกๆส่วนของกระบวนการ รวมทั้งสามารถที่จะปรับตัวควบคุมให้สอดคล้องกับสภาวะที่เปลี่ยนแปลงไป ซึ่งอาจจะทำอันตรายแก่กระบวนการได้ ดังนั้นจึงมีความจำเป็นที่จะต้องมีการศึกษา และ ฝึก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัด เกี่ยวกับกระบวนการในขั้นต้นเสียก่อน แต่ในการศึกษาและทำการทดลองกับกระบวนการจริงๆ อาจเป็นไปได้ยาก ทั้งนี้เพราะ

1. เกิดความไม่สะดวกในการเปลี่ยนแปลงค่าพารามิเตอร์ต่างๆของกระบวนการ และใช้ต้นทุนที่สูง เช่นในการเปลี่ยนแปลงค่าความจุของแทงค์ (Capacity of Tank)

2. เกิดความเสี่ยงต่ออันตรายที่เกิดขึ้น และทำให้เครื่องมือที่ใช้เสียหาย ในกรณีที่การทดลองผิดพลาด

3. การเปลี่ยนแปลงแก้ไข กระบวนการทำได้ยาก ไม่สะดวก ใช้เวลามาก ค่าใช้จ่ายสูง

ดังนั้น จากข้อจำกัดดังกล่าว ในการควบคุมกระบวนการจึงใช้ **รูปแบบ** (Model) ในการจำลองแทน ซึ่งให้ข้อดีกว่าในแง่ของความสะดวกสบาย ใช้ได้ในเนื้อที่ที่เล็กกว่า การเก็บข้อมูลได้ภายในดิสเก็ตต์ และสามารถใช้ได้บนเครื่องคอมพิวเตอร์ทั่วๆไป ใช้งบประมาณที่ถูกกว่า นอกจากนั้นยังมีความยืดหยุ่นกว่า

1.2 รูปแบบ (Model)

รูปแบบ คือ ส่วนที่ใช้แนะนำเกี่ยวกับระบบและใช้ศึกษาแทนระบบจริงๆ เนื่องจากมีข้อยุ่งยากมากถ้าหากจะศึกษากับระบบจริงๆ ทั้งนี้เพราะระบบจริงจะมีราคาที่แพง หาได้ยาก และไม่มีความยืดหยุ่น สำหรับการทำรูปแบบของระบบจะแบ่งเป็น 2 ขั้นตอนกว้างๆ คือ

1. สร้าง รูปแบบ : นิยามซอบเขต รูปธรรม (Entities) คุณลักษณะ (Attributes) การกระทำ (Activities)

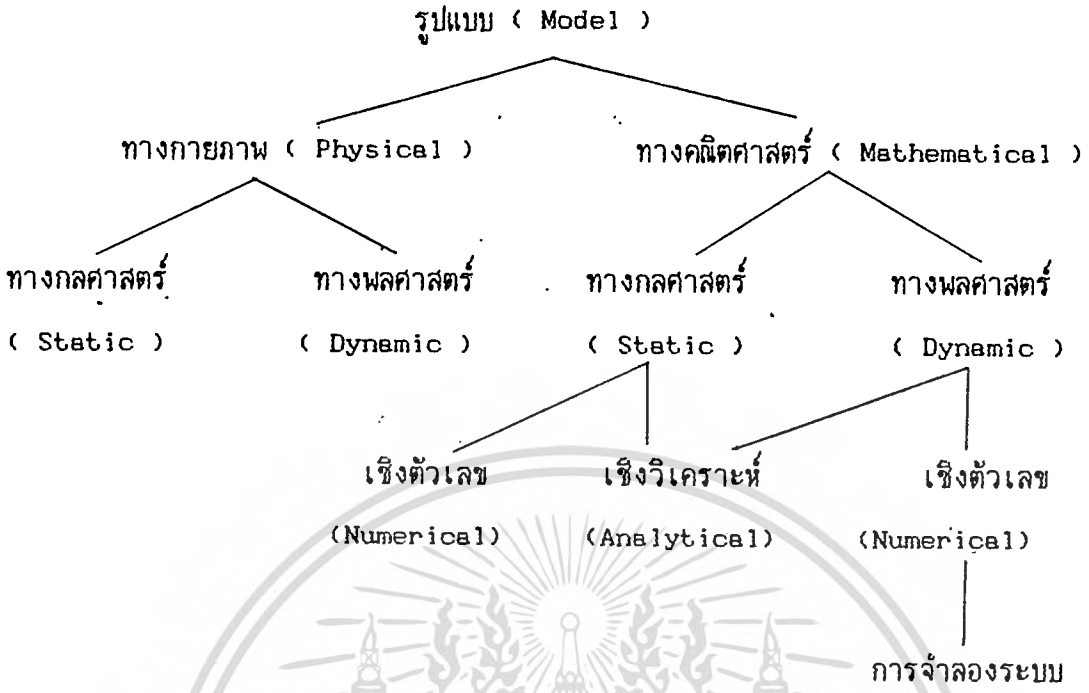
2. ป้อนข้อมูลเข้า :

สำหรับชนิดของรูปแบบ แจกแจงได้ดังรูปที่ 1

* ทางกลศาสตร์(Static) แสดงว่าค่า คุณลักษณะของระบบเมื่อระบบอยู่ในสมดุลย์

* ทางพลศาสตร์ (Dynamic) แสดงว่ามีการติดตามการเปลี่ยนแปลงตลอดเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1

สำหรับการจำลองกระบวนการที่เราใช้ในโครงงาน ใช้รูปแบบชนิด ทางคณิตศาสตร์ โดยจะทำแบบ ทางพลศาสตร์ และใช้วิธีการเชิงตัวเลข เพื่อใช้วิเคราะห์ระบบให้ได้เป็น การจำลองระบบขึ้น ซึ่งจะเขียนแผนผังง่ายๆ ได้ดังนี้

Model =>Mathematical =>Dynamic =>Numerical =>System Simulation

ดังนั้นจะแทนกระบวนการด้วยสมการความสัมพันธ์ทางคณิตศาสตร์ ในลักษณะของทรานสเฟอร์ฟังค์ชั่น (Transfer function) ของอุปกรณ์แต่ละชิ้น แล้วจึงนำมาหาทรานสเฟอร์ฟังค์ชั่นของกระบวนการทั้งหมด จากนั้นจึงหาผลตอบสนองทางเวลา (Time response)ของกระบวนการในลักษณะเชิงตัวเลข ซึ่งสามารถนำค่ามาคำนวณและประมวลผลต่อไปได้ นั่นคือ เมื่อมีสัญญาณอินพุทเข้ามา โปรแกรมจะใช้สัญญาณอินพุทนี้ในการคำนวณหาผลตอบสนองทางเวลา และผลลัพธ์ที่ได้ก็จะถูกส่งกลับไปสู่ตัวควบคุมภายนอกเพื่อทำการปรับค่าให้เข้าสู่ค่าที่ต้องการ นอกจากนั้นผลตอบสนองทางเวลาที่คำนวณได้ ก็จะมาแสดงผลที่มอนิเตอร์ของคอมพิวเตอร์ ในลักษณะของกราฟทางเวลา (Time graph) หรือ กราฟระดับ (Bar graph) ดังนั้นเราจึงต้องทราบถึงความสัมพันธ์พื้นฐาน สำหรับอุปกรณ์ต่างๆในกระบวนการในรูปของทรานสเฟอร์ฟังค์ชั่นเสียก่อน ซึ่งจะกล่าวถึงในหัวข้อถัดไป

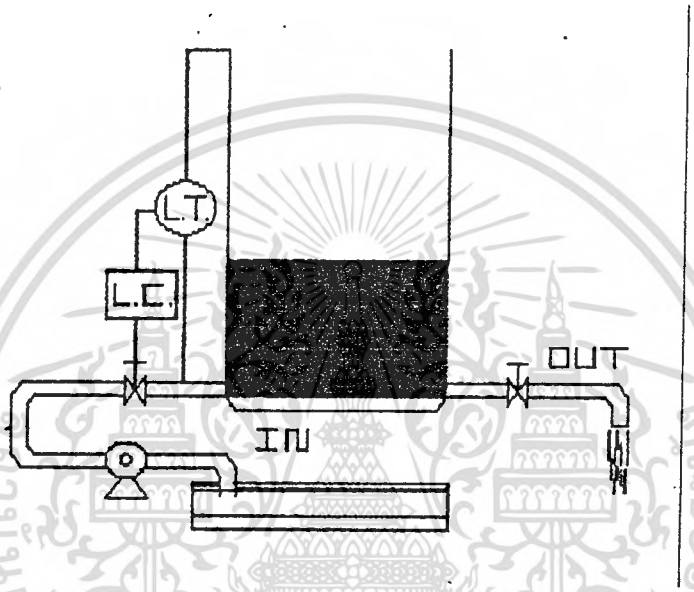
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 การวิเคราะห์หาทรานสเฟอร์ฟังค์ชั่น และ ผลตอบสนองทางเวลา

ในกระบวนการผลิตทั่วๆไป อุปกรณ์ซึ่งเป็นอุปกรณ์พื้นฐานและพบเห็นเป็นประจำ ก็คือ หม้อปฏิกิริยา (Reactor) หรือ แทงค์ และ วาล์ว (Valve) ซึ่งสามารถหาความสัมพันธ์ในรูปของทรานสเฟอร์ฟังค์ชั่น ได้ดังนี้

ก.แทงค์น้ำ (Tank)

ซึ่งมีลักษณะตามรูป



รูปที่ 2

เมื่อ Qin = อัตราการไหลเข้าของของเหลว (m^3 /sec)

H = ระดับความสูงของของเหลวในถัง (m)

R = ความต้านทานการไหล (sec/m^3)

C = พื้นที่หน้าตัดของถัง (m^2)

จากรูปที่ 2 สามารถเขียนเป็นสมการดิฟเฟอเรนเชียลได้ดังนี้

$$Qin - Qo = C * dH/dt \qquad ...(1)$$

$$Qo = H/R \qquad ...(2)$$

แทนค่า Qo ในสมการ (2) ลงในสมการ (1)

$$Qin = C * dH/dt + H/R \qquad ...(3)$$

เมื่อทำการแปลงลาปลาส สมการ (3)

$$Qin(s) = C * S * H(s) + H(s)/R \qquad ...(4)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการ (4) เราสามารถหาทรานสเฟอร์ฟังค์ชั่นของ H(s)/Q(s) ได้ดังนี้

$$H(s)/Q(s) = R/(1+T*S) \quad ...(5)$$

และจากค่าในสมการที่ (2)

$$Qin(s) = H(s)/R \quad ...(6)$$

ดังนั้นจะได้

$$Qo(s)/Qin(s) = 1/(1+T*S) \quad ...(7)$$

สำหรับผลตอบสนองของระดับของของเหลวต่อ สเตปอินพุท Qin(s) =A/S

เมื่อ A = ขนาดของอินพุท

แทนค่าในสมการที่ (7) จะได้

$$H(s) = R*A/(1+T*S)*S \quad ...(8)$$

ใช้วิธีแยกอัตราส่วน จะได้

$$H(s) = [1/S - T/(T+S)]*A*R \quad ...(9)$$

ทำการแปลงอินเวอร์สลาปลาสจะได้

$$H(t) = H(\emptyset) + A*R*(1-e^{(t-t\emptyset)/T}) \quad ...(10)$$

และในทำนองเดียวกัน สมการ(7) สามารถหาผลตอบสนองทางเวลาได้ดังนี้

$$Qo(t) = Qo(\emptyset) + A*(1-e^{(t-t\emptyset)/T}) \quad ...(11)$$

ข. <u>วาล์ว</u> (Valve)

เมื่อพิจารณาถึง วาล์วควบคุม (Control Valve) ในระบบจริงๆ ซึ่งจะทำงานโดยสัญญาณลม(Pneumatic) ที่มีขนาด 1-15 psig สัญญาณนี้จะมาจากตัวควบคุม แต่โดยทั่วไปตัวควบคุมจะส่งสัญญาณมาเป็นค่ากระแส 4-20 mA หรือ ค่าโวลเตจ 1-5 v ดังนั้นจึงต้องมีการแปลงสัญญาณไฟฟ้าไปเป็นสัญญาณลม กรณีนี้จะทำให้วาล์วควบคุมมีคุณสมบัติที่มี เดดไทม์ (Dead Time) ที่สามารถแทนได้ด้วยสมการอนุพันธ์อันดับที่หนึ่ง (First order derivative) ดังนี้

$$Qo(s)/Qin(s) = 1/(T*S+1) \quad ...(12)$$

เมื่อ T = ค่าคงที่ทางเวลา (Time constant)

และจะหาผลตอบสนองทางเวลาได้ คือ

$$Qo(t) = Qo(\emptyset) + A*[1-e^{(t-t\emptyset)/T}] \quad ...(13)$$

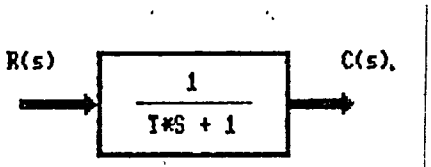
เมื่อ A = ขนาดของอินพุท

สำหรับในโครงงานนี้ได้จำลองอุปกรณ์เพียง 2 แบบคือ แทงค์ กับ วาล์วควบคุมเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลตอบสนองทางเวลาของระบบอันดับหนึ่งต่ออินพุทที่เป็นสเตปฟังค์ชั่น

จากรูปที่ 3 บล็อคไดอะแกรมของระบบอันดับหนึ่ง



รูปที่ 3

ซึ่งจะมีทรานสเฟอร์ฟังค์ชั่นของระบบอันดับหนึ่ง ดังนี้คือ

$$C(s)/R(s) = 1/(Ts+1) \qquad ...(14)$$

เนื่องจากการแปลงลาปลาสของสเตปฟังค์ชั่นหนึ่งหน่วย มีค่าเท่ากับ 1/s ดังนั้นแทนค่าของ R(s) ลงในทรานสเฟอร์ฟังค์ชั่นด้วย 1/s จะได้

$$C(s) = 1/(Ts+1) * 1/s \qquad ...(15)$$

โดยใช้วิธีแยกส่วน (partial fraction)

$$C(s) = 1/s - T/(1+Ts) \qquad ...(16)$$

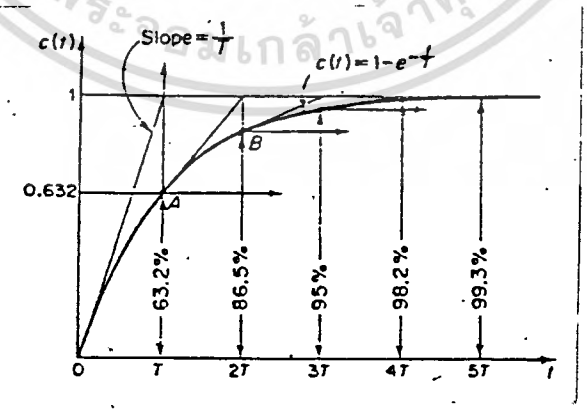
นำ C(s) มาทำการแปลงกลับลาปลาส จะได้ว่า

$$C(t) = [Q(s)] = (1/s - T/(1+Ts)) \qquad ...(17)$$

$$C(t) = 1 - [1/(s + 1/T)] \qquad ...(18)$$

$$C(t) = 1 - e^{-t/T} \quad ;t > 0 \qquad ...(19)$$

ซึ่งเป็นผลตอบสนองต่ออินพุทที่เป็น ยูนิตสเตปฟังค์ชั่น ในโดเมนเวลา สามารถเขียนเส้นโค้ง (curve) ได้ดังนี้



รูปที่ 4

จากสมการที่(19)บอกให้เราทราบว่า ที่ภาวะเริ่มต้น t=0 เอาท์พุทC(t)จะมีค่าเท่ากับ 0 และในที่สุด C(t) จะมีค่าเท่ากับ 1 สำหรับที่ t=T นั้น ค่า C(t) จะมีค่าเท่ากับ 0.632 หรือผลตอบสนองของ C(t) จะมีค่าเข้าสู่ 63.2% ของค่าสุดท้าย ซึ่งจะเห็นได้โดยง่ายจากการแทนค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

t=T ในสมการ(19) นั้นคือ

$$C(t) = 1-e^{-1} = .632$$

ค่า T ซึ่งเป็นค่าคงที่ทางเวลาของระบบ จะเป็นตัวที่ชี้ถึงความเร็วในการตอบสนองของระบบ นั้นคือหากค่า T มีค่าน้อยๆ ความเร็วของผลตอบสนองของระบบจะเร็ว ในทางตรงกันข้าม หากค่า. T มีค่ามากๆ ความเร็วของผลตอบสนองของระบบจะช้า คุณสมบัติที่สำคัญของเส้นสัมผัส (tangent line) ที่ t=0 ซึ่งหาได้จากการหาอนุพันธ์ของสมการที่ (19)

และให้ t=0 นั่นคือ

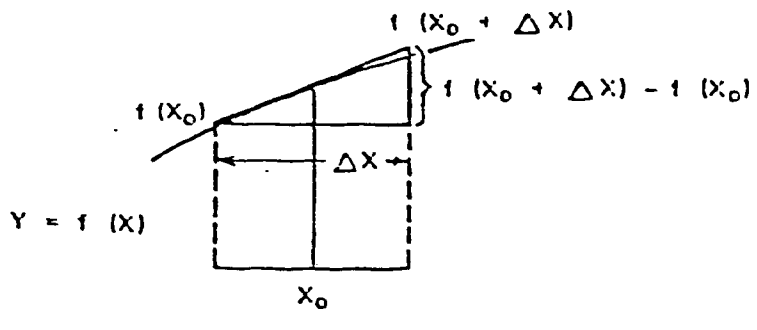
$$dC(t)/dt\Big|_{t=0} = 1/T * e^{-t/T}\Big|_{t=0} = 1/T \quad ...(20)$$

และจากสมการที่ (20) เราจะเห็นว่าความชันของเส้นโค้งของผลตอบสนอง C(t) จะลดลงเรื่อยๆจากค่า 1/T ที่ t=0 ไปเป็น 0 ที่ t= รูปที่ 4 แสดงถึงผลตอบสนองของระบบอันดับหนึ่งต่ออินพุทที่เป็นสเตปฟังค์ชั่น ในช่วงเวลาเท่ากับ 1 คาบค่าคงที่ทางเวลาผลตอบสนองของระบบจะมีค่าจาก 0 ไปจนถึง 63.2% ของค่าสุดท้าย และเมื่อ t=3T,4T,5T ผลตอบสนองมีค่า 95% ,98.2% และ 99.3% ของค่าสุดท้ายตามลำดับ ดังนั้นสำหรับ t > 4T นั้น ผลตอบสนองจะมีค่าอยู่ในช่วง 25 ของค่าสุดท้าย ซึ่งโดยทั่วไปเราถือว่าเมื่อผลตอบสนองมีค่าเข้าสู่ช่วง 2% ของค่าสุดท้าย แสดงว่าได้เข้าสู่สภาวะคงที่ (steady state)แล้ว นั่นก็คือระบบอันดับหนึ่งจะเข้าสู่สภาวะคงที่เมื่อเวลาผ่านไปเท่ากับ 4 ช่วงเวลาของค่าคงที่ทางเวลา

1.4 การคำนวณสมการไดนามิคดิฟเฟอเรนติเอท

นิยามของการดิฟเฟอเรนติเอทมีดังนี้

$$(dy/dx)_{\text{ที่ } x=x_0} = \lim_{x->x_0} f(X_0+ X)-f(X_0) / X \quad ...(21)$$



รูปที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการข้างบนแสดงถึงสโลปของเส้นที่สัมผัสกับพื้นผิว(curve) y=f(x) ที่จุด x0 ดังแสดงในรูปที่ 5 ดังนั้นการประมาณค่าของความชันนี้หาได้เป็น

$$dy/dx = f(x1)-f(x0) / (x1- x0) \quad ...(22)$$

ในกรณีนี้แม้ x=x1-x0 จะมีค่าลดลง การคำนวณด้วยดิจิตอลคอมพิวเตอร์ก็ไม่ได้ทำให้มีความเที่ยงตรงเพิ่มขึ้นแต่ประการใด เนื่องจากค่า f(x1)-f(x0) จะมีค่าลดน้อยลงด้วย ดังนั้นผลลัพธ์ที่ได้จากการคำนวณก็ยังคงห่างไกลค่าที่ถูกต้อง

ในบางกรณีเราสามารถหาค่าที่ถูกต้องเที่ยงตรงของสโลปให้มากขึ้นได้โดยประมาณค่าดิฟเฟอเรนติเอท ถ้าหากเพียงแต่เรารู้ค่าจุดย่อยๆบนพื้นผิวฟังค์ชั่นที่เราจะดิฟเฟอเรนติเอท และต่อจากนั้นเราก็คำนวณหาค่าผลลัพธ์ของสมการพีชคณิต

$$dx/dt = (x[n]-x[n-1])/T[s] \quad ...(23)$$

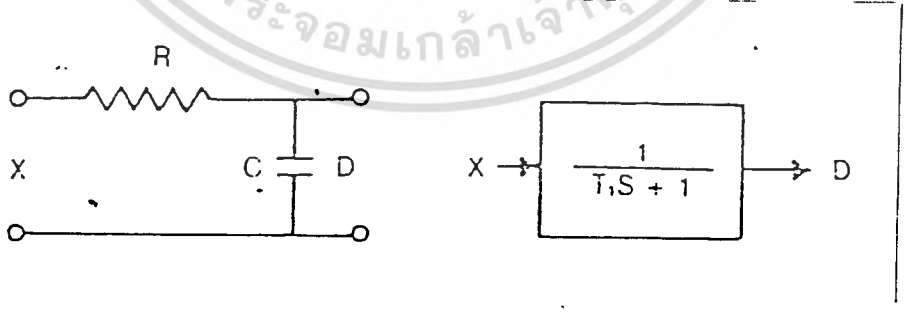
เมื่อ x[n] คือค่าของเอาท์พุท x ที่ระยะเวลาปัจจุบัน

x[n-1] คือค่าของเอาท์พุท x ที่ระยะเวลาก่อนหน้าไปหนึ่งช่วงเวลาการชักตัวอย่าง

T[s] คือระยะเวลาระหว่างการชักตัวอย่าง (Sampling interval)เป็นวินาที

ฉะนั้น $T[s] = T[n]-T[n-1]$ ...(24)

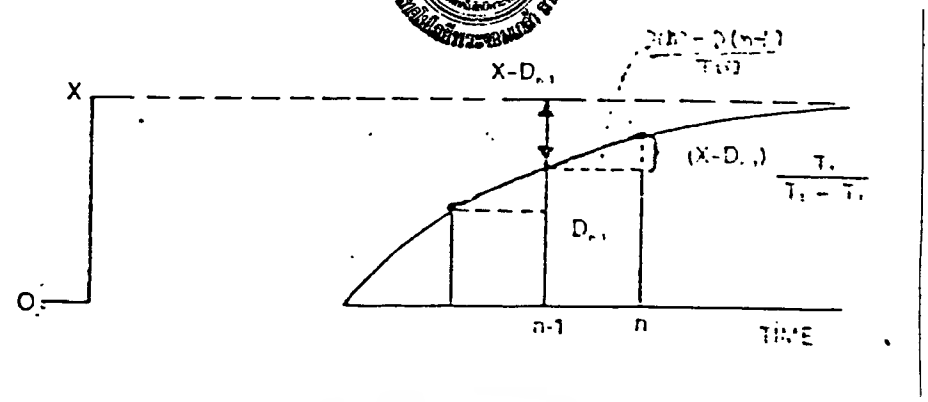
ดังนั้นกระบวนการที่เราได้ทำการหาทรานสเฟอร์ฟังค์ชั่นไว้แล้ว ซึ่งส่วนใหญ่สามารถแทนได้ในรูปของ ระบบอันดับหนึ่งแบบล้าหลัง (First-order lag) โดยจะมีรูปแบบทั่วๆไปทางฮาร์ดแวร์ และทรานสเฟอร์ฟังค์ชั่น ดังแสดงในรูปข้างล่าง



รูปที่ 6

สำหรับผลตอบสนองของสเตปที่เปลี่ยนอย่างทันทีทันใดของโวลตเตจตกคร่อมอินพุทจะทำให้ได้เอาท์พุท D ที่เพิ่มขึ้นเป็นเอ็กโปเนนเชียล ที่มีค่าสโลปที่กำหนดโดยค่าไทม์คอนสแตนท์ RC (T1) ดังแสดงในรูปที่ 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7

ซึ่งเราสามารถพัฒนาการควบคุมสำหรับดิจิตอลไมโครคอมพิวเตอร์ได้ โดยการแปลงสมการดิฟเฟอเรนติเอทแบบต่อเนื่อง ไปเป็นสมการที่เปลี่ยนตามเวลาในรูปขั้นกระโดดขั้นเล็กๆ ด้วยวิธีของ สมการผลต่าง (Difference equation) ซึ่งมีขั้นตอนการหาดิจิตอลอัลกอริทึมดังต่อไปนี้

(1) ให้เขียนรูปแบบของทรานสเฟอร์ฟังค์ชั่นที่หาได้ของกระบวนการ ในฟอร์มของอนาลอค โดยใช้สัญลักษณ์ของลาปลาสทรานสฟอร์ม

(2) เปลี่ยนสัญลักษณ์ของลาปลาส ไปเป็นรูปแบบของสมการดิฟเฟอเรนเชียล

(3) ต่อจากนั้นแทนสมการดิฟเฟอเรนเชียลด้วยสมการผลต่าง โดยใช้การประมาณทางดิจิตอล

$$dD/dx = D_n - D_{n-1} / T_s \qquad \ldots(24)$$

(4) คำนวณหาค่าตัวแปรปัจจุบัน (ที่มีสัญญลักษณ์ห้อยด้วย n) โดยใช้ค่าตัวแปรในอดีตที่อยู่ข้างขวามือของสมการเท่านั้น (สัญญลักษณ์ห้อยด้วย n-i ; เมื่อ i มีค่าเท่ากับหรือ มากกว่า 1)

จากรูปที่ 7 ทรานสเฟอร์ฟังค์ชั่นคือ อัตราส่วนระหว่าง D/X เป็น

$$D/X = 1/(T[1]*S+1) \qquad \ldots(25)$$

จากนั้น

$$T[1]*S*D + D = X \qquad \ldots(26)$$

เนื่องจาก s = d/dt เราสามารถเขียนสมการที่ (26) ใหม่ได้เป็น

$$T[1]*dD/dt + D = X \qquad \ldots(27)$$

สมการที่ (27) เป็นสมการดิฟเฟอเรนเชียล เราจะต้องแปลงสมการ ให้อยู่ในรูปของสมการผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

023099

ต่าง จึงจะนำไปใช้กับดิจิตอลอัลกอริทึมได้ดี ดังนั้นเราต้องแทนเทอมดิริเวทีฟ dD/dt ซึ่งมีค่าโดยประมาณดังนี้

$$dD/dt = [Dn - Dn-1]/T[s] \quad ...(28)$$

เมื่อ Dn คือค่าเอาท์พุท D ที่ระยะเวลาปัจจุบัน

Dn-1 คือค่าของเอาท์พุท D ที่ระยะเวลาก่อนหน้า

T[s] คือระยะเวลาระหว่างจุดการชักตัวอย่าง มีค่าเป็นวินาที

ถ้าแทนสมการที่ (28) ลงในสมการที่(27)เราก็จะสามารถทรานสฟอร์มสมการที่ (27) ไปเป็นสมการผลต่างได้

$$\{T[1](D[n] - D[n-1])/T[s]\} +D[n] = x \quad ...(29)$$

T[1] เป็นค่าคงที่ทางเวลาของวงจรฟิลเตอร์แบบ RC

คำตอบของสมการที่ (29) ที่เป็นจริงหรือที่ใช้งานได้จะมีค่าเป็น

$$D[n] = D[n-1] + \{T[s]/ (T[s]+T[1])\}*(X-D[n-1]) \quad ...(30)$$

ดิจิตอลคอมโปรเซสเซอร์จะคำนวณสมการที่ (30)ซ้ำไปเรื่อยๆ คือคำนวณครั้งหนึ่งๆสำหรับในช่วงการชักตัวอย่างหนึ่งๆ เพื่อรักษาให้เอาท์พุท D[n]เป็นค่าปัจจุบันอยู่เสมอ ด้วยการเพิ่มค่าของ n อินพุทที่ให้แก่การคำนวณนี้คือค่าปัจจุบันของ X (โวลเตจเริ่มต้นเพิ่มขึ้นเป็นค่าสเตป)
ดิจิตอลโปรเซสเซอร์สามารถจำลองการทำงานของอนาลอคฟิลเตอร์แบบ RC ที่แสดงในรูปที่ 6 ได้ด้วยสมการที่ (30)ไปเรื่อยๆ เอาท์พุท D จะมีค่าที่เปลี่ยนไปตามเวลาในรูปของขั้นบันได (step-wise) ในช่วงเวลาที่เป็นช่วงสั้นๆ หนึ่งๆ

--- [n-2] , [n-1] , [n]

ในรูปที่ 7 เราจะเห็นว่าค่า D[ ] จะมีค่าเข้าใกล้ค่า X ไปเรื่อยๆ และจากสมการที่ (30) เราจะเห็นได้ว่า D มีค่าเพิ่มขึ้นในแต่ละช่วง โดยที่เพิ่มขึ้นจากค่าเดิม D[n-1] ด้วยจำนวนที่เท่ากับผลต่าง X-D[n-1]คูณด้วย "ฟิลเตอร์คอนสแต้นท์" ซึ่งมีค่าเท่ากับ T[s]/(T[s]+T[1]) ค่าเอาท์พุทจะเข้าใกล้ค่าสเตปอินพุท X ด้วยอัตราที่กำหนดได้ด้วยค่า RC ไทม์คอนสแตนท์ T[1] ของอนาลอคฟิลเตอร์ เมื่อ T[1] มีค่าน้อยอัตราการเพิ่มขึ้นของเอาท์พุทโวลเตจของฟิลเตอร์ก็จะมากขึ้นด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

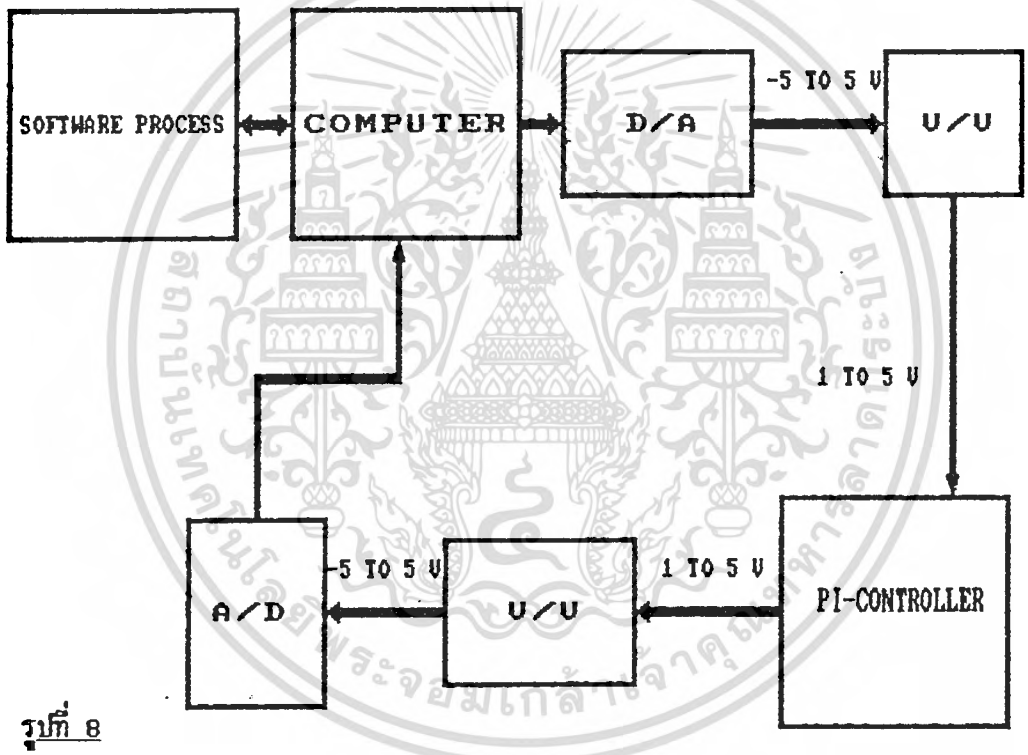
***ตามทฤษฎีของ Shanon : ในการทำงานของดิจิตอลโปรเซสเซอร์ที่ใช้ ตัวแปรที่มีผลโดยตรงกับความถูกต้องในการชักตัวอย่าง ก็คือคาบเวลาในการชักตัวอย่าง T[s] โดยทีเราต้องเลือกค่าทีเหมาะสมเมื่อเทียบกับความถี่ปกติ หรือ ค่าคงที่ทางเวลาของกระบวนการของเรา ซึ่งตามหลักการเราต้องเลือกดให้ค่า T[s] มีค่ามากกว่า 2 เท่าของความถี่ของกระบวนการทีจะทำการชักตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

การเชื่อมต่อทางด้านฮาร์ดแวร์

จากรูปข้างล่างซึ่งแสดงการเชื่อมต่อทั้งหมดที่ใช้ในการจำลอง ซึ่งจะมีกระบวนการเป็นลักษณะของซอฟแวร์ แทนได้ด้วยโปรแกรมที่ป้อนให้กับคอมพิวเตอร์ และตัวควบคุมภายนอก นอกจากนั้นยังแสดงถึงวงจรที่เชื่อมต่อระหว่างคอมพิวเตอร์ และตัวควบคุม เพื่อทำการปรับสัญญาณให้เหมาะสม ทั้งนี้เพราะว่าคอมพิวเตอร์จะประมวลด้วยสัญญาณแบบดิจิตอล แต่ตัวควบคุมจะรับสัญญาณแบบอนาลอค แล้วประมวลผลด้วยสัญญาณดิจิตอล



รูปที่ 8

จากรูปที่ 8 จะพบว่าการเชื่อมต่อระหว่างตัวควบคุมและคอมพิวเตอร์ ประกอบด้วยส่วนต่างๆ ดังนี้

1. วงจรปรับระดับโวลตเตจ (V/V converter)
2. วงจรแปลงกระแสเป็นโวลตเตจ (I/V converter)
3. วงจรแปลงโวลเตจเป็นกระแส (V/I converter)
4. วงจรแปลงสัญญาณอนาลอคเป็นดิจิตอล (A/D converter)
5. วงจรแปลงสัญญาณดิจิตอลเป็นอนาลอค (D/A converter)
6. ตัวควบคุมแบบ PI (PI-controller)

ซึ่งมีรายละเอียดของส่วนต่างๆดังต่อไปนี้

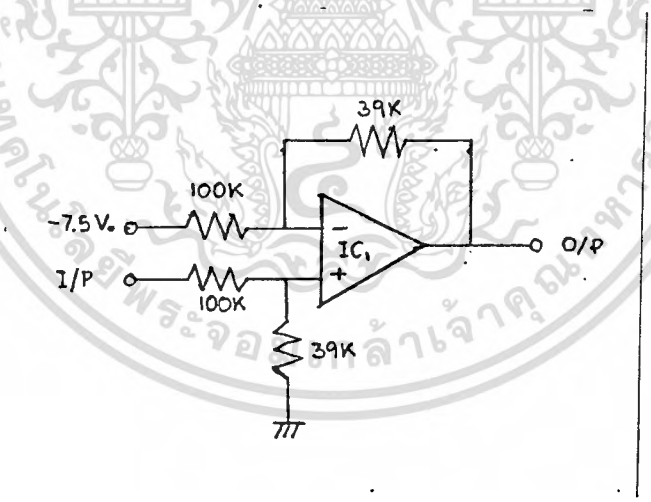
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1 วงจรปรับระดับโวลเตจ

ในการเชื่อมต่อกับตัวควบคุมภายนอกนั้น ตัวควบคุมอาจจะเป็นตัวควบคุมแบบอนาลอค หรือ ตัวควบคุมแบบดิจิตอลก็ได้ ซึ่งจะมีคุณลักษณะของอินพุท-เอาท์พุทที่ต่างกัน ดังนั้น สัญญาณที่รับและส่งมาจากคอมพิวเตอร์ ผ่านวงจรแปลงสัญญาณอนาลอคเป็นดิจิตอล หรือดิจิตอลเป็นอนาลอค ต้องผ่านวงจรที่จะทำการปรับรูปสัญญาณให้เหมาะสมก่อนที่จะผ่านต่อไปยังตัวควบคุมต่อไป สำหรับกรณีที่ตัวควบคุมรับและส่งสัญญาณในรูปของสัญญาณโวลเตจ วงจรที่ต้องใช้จึงเป็นวงจรในการปรับระดับสัญญาณเท่านั้น โดยจะพิจารณาทั้งทางด้านอินพุท และ เอาท์พุท

ทางด้านอินพุท

เป็นสัญญาณอนาลอค ซึ่งมีระดับสัญญาณโวลเตจระหว่าง-5 ถึง+5 โวลต์ ให้เป็นระดับสัญญาณโวลเตจระหว่าง 1 ถึง 5 โวลต์แทน ซึ่งเป็นพิกัดใช้งานของตัวควบคุม โดยมีลักษณะวงจรดังรูปที่ 9



รูปที่ 9

จากวงจร ไอซี 1 ซึ่งต่อในลักษณะของดิฟเฟอร์เรนเชียลแอมป์ (Differential amplifier) เพื่อทำหน้าที่ในการปรับระดับสัญญาณ โดยจะมีเอาท์พุทโวลตเตจ ดังสมการ

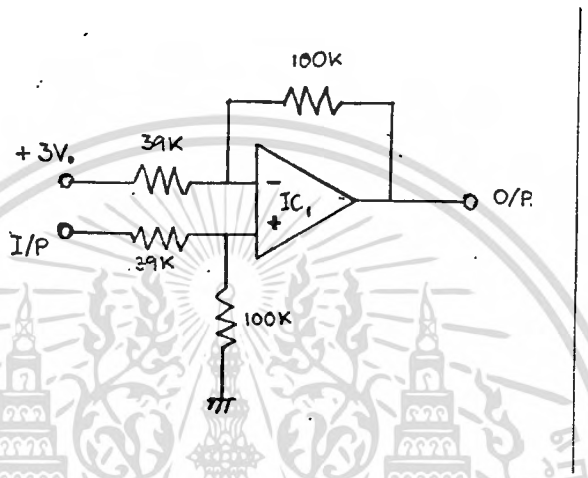
$$V2 = (V1 + 7.5)/2.5 \quad \text{โวลต์}$$

และจะได้ V2 อยู่ในช่วง 1-5 โวลต์ตามต้องการ และเนื่องจากความต้านทานอินพุทขาเข้าของไอซี 1 มีค่ามาก ดังนั้นจะไม่มีปัญหาเรื่องโหลดดิ้งเอฟเฟค (Loading effect)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>ทางด้านเอาท์พุท</u>

ทำหน้าที่ในการแปลงสัญญาณโวลเตจที่เป็นเอาท์พุทของตัวควบคุม ซึ่งอยู่ในช่วงสัญญาณโวลเตจ 1 ถึง 5 โวลต์ ให้เป็นสัญญาณโวลเตจ -5 ถึง +5 โวลต์แทน เพื่อป้อนให้กับวงจรแปลงสัญญาณอนาลอคเป็นดิจิตอล ผ่านไปยังคอมพิวเตอร์ต่อไป โดยจะมีลักษณะวงจรการทำงานดังรูปที่ 10



<u>รูปที่ 10</u>

จากรูปวงจร ไอซี 1 ต่ออยู่ในลักษณะของดิฟเฟอเรนเชียลแอมป์ เพื่อทำหน้าที่ปรับระดับสัญญาณ โดยจะมีเอาท์พุทโวลเตจ ดังสมการ

$$V2 = (V1-3) * 2.5 \text{ โวลต์}$$

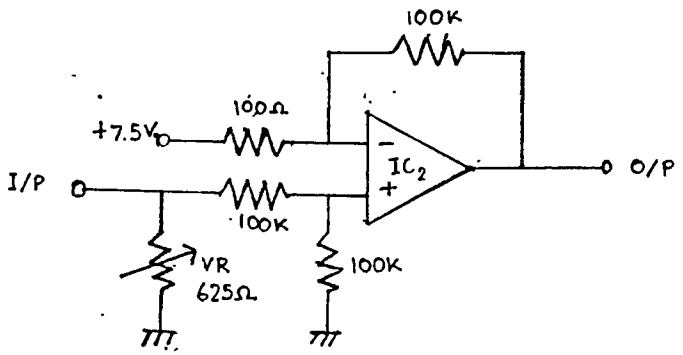
และจะได้ค่า V2 อยู่ในช่วง -5 ถึง +5 โวลต์ตามต้องการ

* ในกรณีที่ตัวควบคุมภายนอกที่ใช้ มีคุณลักษณะทางอินพุท-เอาท์พุทที่รับสัญญาณกระแสในช่วง 4-20 มิลลิแอมป์แทน เราจะเปลี่ยนวงจรที่ใช้เชื่อมต่อ จากวงจรปรับระดับสัญญาณโวลเตจดังที่กล่าวมาแล้วข้างบน มาเป็นวงจรแปลงกระแสเป็นโวลเตจ และ วงจรแปลงโวลเตจเป็นกระแส แทน ซึ่งจะกล่าวต่อไปข้างล่างนี้

2.2 <u>วงจรแปลงกระแสเป็นโวลเตจ</u>

ทำหน้าที่แปลงสัญญาณกระแสจากตัวควบคุม ซึ่งอยู่ในช่วง 4-20 มิลลิแอมป์ ให้เป็นสัญญาณโวลเตจในช่วง -5 ถึง +5 โวลต์ ก่อนที่จะส่งต่อไปยังวงจรแปลงสัญญาณอนาลอคเป็นดิจิตอล ซึ่งวงจรมีลักษณะดังรูปที่ 11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

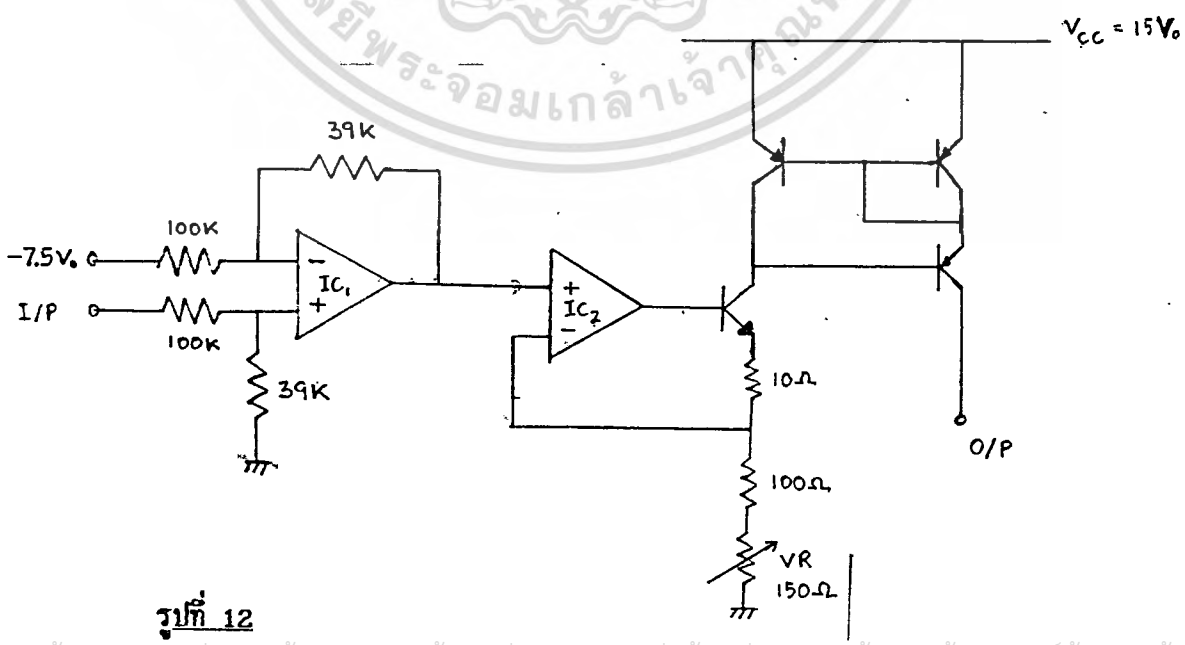


รูปที่ 11

เมื่อมีสัญญาณกระแสอินพุท ซึ่งอยู่ในช่วง 4-20 มิลลิแอมป์ ผ่านตัวต้านทาน 625 โอห์ม จะทำให้เกิดโวลเตจตกคร่อมตัวมัน ซึ่งจะอยู่ในช่วง 2.5-12.5 โวลต์ จากนั้นจึงทำการลดระดับสัญญาณลง ด้วยการนำสัญญาณระดับดีซี 7.5 โวลต์ ป้อนเข้าที่ขานอนอินเวอร์ทติ้งของไอซี 2 ทำให้ได้สัญญาณในช่วง -5 ถึง +5 โวลต์ ส่งต่อไปยังวงจรแปลงสัญญาณอนาลอคเป็นดิจิตอลต่อไป

2.3 วงจรแปลงโวลเตจเป็นกระแส

ใช้ในการแปลงสัญญาณโวลเตจเป็นกระแสที่ได้จากวงจรแปลงสัญญาณดิจิตอลเป็นอนาลอค ซึ่งมีระดับสัญญาณระหว่าง -5 โวลต์ ถึง +5 โวลต์ ให้เป็นสัญญาณกระแสขนาด 4-20 มิลลิแอมป์ ป้อนให้กับตัวควบคุม ดังวงจรในรูปที่ 12



รูปที่ 12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากวงจร ไอซี 1 ซึ่งต่อในลักษณะเป็นดิฟเฟอเรนเชียลแอมป์(Differential amplifier) มีหน้าที่ปรับระดับสัญญาณอินพุทที่รับมาจากวงจรแปลงสัญญาณดิจิตอลเป็นอนาลอค ตามสมการ

$$V2 = (V1 + 7.5)/2.5 \quad \text{โวลต์}$$

และจะได้ V2 อยู่ในช่วง 1-5 โวลต์ จากนั้นก็จะผ่านวงจรไอซี 2 และ Q1-Q4 ซึ่งต่ออยู่ในลักษณะของวงจรสะท้อนกระแส (current mirror) ซึ่งจะทำให้ได้กระแสเอาท์พุทที่ขาคอลเลคเตอร์ของ Q4 ตามสมการข้างล่าง

$$I = V2/(R1 + R2)$$

เมื่อ $R1 = 100$

$R2 = 150$ โอหม์ (ปรับค่า)

2.4 <u>วงจรแปลงสัญญาณดิจิตอลเป็นอนาลอค และ อนาลอคเป็นดิจิตอล</u>

ใช้ในการเชื่อมต่อระหว่าง ไมโครคอมพิวเตอร์ และ ตัวควบคุมภายนอกได้ โดยใช้การ์ดที่มีอยู่แล้วคือการ์ด PCL 714 ซึ่งมีคุณลักษณะดังนี้

- มี 16 อนาลอคอินพุท ที่แตกต่างกัน ด้วยค่ารีโซลูชั่น เท่ากับ 14
- มี 2 อนาลอคเอาท์พุท ด้วยค่ารีโซลูชั่น เท่ากับ 14
- มีการประมาณทางฮารดแวร์สำหรับเวลาในการแปลงข้อมูลที่ดีกว่า และเร็วกว่า
- เวลาในการแปลงสัญญาณ อนาลอค/ดิจิตอล มีค่าน้อยกว่า 40 ไมโครวินาที
- มี 16 ดิจิตอลเอาท์พุท
- มี 16 ดิจิตอลอินพุท
- ทุกๆช่องของดิจิตอลอินพุท และ เอาท์พุท สามารถใช้ได้กับ ทีทีแอล(TTL compatible)
- มีโปรแกรมตั้งช่วงเวลา ซึ่งสามารถโปรแกรมได้

สำหรับข้อกำหนดในการทำงานของ PCL 714 มีดังนี้

*** อนาลอค / ดิจิตอล ***

ช่วงกว้างอินพุท (Input range) : -5 ถึง +5 โวลต์

ความแม่นยำ (Accuracy) : 0.15% max. ที่ 25 องศาเซลเซียส

ความต้านทานอินพุท (Input impedance) : 10 เมกกะโอห์ม

เวลาในการแปลง (Converstion time) : น้อยกว่า 40 ไมโครวินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

 # ดิจิตอล / อนาลอค #

ช่วงกว้างเอาท์พุท (Output range) : -5 ถึง +5 โวลต์

ความแม่นยำ (Accuracy) : o.i% max. ที่ 25 องศาเซลเซียส

เวลาในการเข้าถึง (Settling time) : น้อยกว่า 30 ไมโครวินาที

 # ดิจิตอล-อินพุท #

อินพุทระดับต่ำ (Low level) : min. -0.5 โวลต์,max. -0.8 โวลต์

อินพุทระดับสูง (High level) : min. 0.5 โวลต์ ,max. 5.0 โวลต์

 # ดิจิตอล-เอาท์พุท #

เอาท์พุทระดับต่ำ (Low level) : max. 0.5 โวลต์ ที่ 24 มิลลิแอมป์

max. 0.4 โวลต์ ที่ 12 มิลลิแอมป์

เอาท์พุทระดับสูง (High level) : min. 2.0 โวลต์ ที่ 15 มิลลิแอมป์

min. 2.4 โวลต์ ที่ 3 มิลลิแอมป์

2.5 ตัวควบคุม (Controller)

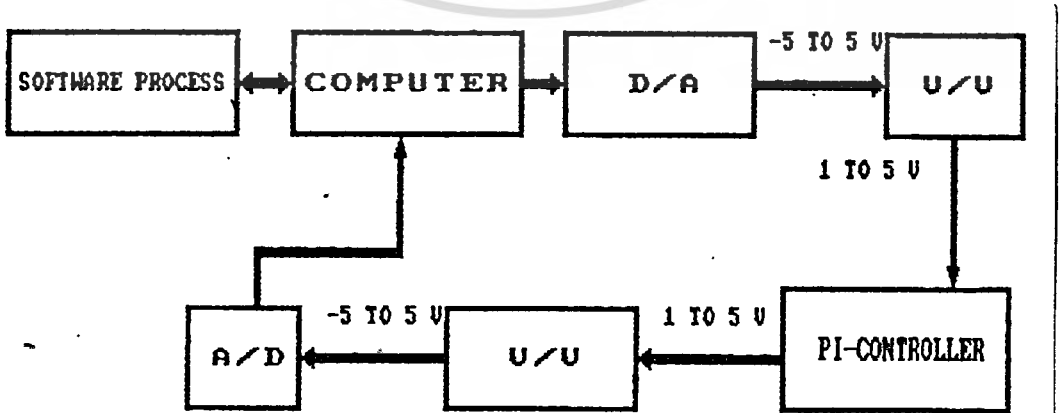
ในปริญญานิพนธ์นี้ มีการใช้ตัวควบคุม 2 แบบ คือ

1. ตัวควบคุมภายนอก (Outside controller)
2. ตัวควบคุมภายใน (Inside controller

โดยโปรแกรมจะให้ผู้ใช้เลือกว่าจะใช้แบบใด ลักษณะของแต่ละแบบจะเป็นดังนี้

1. ตัวควบคุมภายนอก (Outside controller)

ซึ่งมีลักษณะดังรูปที่ 13



รูปที่ 13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ซอฟแวร์ (Software)

โปรแกรมการจำลองกระบวนการที่เขียนขึ้น เขียนโดยใช้ภาษาเบสิค ใช้Quick BASIC Version 4.00 ของบริษัทไมโครซอฟ จำกัด เป็นคอมพายเลอร์ (Compiler)ลักษณะของโปรแกรม จะเป็นดังนี้

1. ส่วนเริ่มต้นของโปรแกรมจะมีการแสดงเมนูหลัก (Main menu) มาให้ผู้ใช้เลือก ดังแสดงในโฟลว์ชาร์ตที่ 1 ซึ่งตัวเลือกจะมีทั้งหมด 7 ตัว และจะทำงานเมื่อผู้ใช้กดคีย์ที่เป็นอักษรตัวแรกของชื่อตัวเลือกนั้น โปรแกรมจะทำงานวนลูปคอยตรวจเช็คว่ามีตัวเลือกใดถูกกด ถ้ามีก็จะไปทำงานตามตัวเลือกนั้น แต่ถ้าไม่มีก็จะวนอยู่จนกว่าจะมีคีย์ใดถูกกด

ตัวเลือกทั้ง 7 คือ

-Load process เป็นการโหลดไฟล์เดิมมาใช้งาน มีรายละเอียดแสดงในโฟล์วชาร์ตที่ 2

-New process เป็นการสร้างไฟล์ใหม่ขึ้นมา ซึ่งมีรายละเอียดดังแสดงในโฟล์วชาร์ตที 3

-Run process เป็นการจำลองผลลัพธ์จากการจำลองระบบ ซึ่งมีรายละเอียดดังแสดงในโฟล์วชาร์ตที 4

-File name เป็นการไดเรคตอรี่ไฟล์ที่มีอยู่ ดังแสดงในโฟล์วชาร์ตที 5

-Edit data จะเป็นการแก้ไขข้อมูลเดิมทีทำการโหลดมา

-Save data จะเป็นการเก็บข้อมูลกลับไปไว้ในดิสเก็ตต์

-Quit จะเป็นการกลับไปสู่ DOS

2.สำหรับโฟลว์ชาร์ตแสดงขั้นตอนการทำงานของแต่ละตัวเลือก จะแสดงในโฟลว์ชาร์ตที่ 2-5 ซึ่งมีรายละเอียดดังนี้ คือ

1.โฟล์วชาร์ตที 2 แสดงส่วนการโหลดข้อมูลของกระบวนการเดิม ทีเก็บเป็นไฟล์ไว้ในดิสเก็ตต์ เมื่อผู้ใช้กดคีย์ L ในเมนูหลัก โปรแกรมจะเรียก sub 1100 และจะถามว่าต้องการโหลดไฟล์ใด โปรแกรมจะทำการโหลด และ แสดงข้อมูลออกทางจอ ซึ่งในขั้นตอนนี้ผู้ใช้จะมีทางเลือก 3 ทางคือ แก้ไขข้อมูล (Edit) ,โหลดข้อมูลมาดูต่อ (Continue) ,และเมนูหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. โฟล์วชาร์ตที 3 แสดงการสร้างกระบวนการใหม่ (New process) เมื่อผู้ใช้กดคีย์ N ในเมนูหลัก โปรแกรมจะลบจอภาพ และเปลี่ยนไปอยู่ในโหมดกราฟิก (SCREEN 3) จากนั้นจะแสดงเคอร์เซอร์ขึ้นมา เคอร์เซอร์นี้จะเป็นตัวกำหนดตำแหน่งรูปที่ผู้ใช้จะวาดขึ้น โดยจะเปลี่ยนไปมาได้ด้วยการกดคีย์ลูกศร เมื่อผู้ใช้เลื่อนมายังตำแหน่งที่ต้องการ ให้กดคีย์ S โปรแกรมจะแสดงเมนูมาให้เลือก ดังแสดงในโฟล์วชาร์ต sub A100 ถ้าผู้ใช้ต้องการรูปใด ก็ให้กดคีย์ตัวนั้น รูปก็จะแสดงออกมาทีตำแหน่งของเคอร์เซอร์ จากนั้นเมนูจะถูกลบ และ เคอร์เซอร์ก็สามารถเลื่อนไปทีอื่นๆ เพื่อสร้างรูปอื่นๆต่อไปได้

ถ้าผู้ใช้เลือกรูป แทงค์ หรือ วาล์ว ซึ่งจำเป็นต้องมีการใส่ข้อมูล โปรแกรมจะแสดงส่วนใส่ข้อมูล ขึ้นมาให้เติมลงไป

เมื่อผู้ใช้ต้องการเก็บรูปภาพและข้อมูลก็ให้กดคีย์ S และจากเมนูให้กดคีย์ E (Exit) โปรแกรมจะขึ้นตัวเลือกมา ให้กดคีย์ S(Savepic) โปรแกรมก็จะเก็บรูปและข้อมูลทั้งหมดที่ผู้ใช้สร้างขึ้นมาใหม่

3. โฟล์วชาร์ตที 4 แสดงโปรแกรมส่วนการคำนวณ ,รับส่ง อินพุท-เอาท์พุท , แสดงผลลัพธ์ทางจอ โดยจากเมนูหลัก ให้ผู้ใช้กดคีย์ R(Run process) โปรแกรมส่วนนี้ก็จะทำงาน โดยจะขึ้นตัวเลือกให้ผู้ใช้เลือกว่าต้องการตัวควบคุมแบบใด เป็นตัวควบคุมภายนอก หรือ ภายใน สำหรับผลลัพธ์ที่แสดงทางจอจะมีหลายลักษณะ คือ

- แสดงโดยอาศัยรูปทีผู้ใช้ทำขึ้น และเก็บไว้ในดิสเก็ตต์ ซึ่งจะแสดงทั้งระดับน้ำและรูปบาร์กราฟ (Bargraph) ทั้งQin และ Qout ถ้าผู้ใช้ต้องการแสดงรูปแบบนี้ให้กดคีย์ B ตามด้วยหมายเลขของ หน้า(page) และคีย์ S

- แสดงในรูปกราฟทางเวลา (Time graph) ซึ่งจะแสดงผลตอบสนองทางเวลาของแต่ละสมการ (สมการอนุพันธ์อันดับหนึ่ง 1 สมการจะแสดงกราฟนี้ 1 รูป) โดยจะมีการแสดงทั้ง Qin และ Qout ถ้าผู้ใช้ต้องการแสดงแบบนี้ให้กดคีย์ T ตามด้วยหมายเลขของแต่ละสมการ และตามด้วยคีย์ S ด้วย

- กรณีผู้ใช้เลือกตัวควบคุมภายใน ถ้าผู้ใช้กดคีย์ C (Controller) โปรแกรมจะแสดงรูปของตัวควบคุมขึ้นมา โดยเราสามารถปรับตั้งค่า Kp , Ki, Kd ได้ตามต้องการ โดยแต่ละค่าจะอยู่ในช่วง 0 ถึง 100

- ถ้าผู้ใช้กดคีย์ S(Setpoint) จะเป็นการตั้งค่าเป้าหมายให้กับตัวควบคุมภายใน และค่าที่จะมีแสดงออกบนจอด้วย

เอกสารนี้จัดทำขึ้นสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อต้องการกลับไปที่เมนูหลัก ก็กดคีย์ M

4. โฟล์วชาร์ตที่ 5 แสดงการไดเรคตอรี่ไฟล์ในแผ่นดิสเก็ตต์มาดู ว่ามีไฟล์ใดบ้าง ผู้ใช้สามารถลบไฟล์ทิ้งไป หรือ เปลี่ยนชื่อไฟล์ได้

ตัวอย่างแสดงขั้นตอนของผู้ใช้ในการจำลองระบบระดับของเหลวโดยใช้โปรแกรมนี้

1. โหลดโปรแกรม"Process Control Simulation"เข้าเครื่องคอมพิวเตอร์ เมื่อเรียบร้อย โปรแกรมก็จะแสดง เมนูหลัก มาให้ผู้ใช้เลือก

2. เลือก New Process โดยให้กดคีย์ N ซึ่งจะเป็นการสร้างรูปและกำหนดข้อมูลใหม่ โปรแกรม New process จะแสดงได้ในโฟลว์ชาร์ตที่ 3ซึ่งให้ผู้ใช้กดคีย์ N อีกที(New pic) โปรแกรมก็จะแสดงจอภาพที่มีเคอร์เซอร์ (cursur) เลื่อนไปมาได้โดยใช้คีย์ลูกศร ขึ้น, ลง, ซ้าย , ขวา

3. เลื่อนเคอร์เซอร์ ไปตำแหน่งที่ต้องการจากนั้นกดคีย์ S(stop) จะมีเมนูขึ้นมาให้เลือก สมมติว่าผู้ใช้กดคีย์ T (tank) ก็จะแสดงรูปแท็งค์ขึ้นมา ผู้ใช้สามารถทำตามขั้นตอนนี้ไปได้เรื่อยๆ จนพอใจ

4. กรณี แทงค์น้ำ และ วาล์ว ซึ่งเป็นอุปกรณ์ที่เราจะจำลองพฤติกรรม ด้วยระบบอันดับที่หนึ่งนั้น หลังจากแสดงภาพของ แทงค์น้ำ หรือ วาล์ว แล้ว ก็จะมีส่วนป้อนข้อมูลของตัวมันขึ้นมา ลักษณะจะเป็นดังรูปที่ ให้ทำการป้อนข้อมูลให้ครบ ข้อมูลนี้จะใช้ในการโปรแกรมส่วนคำนวน

5. เมื่อผู้ใช้ต้องการเก็บข้อมูลและเก็บรูปภาพที่สร้างขึ้นมาให้กดคีย์ S (save pic) โปรแกรมจะทำการเปิดไฟล์ ให้ผู้ใช้กำหนดชื่อไฟล์ โปรแกรมจะเก็บรูปภาพและข้อมูลลงไว้ในดิสเก็ตต์ หลังจากนี้ผู้ใช้สามารถสร้างรูปต่อได้ หรืออาจกลับไปที่เมนูหลัก เพื่อทำการทดสอบระบบที่สร้างขึ้นได้

6. เมื่ออยู่ในเมนูหลัก และต้องการทดสอบระบบ (run process) ที่สร้างขึ้น ให้กดคีย์ R โปรแกรมก็จะถามว่า ต้องการใช้ตัวควบคุมภายใน หรือ ภายนอก ซึ่งผู้ใช้ต้องเลือก สำหรับลักษณะการทำงานของโปรแกรม (Run process) แสดงในโฟล์วชาร์ตที่ 4 กรณีผู้ใช้เลือกตัวควบคุมภายนอก ก็จะมีความจำเป็นที่ต้องมีการรับและส่งข้อมูลกับอุปกรณ์ภายนอก แต่ถ้าเป็นตัวควบคุมภายใน โปรแกรมก็จะรับและส่งค่าภายในโปรแกรมเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. เมื่ออยู่ในส่วน Run process ถ้าผู้ใช้ต้องการกลับไปที่เมนูหลักเพื่อทำงานอื่นๆ ให้กดคีย์ E (exit) การทำงานก็จะหยุด โปรแกรมจะแสดงเมนูหลักมาอีกครั้ง

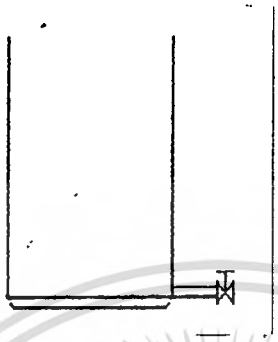
8. ถ้าผู้ใช้ต้องการ Load program เดิมที่มีการเก็บไว้ในดิสเก็ตต์มาใช้ ให้กดคีย์ L โปรแกรมก็จะถามชื่อไฟล์ ที่ต้องการ พร้อมทั้งทำการดึงไฟล์นั้นเข้ามา ดังแสดงในโฟล์วชาร์ตที่ 2

9. ถ้าผู้ใช้ต้องการกลับไปที่ Dos promt ให้กดคีย์ Q (quit) ในเมนูหลัก โปรแกรม Process Control Simulation ก็จะยุติลง

สำหรับโปรแกรมที่เขียนขึ้นนี้มีข้อจำกัด คือ จะทำงานได้เฉพาะเครื่องที่มี เฮอร์คิวลิสการ์ด เท่านั้น เพราะในภาษาเบสิค เมื่อจะใช้คำสั่งทางกราฟฟิค ต้องใช้คำสั่ง SCREEN 3 เสมอ และกำหนดขนาดของจอภาพ ขนาดของรูปภาพก็จะแตกต่างไปด้วยสำหรับโปรแกรมทั้งหมดที่เขียนขึ้นทั้งหมดได้มีแสดงไว้ในภาคผนวก ก. ตอนท้ายเล่มแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับกรณีที่ผู้ใช้ . เลือกกดคีย์ N ในเมนูหลัก เพื่อเป็นการสร้างกระบวนการใหม่ โปรแกรมก็จะทำงานตามโฟล์วชาร์ตที่ 3 ซึ่งผู้ใช้จะต้องเลือกตำแหน่ง และ อุปกรณ์ต่างๆที่ต้องการ ซึ่งอุปกรณ์ต่างๆที่มีการจำลองและสามารถเรียกใช้ได้มีดังนี้ คือ

1. 

แท็งค์ ซึ่งจะต้องมีท่อและวาล์วทางด้านเอาท์พุทด้วยเสมอ

2. ปั๊มน้ำ

3. ท่อ

4.

ข้อต่อ ซึ่งจะมีอยู่ด้วยกัน 4 ลักษณะด้วยกัน ให้เลือกใช้

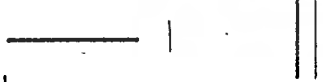
5. วาล์ว ซึ่งใช้เป็นวาล์วควบคุม

6. ตัววัดระดับ (Level transmitter)

ซึ่งแทนได้ด้วยสัญญลักษณ์ L.T

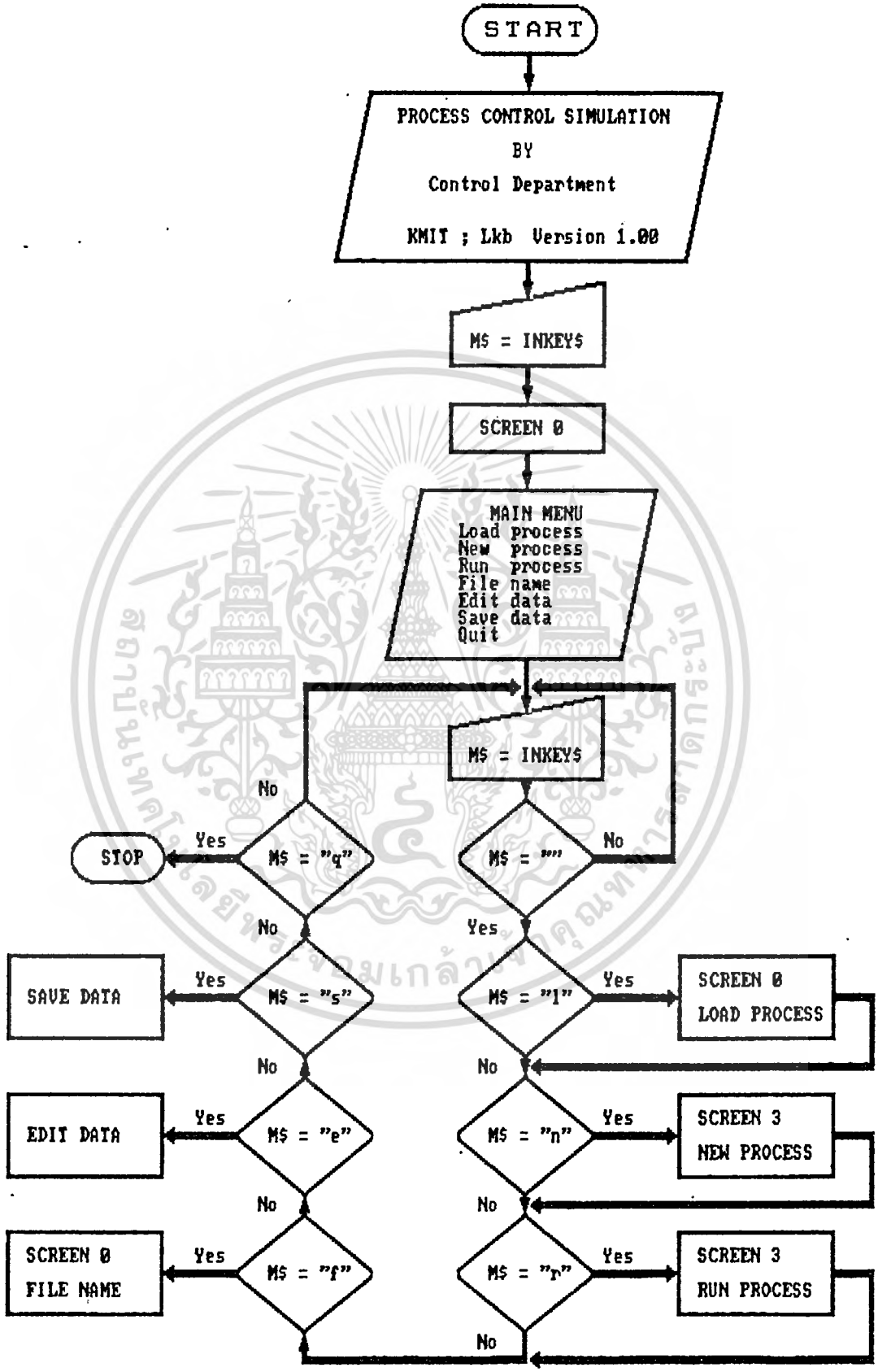
7. ตัวควบคุมระดับ(Level controller)

ซึ่งแทนได้ด้วยสัญญลักษณ์ LC

8. 

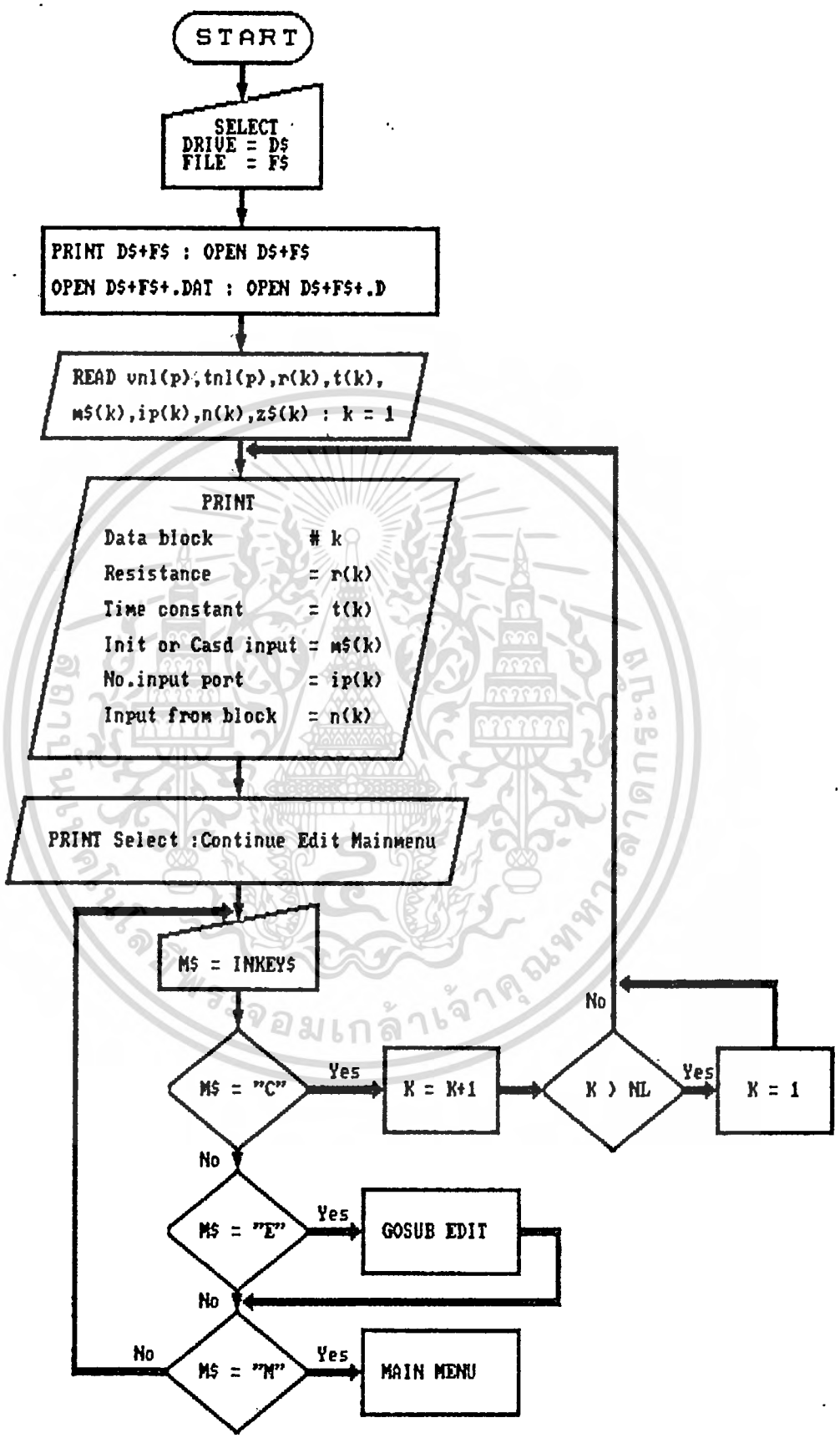
เส้น ซึ่งจะแทนสายส่งสัญญาณ (Transmission line) ในการส่งสัญญาณที่วัดได้ไปยังตัวควบคุม หรือ สัญญาณจากตัวควบคุมไปยังวาล์ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



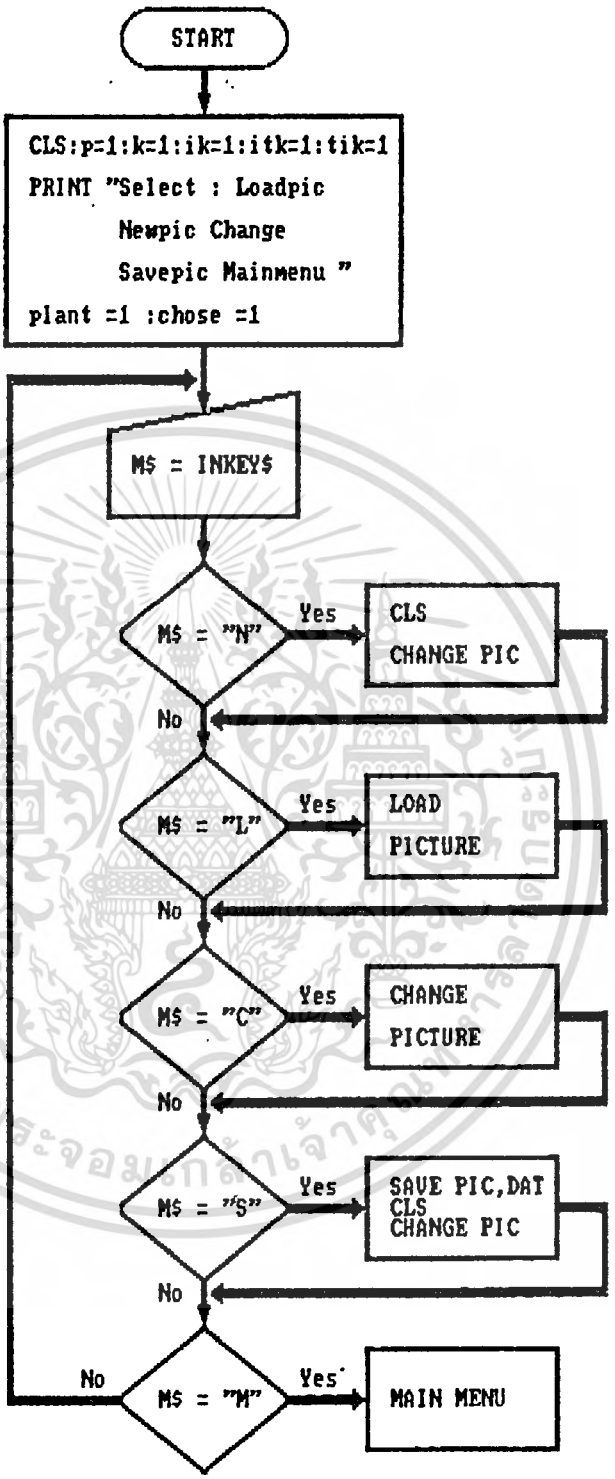
FLOW CHART 1. Main menu

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



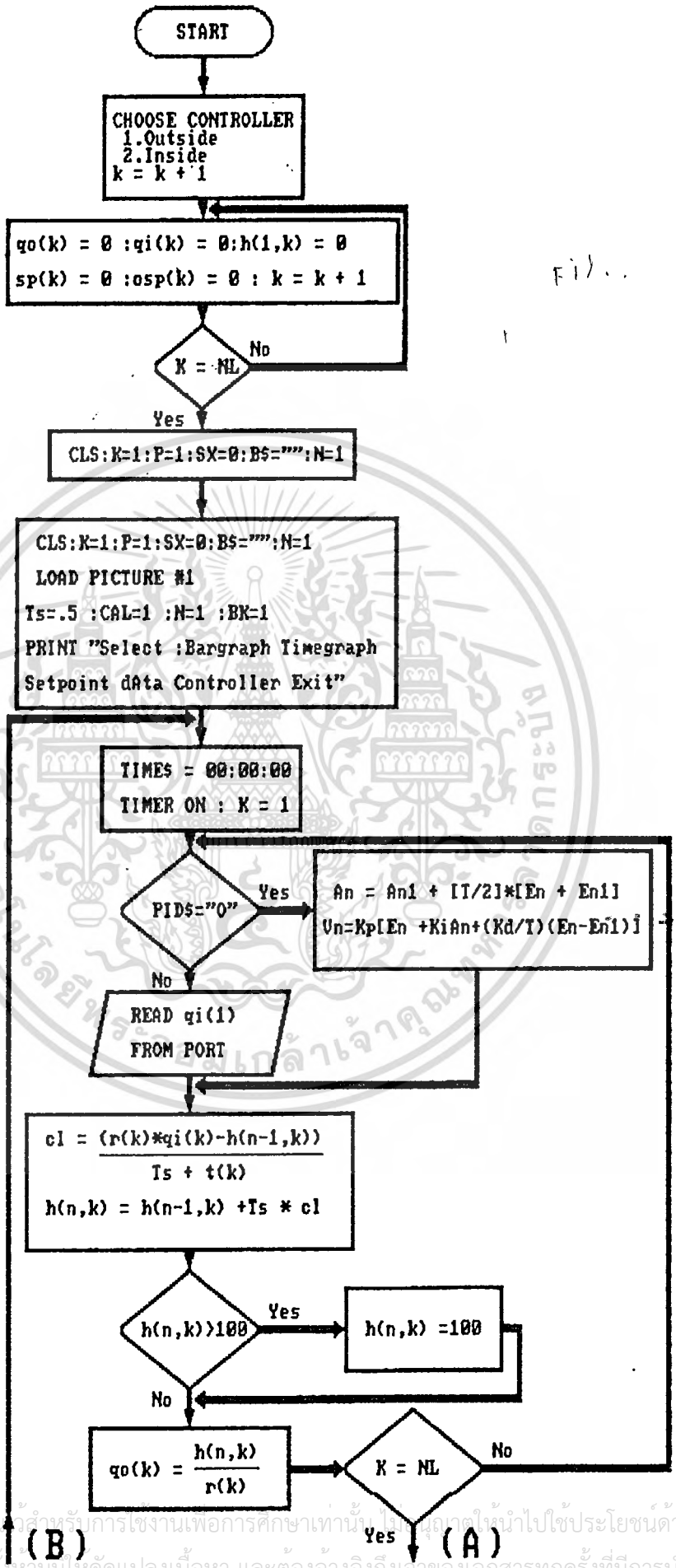
FLOW CHART 2. Load data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

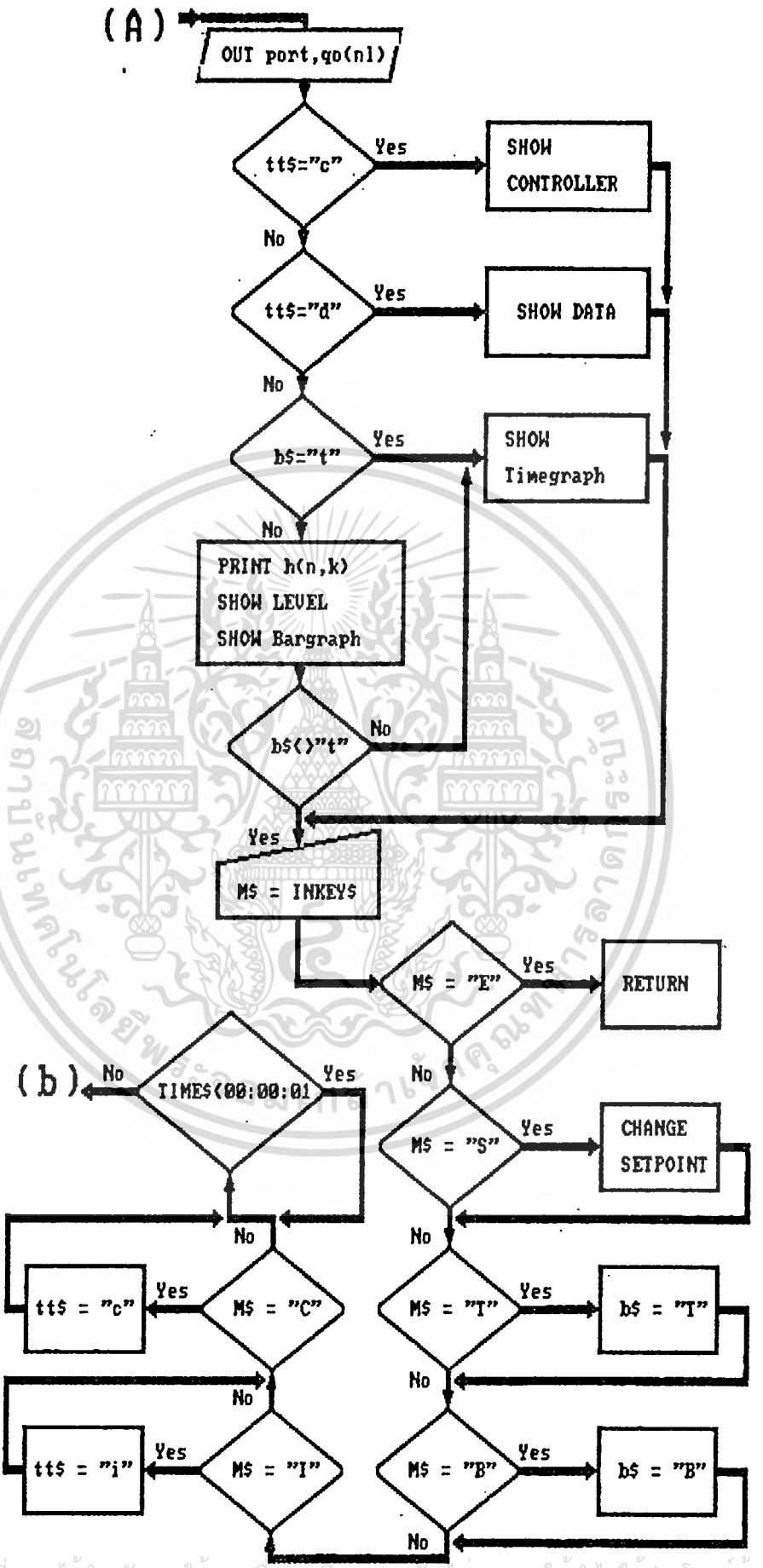


FLOW CHART 3. New process

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

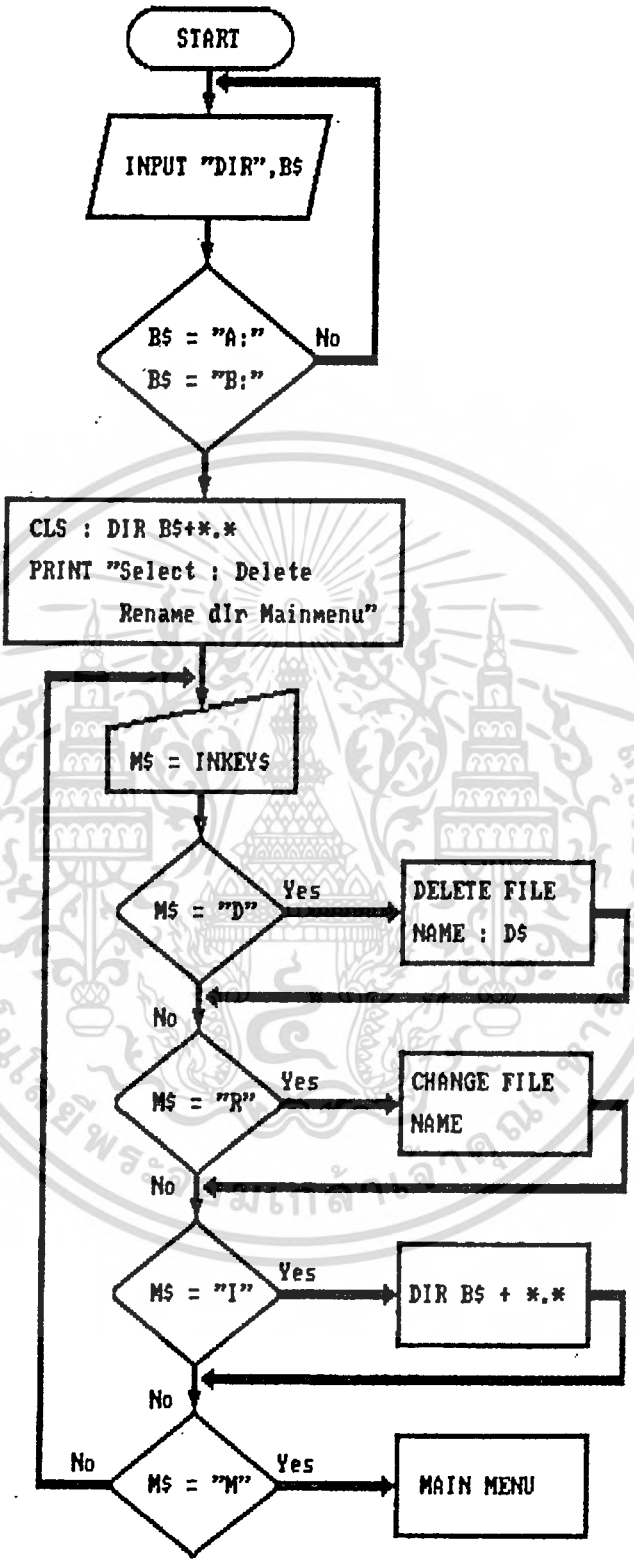


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



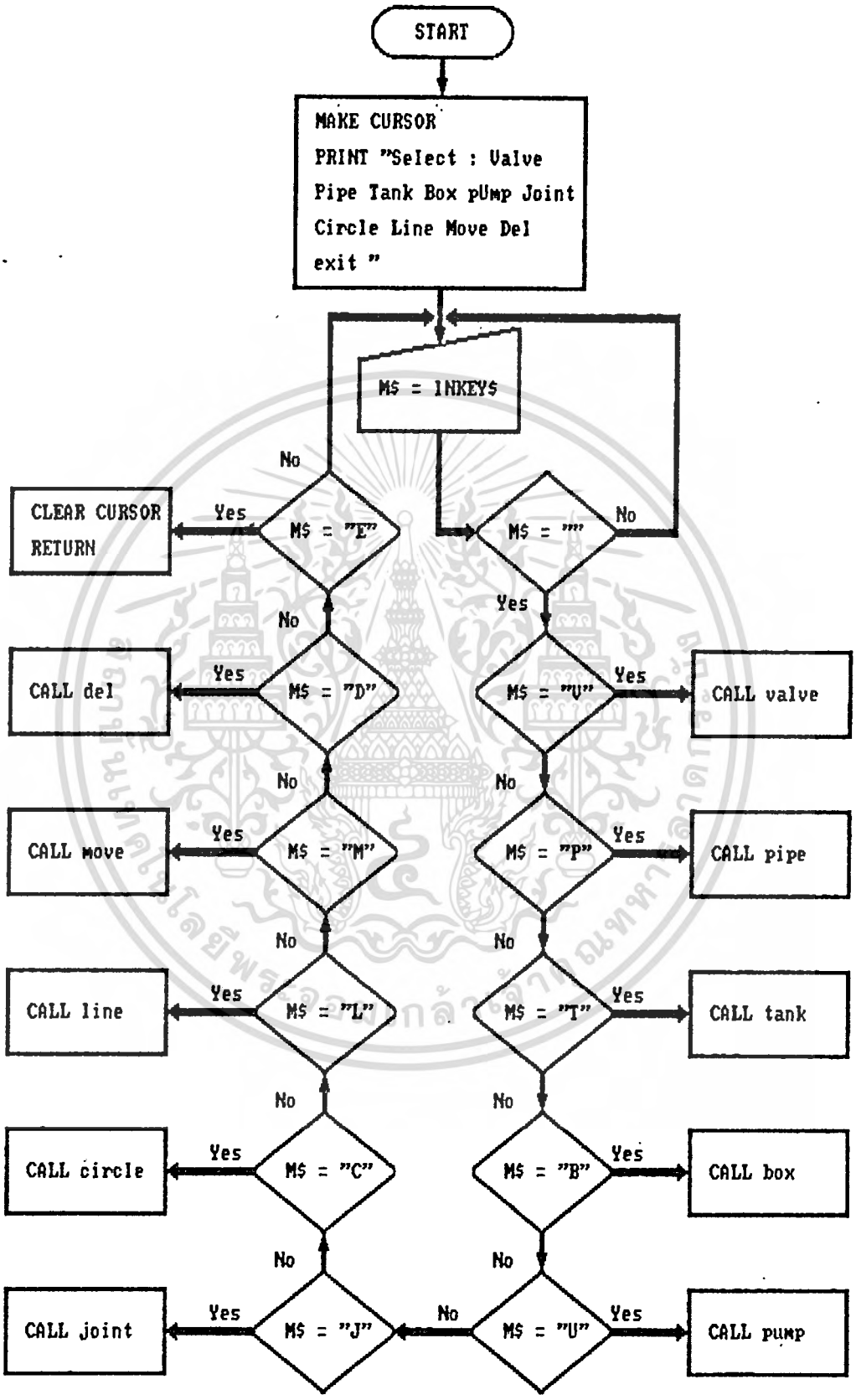
FLOW CHART 4. Run process

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



FLOW CHART 5. Dir file

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



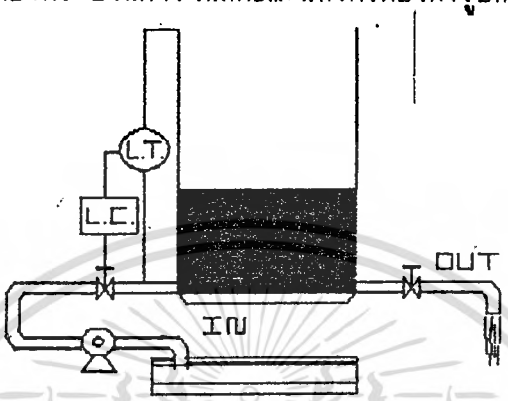
SUB A1000 : Change picture

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

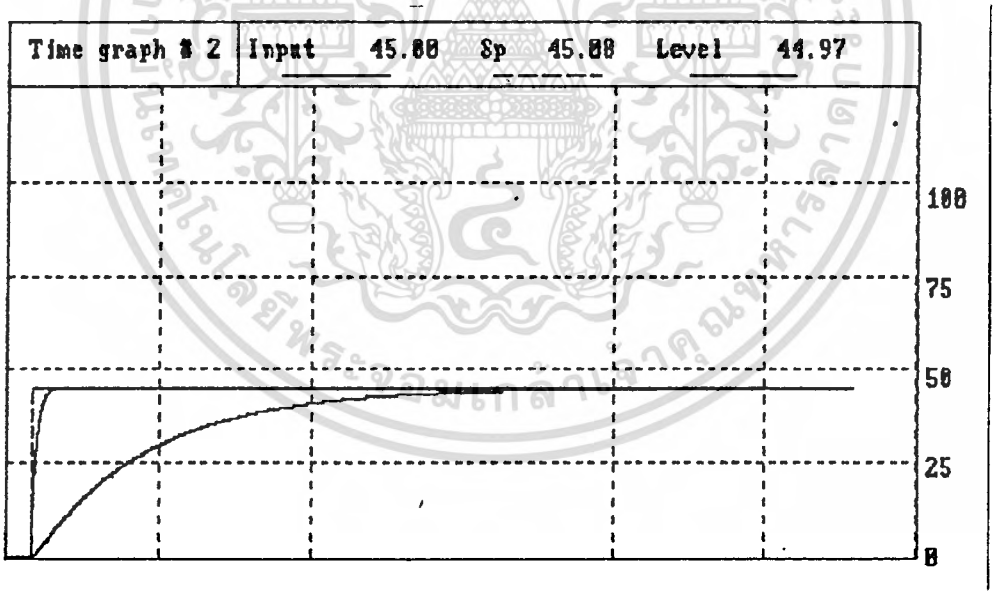
ผลการทดลอง

เมื่อทำการจำลองกระบวนการในลักษณะแทงค์เดี่ยวดังรูปที่ 15



รูปที่ 15

ซึ่งสามารถแทนได้ด้วย ระบบอันดับหนึ่งแบบล้าหลัง ที่ยังไม่มีการป้อนกลับ และ มีอินพุทเป็นสเตป ขนาด 45 % เมื่อพิมพ์ผลตอบสนองทางเวลาที่ปรากฏบนจอมอนิเตอร์ออกมาพิจารณา จะได้ผลตามรูปที่ 16

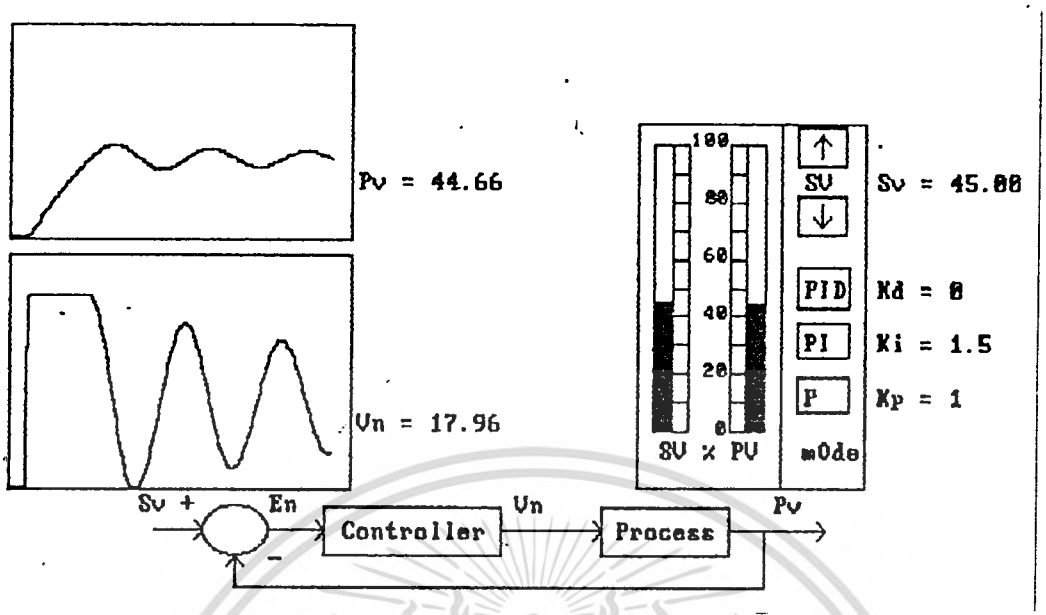


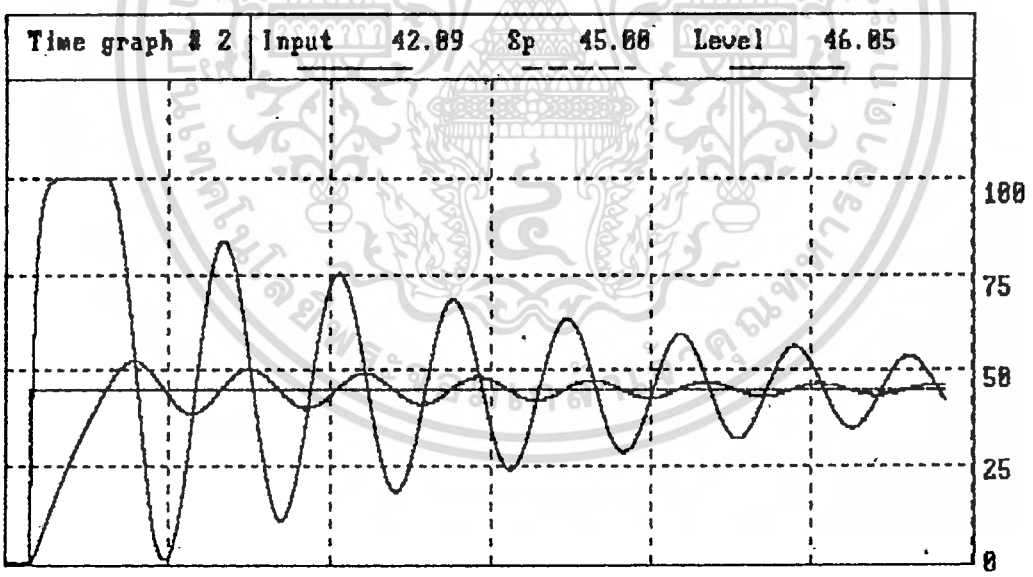
รูปที่ 16

ซึ่งผลที่ได้มีลักษณะเป็นเอ็กโปเนนเชียลที่จะมีค่าค่อยๆเพิ่มขึ้น และจะเข้าสู่ค่าเป้าหมายเมื่อเวลาผ่านไปประมาณ 4 เท่าของค่าคงที่ทางเวลาของกระบวนการ เมื่อหาค่าความชันที่เวลาเริ่มต้น (t=0) มีค่าเท่ากับ 1/(ค่าคงที่ทางเวลา)จริง ซึ่งสอดคล้องกับผลที่ได้จากทฤษฎีดังรูปที่ 4

เมื่อทดลองต่อกระบวนการที่ทำการจำลองกับตัวควบคุมภายใน ในลักษณะลูปปิดแล้ว จะได้ผลตอบสนองทางเวลา ดังรูปที่ 17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

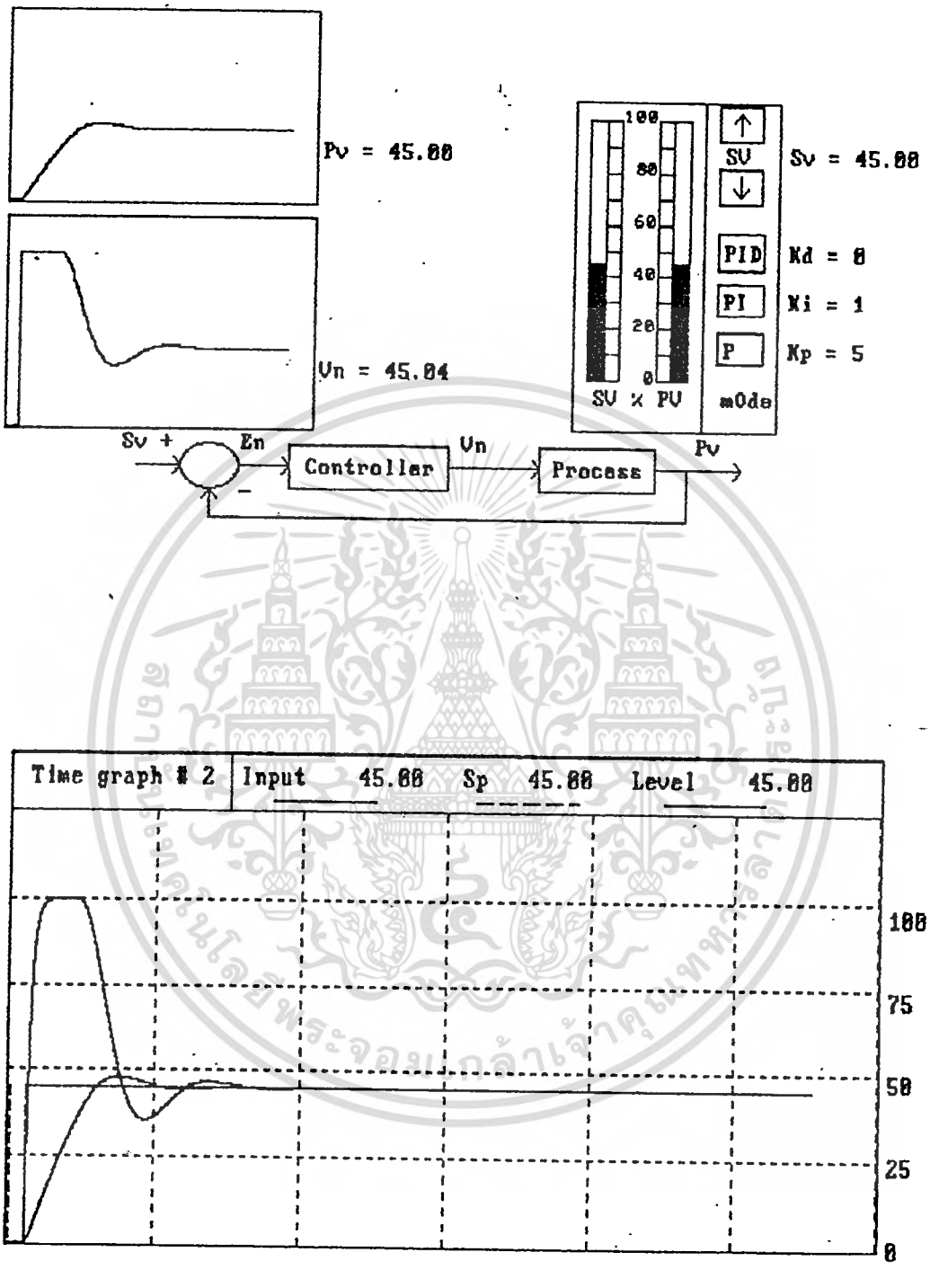




รูปที่ 17

ซึ่งเป็นผลที่ได้จากการทดลองปรับค่าให้ Kp = 1 และ Ki = 1.5 ซึ่งผลตอบสนองจะมีค่าโอเวอร์ชูทที่สูง และ จะมีการแกว่งมากก่อนเข้าสู่ค่าเป้าหมาย แต่เมื่อปรับค่าใหม่โดยที่ให้ Kp = 5 และ Ki = 1 จะได้ผลตอบสนองใหม่ดังรูปที่ 18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 18

ซึ่งผลตอบสนองจะมีโอเวอร์ชูทลดลง และ การเข้าสู่ค่าเป้าหมายที่เร็วขึ้น ดังนั้นผลการทดลองที่ได้แสดงว่าสอดคล้องกับทฤษฎี

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุปและวิจารณ์

สรุปการทำปริญญานิพนธ์

สำหรับการวิจัยนี้มีจุดประสงค์หลัก คือการจำลองกระบวนการควบคุมระดับของเหลว โดยใช้ไมโครคอมพิวเตอร์ ดังนั้นงานหลักๆจึงมี 2 ด้าน คือ

1. การใช้สมการทางคณิตศาสตร์จำลองระบบจริง

2. เขียนโปรแกรมจำลองระบบ โดยใช้รูปแบบทางคณิตศาสตร์จากข้อ 1 เนื่องจากการจำลอง ใช้หลักการพิจารณาด้วยสมการอนุพันธ์อันดับหนึ่งมาต่อกันในลักษณะนอนอินเตอร์แอกติ้ง ดังนั้นโปรแกรมที่ทำการจำลองจึงเป็นการประมาณระบบจริงๆเท่านั้น เมื่อเขียนโปรแกรมเป็นที่เรียบร้อย แล้วทำการทดลองโปรแกรมสามารถทำงานได้ ส่วนข้อผิดพลาดที่เกิดขึ้น เนื่องมาจาก

1. วงจรแปลงสัญญาณดิจิตอลเป็นอนาลอค และ อนาลอคเป็นดิจิตอล มีค่าผิดพลาดในการแปลงค่าเล็กน้อยเป็นจุดทศนิยม

2. วงจรปรับระดับโวลเตจ มีค่าผิดพลาดเล็กน้อย

ผลของข้อผิดพลาดข้างต้น ทำให้ค่า ตัวแปรกระบวนการ และ ตัวแปรที่ใช้ในการควบคุมระหว่างคอมพิวเตอร์ และ ตัวควบคุม มีค่าแตกต่างกัน

จากโปรแกรมสามารถเลือกตัวควบคุมได้ 2 แบบนั้น ทั้งนี้เพราะจุดประสงค์หลักเพื่อนำไปใช้กับตัวควบคุมภายนอก แต่ในบางครั้งที่ผู้ใช้ไม่มีตัวควบคุมภายนอก ก็สามารถที่จะใช้ตัวควบคุมที่จำลองขึ้นได้

ประโยชน์ที่คาดว่าจะได้จากปริญญานิพนธ์นี้ คือ สามารถนำไปใช้ในการศึกษาเกี่ยวกับการควบคุมระดับของเหลว , ศึกษาวิธีการปรับค่าตัวควบคุม ซึ่งจะทำให้เข้าใจได้ง่ายขึ้น สำหรับผู้ที่เริ่มศึกษาทางด้านระบบควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แนวทางในการปรับปรุง

โดยทั่วไปแล้วการควบคุมกระบวนการมีตัวแปรที่สำคัญ 4 ตัวคือ ระดับ ,อัตราการไหล ,ความดัน และ อุณหภูมิ ในปริญญานิพนธ์นี้ ศึกษาแต่ทางระดับเท่านั้น จึงยังเป็นการจำลองกระบวนการที่ไม่สมบูรณ์ ดังนั้นผู้ที่สนใจสามารถที่จะทำการจำลองตัวแปรที่เหลือได้ เพราะหลักการจำลองสามารถแทนระบบจริงๆด้วยสมการอนุพันธ์อันดับหนึ่งได้เช่นกัน

สำหรับข้อผิดพลาดทั้งหมดในปริญญานิพนธ์ ผู้จัดทำต้องขออภัยมา ณ ที่นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
-----------------------------------------------
```

โปรแกรม

ชื่อ การจำลองระบบควบคุมกระบวนการ (Process control simulation)

โดย

กนกเวทย์	ตั้งพิมลรัตน์	281014
ชยากร	เลี้ยงรื่นรมย์	281058
ชัชพล	ประสพโชค	281059

เขียนโดยภาษา เบสิค (Quick BASIC Version 4.00)

เป็นส่วนหนึ่งของปริญญานิพนธ์ ระดับปริญญาตรี ประจำปี 2531

ภาควิชาระบบควบคุม คณะวิศวกรรมศาสตร์ ; ลาดกระบัง

```
-----------------------------------------------

DECLARE SUB bargraph (b, tnl!, x!(), y!())
DECLARE SUB Tdata (k!, r!(), t!(), ip!(), n!(), m$())
DECLARE SUB move (x!, y!, x1!, y1!, c!())
DECLARE SUB crs (x!, y!, c!())
DECLARE SUB del (x, y, x1, y1, c())
DECLARE SUB Lne (x, y, x1, y1, c())
DECLARE SUB Ccle (x1, y1)
DECLARE SUB Joint (x, y)
DECLARE SUB pump (x, y, x1, y1, c())
DECLARE SUB tank(p,x(),y(),z$(),tnl,x1,y1,tx(),ty(),r(),t(),n(),m$())
DECLARE SUB valve(p,z$(),vnl,x,y,x1,y1,c(),r(),t(),ip(),n(),m$())
DECLARE SUB box (x, y, x1, y1, c())
DECLARE SUB pipe (x, y, x1, y1, c())
DIM h(100,99),qi(99),qo(99),tx(99,99),ty(99,99),tnl(99),vnl(99)
DIM r(99), t(99), m$(99), n(99), z$(99), x(99, 99), y(99, 99)
DIM oq(99), tpg(99), ip(99), sp(99), osp(99)
cx = 1: x = 1: lx = 1: nl = 0: tnl = 0: vnl = 0: CLS :
'-------------------------<<Start>>-----------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
FOR x = 4 TO 16
LOCATE x, 25: PRINT CHR$(186)
NEXT x
FOR y = 25 TO 55
LOCATE 4, y: PRINT CHR$(178)
LOCATE 17, y: PRINT CHR$(178)
NEXT y
COLOR 15, 0
LOCATE 6, 28: PRINT "PROCESS CONTROL SIMULATION"
LOCATE 8, 39: PRINT "BY"
LOCATE 10, 29: PRINT "Chatchaphol Prasobchoke"
LOCATE 12, 31: PRINT "Control Department"
LOCATE 14, 29: PRINT "KMIT ;Lkb  Version 1.00"
LOCATE 16, 27: PRINT "<Press any key to continue>"
WHILE INKEY$ = "" : WEND:
'-----------------------<<Main menu>>-------------------------
CLS : COLOR 7, 0
GOSUB ik10
WHILE cx
mm$ = INKEY$
IF mm$ <> "" THEN
IF mm$ = "l" OR mm$ = "L" THEN GOSUB l100
IF mm$ = "n" OR mm$ = "N" THEN
SCREEN 3: GOSUB p100
END IF
IF mm$ = "r" OR mm$ = "R" THEN
IF d$ <> "" OR f$ <> "" THEN GOSUB r100
END IF
IF mm$ = "f" OR mm$ = "F" THEN
SCREEN 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
GOSUB a2000
END IF
IF mm$ = "s" OR mm$ = "S" THEN
IF d$ <> "" AND f$ <> "" THEN
s$ = "d": ik = 1: GOSUB s110
END IF
END IF
IF mm$ = "e" OR mm$ = "E" THEN
IF d$ <> "" OR f$ <> "" THEN GOSUB l2000
END IF
IF mm$ = "q" OR mm$ = "Q" THEN END
GOSUB ik10
END IF
WEND
'------------------------New process-----------------------------'
p100:
CLS
plant=1: chose = 1:p=1:k=1:ik=1:itk =1:tik=0:tnl=0:vnl=0
LOCATE 23, 2
PRINT "Select: Loadpic Newpic Change Savepic Mainmenu "
WHILE chose
m$ = INKEY$
IF m$ = "n" OR m$ = "N" THEN
CLS : GOSUB a1000
END IF
IF m$ = "l" OR m$ = "L" THEN GOSUB a4000
IF m$ = "c" OR m$ = "C" THEN GOSUB a1000
IF m$ = "s" OR m$ = "S" THEN
GOSUB s100: CLS : GOSUB a1000
END IF
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
IF m$ = "m" OR m$ = "M" THEN chose = 0
WEND
CLS
RETURN
'------------------------Change picture-------------------------
a1000:
LOCATE 22, 70: PRINT "page # "; p
DIM c(10)
x = 0: y = 0
LINE (1, 1)-(4, 4), 1, BF
GET (0, 0)-(4, 4), c
WHILE chose
CALL crs(x, y, c())
x1 = x: y1 = y
LOCATE 23, 2
PRINT"Select:Valve Pipe Tank Box pUmp Joint Circle Line Move Del Exit"
plant = 1
WHILE plant
m$ = INKEY$
IF m$ <> "" THEN
IF m$="v" OR m$="V" THEN
CALL valve(p,z$(),vn!,x,y,x1,y1,c(), r(), t(), ip(), n(), m$())
END IF
IF m$ = "p" OR m$ = "P" THEN CALL pipe(x, y, x1, y1, c())
IF m$="t"OR m$="T " THEN
CALL tank(p,x(),y(),z$(),tn!,x1,y1,tx(),ty(),r(),t(),ip(),n(),m$())
END IF
IF m$ = "b" OR m$ = "B" THEN CALL box(x, y, x1, y1, c())
IF m$ = "u" OR m$ = "U" THEN CALL pump(x, y, x1, y1, c())
IF m$ = "j" OR m$ = "J" THEN CALL Joint(x, y)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
IF m$ = "c" OR m$ = "C" THEN CALL Ccle(x1, y1)
IF m$ = "l" OR m$ = "L" THEN CALL Lne(x, y, x1, y1, c())
IF m$ = "m" OR m$ = "M" THEN CALL move(x, y, x1, y1, c())
IF m$ = "d" OR m$ = "D" THEN CALL del(x, y, x1, y1, c())
IF m$ = "e" OR m$ = "E" THEN
LOCATE 23, 2
PRINT "Select: Loadpic Newpic Change Savepic Mainmenu       "
LINE (x + 1, y)-(x + 5, y + 5), 0, BF
m$ = ""
IF tnl <= vnl THEN nl = vnl ELSE nl = tnl
RETURN
END IF
plant = 0
END IF
WEND
WEND
'-------------------------Dir File---------------------------------
a2000:
LOCATE 1, 1
COLOR 15, 0
INPUT "Dir drive :", b$
IF b$="a:" OR b$="b:" OR b$="A:" OR b$="B:" THEN  ELSE GOTO a2000
CLS
FILES (b$ + "*.*")
LOCATE 23, 2: PRINT "Select : "; : COLOR 9, 0: PRINT "D";
COLOR 7, 0: PRINT "el "; : COLOR 9, 0: PRINT "R"; : COLOR 7, 0
PRINT "en "; : PRINT "d"; : COLOR 9, 0: PRINT "I"; : COLOR 7, 0
PRINT "r ";: COLOR 9, 0: PRINT "M"; : COLOR 7, 0: PRINT "ainmenu"
fi = 1
WHILE fi
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
m$ = INKEY$
IF m$ = "d" OR m$ = "D" THEN
LOCATE 22, 2: PRINT "Delete file name :";
INPUT "", gd$: KILL b$ + gd$
LOCATE 22, 2: PRINT "                                                    "
END IF
IF m$ = "r" OR m$ = "R" THEN
LOCATE 21, 2: PRINT "Old name :"; : INPUT "", od$
LOCATE 22, 2: PRINT "New name :"; : INPUT "", nd$
LOCATE 21, 2: PRINT "                                                    "
LOCATE 22, 2: PRINT "                                                    "
NAME b$ + od$ AS b$ + nd$
END IF
IF m$ = "i" OR m$ = "I" THEN
VIEW PRINT 1 TO 10
CLS : LOCATE 1, 1: COLOR 15, 0: FILES (b$ + "*.*"): COLOR 7, 0
VIEW PRINT
END IF
IF m$ = "m" OR m$ = "M" THEN fi = 0
WEND
LOCATE 23, 1: PRINT "                                                    "
RETURN
'----------------Load picture--------------------------------
a4000:
DEF SEG = &HB000
LOCATE 23, 10: PRINT "                                                   "
LOCATE 23, 10: INPUT "Drive :", a$
IF a$ = "a:"OR a$="b:" OR a$="A:" OR a$="B:" THEN  ELSE GOTO a4000
LOCATE 23, 20: INPUT "File name :", n$
p = 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
k$ = RIGHT$(STR$(p), 1)
BLOAD a$ + n$ + k$ + ".pic", 0
p = 2
LOCATE 23, 2: PRINT "Select : Continue Exit       "
Load = 1
WHILE Load
m$ = INKEY$
IF m$ = "c" OR m$ = "C" THEN
k$ = RIGHT$(STR$(p), 1)
BLOAD a$ + n$ + k$ + ".pic", 0
p = p + 1
LOCATE 23, 2: PRINT "Select : Continue Exit"
END IF
IF m$ = "e" OR m$ = "E" THEN Load = 0
WEND
p = 1
LOCATE 23, 2
PRINT "Select : Loadpic Newpic Change Save Mainmenu "
RETURN
'----------------------Save picture ------------------------------
s100:
LOCATE 23, 10
PRINT "                                                "
LOCATE 23, 10: INPUT "Drive :", d$
IF d$ ="a:" OR d$="b:" OR d$= "A:" OR d$="B:" THEN ELSE GOTOs100
LOCATE 23, 20: INPUT "File name :", f$
LOCATE 23, 40: PRINT d$ + f$
LINE (0, 306)-(712, 323), 0, BF
k$ = RIGHT$(STR$(p), 1)
DEF SEG = &HB000
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
BSAVE d$ + f$ + k$ + ".pic", 0, 40000
'-----------------Save data ---------------------------------
nl = tnl + vnl
LINE (1, 307)-(710, 322), 0, BF
s110:
OPEN d$ + f$ + ".dat" FOR RANDOM AS #2 LEN = 39
FIELD #2, 4 AS nn$, 4 AS tt$, 1 AS mm$, 4 AS qq$, 4 AS n$, 1 AS zz$
FOR k = ik TO nl
LSET nn$ = MKS$(n(k))
LSET tt$ = MKS$(t(k))
LSET mm$ = m$(k)
IF m$(k) = "c" OR m$ = "C" THEN
LSET qq$ = MKS$(0)
ELSE
LSET qq$ = MKS$(ip(k))
END IF
LSET n$ = MKS$(n(k))
LSET zz$ = z$(k)
PUT #2, k
NEXT k
CLOSE #2
IF s$ = "d" THEN RETURN
OPEN d$ + f$ + ".d" FOR RANDOM AS #1 LEN = 16
FIELD #1, 4 AS tx$, 4 AS ty$, 4 AS xx$, 4 AS yy$
FOR k = itk TO tnl
LSET tx$ = MKS$(tx(p, k))
LSET ty$ = MKS$(ty(p, k))
LSET xx$ = MKS$(x(p, k))
LSET yy$ = MKS$(y(p, k))
PUT #1, k
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
NEXT k
ik = nl: itk = tnl + 1
CLOSE #1
OPEN d$ + f$ FOR RANDOM AS #1 LEN = 8
FIELD #1, 4 AS vnl$, 4 AS tnl$
ivk = vnl - ivk
tik = tnl - tik
LSET vnl$ = MKS$(ivk)
LSET tnl$ = MKS$(tik)
PUT #1, p
p = p + 1: ivk = vnl: tik = tnl
CLOSE #1
RETURN
'-----------------------load data-----------------------------------
l100:
CLS : p = 1: nl = 0: pm = 1
l110:
LOCATE 23, 1: COLOR 15, 0: PRINT " Select :";
COLOR 7, 0: PRINT "Drive :"; : COLOR 9, 0: INPUT "", d$
IF d$="a:" OR d$="b:" OR d$="A:" OR d$ = "B:" THEN  ELSE GOTO l110
LOCATE 23, 20: COLOR 7, 0: PRINT "File name :";
COLOR 9, 0: INPUT "", f$: COLOR 7, 0
name$ = d$ + f$
LOCATE 23, 50: PRINT name$
OPEN name$ FOR RANDOM AS #1 LEN = 8
FIELD #1, 4 AS vnl$, 4 AS tnl$
WHILE NOT EOF(1)
GET #1, p
vnl(p) = CVS(vnl$)
tnl(p) = CVS(tnl$)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
nl = nl + vnl(p) + tnl(p)
tpg(p) = vnl(p) + tnl(p) + tpg(p - 1)
p = p + 1
WEND
page = p - 2
CLOSE #1
OPEN name$ + ".dat" FOR RANDOM AS #2 LEN = 39
FIELD #2, 4 AS rr$, 4 AS tt$, 1 AS mm$, 4 AS qq$, 4 AS n$, 1 AS zz$
dp = 1: p = 1
FOR k = 1 TO nl
GET #2, k
r(k) = CVS(rr$): t(k) = CVS(tt$)
m$(k) = mm$
ip(k) = CVS(qq$)
n(k) = CVS(n$)
z$(k) = zz$
NEXT k
OPEN name$ + ".d" FOR RANDOM AS #1 LEN = 16
FIELD #1, 4 AS tx$, 4 AS ty$, 4 AS xx$, 4 AS yy$
pm = 1: pkk = 1
pmm = tnl(1)
FOR p = 1 TO page
FOR pk = pm TO pmm
GET #1, pk
tx(p, pkk) = CVS(tx$)
ty(p, pkk) = CVS(ty$)
x(p, pkk) = CVS(xx$)
y(p, pkk) = CVS(yy$)
pkk = pkk + 1
NEXT pk
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
pm = pm + tnl(p): pmm = tnl(p + 1) + pmm: pkk = 1
NEXT p
CLOSE #1
'----------------------------Show data-------------------------
LOCATE 1, 1
PRINT "Data block           #"
PRINT "Resistance           ="
PRINT "Time constant        ="
PRINT "Init or Casd input   ="
PRINT "No.input port        ="
PRINT "Input from block     ="
k = 1
LOCATE 1, 52: PRINT USING "###"; k
LOCATE 2, 51: PRINT USING "####.##"; r(k)
LOCATE 3, 52: PRINT USING "###.##"; t(k)
LOCATE 4, 54: PRINT m$(k)
LOCATE 5, 52: PRINT USING "###"; ip(k)
LOCATE 6, 52: PRINT USING "###"; n(k)
CLOSE
LOCATE 23, 10: COLOR 9, 0: PRINT "C";
COLOR 7, 0: PRINT "ontinue "; : COLOR 9, 0: PRINT "E";
COLOR 7, 0: PRINT "dit "; : COLOR 9, 0: PRINT "M";
COLOR 7, 0: PRINT "ainmenu      "
WHILE 1x
m$ = INKEY$
IF m$ = "c" OR m$ = "C" THEN
k = k + 1: IF k > n1 THEN k = 1
GOSUB 11000
END IF
IF m$ = "e" OR m$ = "E" THEN GOSUB 12000
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
IF m$ = "s" OR m$ = "S" THEN GOSUB s100
IF m$ = "m" OR m$ = "M" THEN RETURN
WEND
'------------------Continue data-----------------------------------
l1000:
LOCATE 1, 52: PRINT USING "###"; k
LOCATE 2, 51: PRINT USING "####.##"; r(k)
LOCATE 3, 52: PRINT USING "###.##"; t(k)
LOCATE 4, 54: PRINT m$(k)
LOCATE 5, 52: PRINT USING "###"; ip(k)
LOCATE 6, 52: PRINT USING "###"; n(k)
RETURN
'--------------------Edit data-------------------------------------
l2000:
SCREEN 3
LOCATE 23, 1: PRINT " Select : Continue Exit                    "
LOCATE 1, 1
PRINT "Data block          #"
PRINT "Resistance          ="
PRINT "Time constant       ="
PRINT "Init or Casd input  ="
PRINT "No.input port       ="
PRINT "Input from block    ="
k = 1
l2010:
LOCATE 1, 52: PRINT USING "###"; k
LOCATE 2, 51: PRINT USING "####.##"; r(k)
LOCATE 3, 52: PRINT USING "###.##"; t(k)
LOCATE 4, 54: PRINT m$(k)
LOCATE 5, 52: PRINT USING "###"; ip(k)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LOCATE 6, 52: PRINT USING "###"; n(k)
LINE (450, 13)-(540, 26), 1, B
y = 0: Edit = 1
WHILE Edit
ex = 1
WHILE ex
m$ = INKEY$
IF m$ = "c" OR m$ = "C" THEN
LINE (450, 13 + y)-(540, 26 + y), 0, B
k = k + 1: IF k > n( THEN k = 1
GOTO 12010
END IF
IF m$ = "e" OR m$ = "E" THEN
ex = 0: Edit = 0
END IF
IF m$ <> "" THEN
mm = ASC(RIGHT$(m$, 1))
IF mm = 80 THEN
y = y + 14
IF y >= 56 THEN y = 56
LINE (450, 13 + y - 14)-(540, 26 + y - 14), 0, B
LINE (450, 13 + y)-(540, 26 + y), 1, B
END IF
IF mm = 72 THEN
y = y - 14
IF y <= 0 THEN y = 0
LINE (450, 13 + y + 14)-(540, 26 + y + 14), 0, B
LINE (450, 13 + y)-(540, 26 + y), 1, B
END IF
END IF
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
IF m$ = "s" OR m$ = "S" THEN GOTO e1100
WEND
e1100:
yy = 9 - (126 - y) / 14
ss = 0: de = 0: ye = 0: ttt = 0: tttt = 0
WHILE ex
a$ = INKEY$
IF a$ = "i" OR a$ = "I" THEN
ss$ = a$: GOTO e10010
END IF
IF a$ = "c" OR a$ = "C" THEN
ss$ = a$: GOTO e10010
END IF
IF LEN(a$) <> 0 THEN
f$(e) = a$: e = e + 1
IF a$ <> "." THEN jkk = jkk + 1 ELSE jk = jkk
END IF
IF a$ <> "s" THEN GOTO e1200
nn = e - 2: e = 0
FOR fe = 0 TO nn
IF f$(fe) = "." THEN
ye = 1: ge = 1: GOTO e10000
END IF
IF ge = 0 THEN
IF jk >= 1 THEN
ttt = ttt + VAL(f$(fe)) * 10 ^ (jk - 1 - fe)
ELSE
ttt = ttt + VAL(f$(fe)) * 10 ^ (nn - fe)
END IF
END IF
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
IF ge = 1 THEN
tttt = tttt + VAL(f$(fe)) * 10 ^ (-ye)
ye = ye + 1
END IF
ss = ttt + tttt
e10000:
NEXT fe
jk = 0: jkk = 0: ge = 0: ex = 0
IF yy = 0 THEN r(k) = ss
IF yy = 1 THEN t(k) = ss
e10010:
IF yy = 2 THEN
m$(k) = ss$: LOCATE 4, 54: PRINT m$(k): ex = 0
END IF
IF yy = 3 THEN
IF m$(k) = "c" OR m$(k) = "C" THEN ip(k) = 0
IF m$(k) = "i" OR m$(k) = "I" THEN ip(k) = ss
END IF
IF yy <> 2 THEN
LOCATE 2 + yy, 51
IF yy >= 4 THEN
IF ss >= 99 THEN ss = 99
PRINT USING "####"; ss
ELSE
PRINT USING "####.##"; ss
END IF
END IF
IF yy = 4 THEN n(k) = ss
LINE (450, 13 + y)-(540, 26 + y), 1, B
jk = 0: jkk = 0: ge = 0: ex = 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
e1200:
WEND
WEND
LINE (450, 13 + y)-(540, 26 + y), 0, B
LOCATE 23, 10: PRINT " Continue Edit Mainmenu          "
RETURN
'----------------------Run program---------------------------------
r100:
LOCATE 19, 32: COLOR 15, 0: PRINT "Select controller"
LOCATE 20, 32: PRINT " 1)";:COLOR 7,0: PRINT "Outside ":COLOR 15, 0
LOCATE 21, 32: PRINT " 2)"; : COLOR 7, 0: PRINT " Inside ":
LOCATE 22, 33: INPUT "", pid
IF pid > 2 THEN GOTO r100
IF pid = 1 THEN pid$ = "o"
IF pid = 2 THEN
pid$ = "i": Kp = 1: Ki = 1.5: Kd = 0: sv = 0
END IF
SCREEN 3
FOR k = 1 TO n1
qo(k) = 0: qi(k) = 0: h(1, k) = 0: sp(k) = 0: osp(k) = 0
NEXT k
CLS
k = 1: p = 1: sx = 0: b$ = "b": n = 1
Vn1 = 0: Vn = 0: En = 0: En1 = 0: En2 = 0
DEF SEG = &HB000
k$ = RIGHT$(STR$(p), 1)
BLOAD name$ + k$ + ".pic", 0
GOSUB k3500
GOSUB k3550
GOSUB k3600
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Ts = .4
cal = 1: n = 1: bk = 1
LOCATE 1, 1
TIME$ = "00:00:00"
LOCATE 23, 1
PRINT " Select : Bargraph Timegraph Setpoint dAta Controller Exit ";
LOCATE 23, 65: PRINT name$
WHILE cal
TIME$ = "00:00:00"
TIMER ON
IF pid$ = "i" THEN
En = sv - h(n - 1, nl)
Ic = Kd * (En - 2 * En1 + En2) / Ts
Vn = Vn1 + Kp * (En - En1) + (Ki * En * Ts) + Ic
IF Vn >= 100 THEN Vn = 100
IF Vn <= 0 THEN Vn = 0
qi(1) = Vn / 5000                    'Qin maximum = .02'
END IF
IF pid$ = "o" THEN
bass% = &H220
OUT bass% + 10, ip(1)
OUT bass% + 11, 1
hi = INP(bass% + 5)
lo = INP(bass% + 4)
v = (hi * 256 + lo - 8192) * 10 / 16384
qi(1) = (v + 5) * .002               'Qin maximum = .02'
END IF
FOR k = 1 TO nl
oq(k) = qi(k)
IF m$(k) = "c" THEN qi(k) = qo(n(k))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
cl = (r(k) * qi(k) - h(n - 1, k)) / (Ts + t(k))
h(n, k) = h(n - 1, k) + Ts * cl
IF h(n, k) >= 100 THEN h(n, k) = 100
qo(k) = h(n, k) / r(k)
NEXT k
vv = (h(n, n1) / 10) - 5
vx% = (vv / 10 * 16384) + 8192
hi% = vx% \ 256
lo% = vx% - (hi% * 256)
OUT bass% + 4, lo%
OUT bass% + 5, hi%
IF tt$ = "d" OR tt$ = "D" THEN GOTO cb300
IF b$ = "b" THEN
 '------------------Bar graph----------------------
cb100:
FOR k = tpg(p - 1) + 1 TO tpg(p)
IF z$(k) = "t" THEN
sx = sx + 1
IF sx <= tn1(p) THEN
ht = h(n, k) * 5000 / r(k): hth = h(n - 1, k) * 5000 / r(k)
LINE (x(p, sx), y(p, sx) + 80)-(x(p, sx) + 105, y(p, sx) + 80)
VIEW (x(p, sx), y(p, sx))-(x(p, sx) + 105, y(p, sx) + 80)
WINDOW (0, 0)-(105, 101)
IF sv <= osv THEN LINE (15, osv)-(30, sv), 0, BF
IF qi(k)<=oq(k) THEN LINE(45,oq(k)*5000)-(60,qi(k)*5000), 0, BF
IF h(n, k) >= h(n - 1, k) THEN GOTO c100
LINE (75, ht)-(90, hth), 0, BF
c100:
LINE (15, 0)-(30, sv), 1, BF
LINE (45, 0)-(60, qi(k) * 5000), 1, BF
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LINE (75, 0)-(90, ht), 1, BF
LOCATE posX(sx), posY(sx) - 8
PRINT USING "###.##"; sv
LOCATE posX(sx), posY(sx)
PRINT USING "###.##"; ht
WINDOW
VIEW
 ' ---------------Show  Level----------------------
VIEW (tx(p, sx) + 1, ty(p, sx))-(tx(p, sx) + 109, ty(p, sx) + 125)
WINDOW (0, 0)-(100, 125)
'IF h(n, k) < hc(k) THEN LINE (0, h(n, k))-(100, hc(k)), 0, BF
IF h(n, k) >= h(n - 1, k) THEN GOTO cc100
LINE (0, ht)-(100, hth), 0, BF
cc100:
LINE (0, 0)-(100, ht), 1, BF
WINDOW
VIEW
END IF
IF sx = tn((p) THEN sx = 0
END IF
NEXT k
END IF
IF b$ = "t" THEN
 ' ---------------Show Timegraph-------------------
cb200:
VIEW (50, 81)-(650, 300), , 1
WINDOW (0, 0)-(600, 125)
LOCATE 5, 32: PRINT USING "###.##"; qi(bk) * 5000
LOCATE 5, 45: PRINT USING "###.##"; sv
LOCATE 5, 62: PRINT USING "###.##"; h(n, bk) * 5000 / r(bk)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
IF osv <> sv THEN
LINE (x, osv)-(x, sv), , , &HFCFC
osv = sv
END IF
IF cc = 7 THEN
c = 0: cc = 0
ELSE c = 1: cc = cc + 1
END IF
PSET (x, sv), c
IF oq(bk)<>qi(bk) THEN LINE(x,oq(bk)*5000)-(x,qi(bk)*5000),1
IF oq(bk)=qi(bk) THEN PSET(x,qi(bk)*5000)
IF h(n-1,bk)<>h(n,bk) THEN
LINE(x,h(n-1,bk)*5000/r(bk))-(x,h(n,bk)*5000/r(bk)),1
END IF
PSET (x, h(n, bk) * 5000 / r(bk))
x = x + 1
IF x >= 600 THEN
CLS : x = 0
LINE (0, 100)-(600, 100), , , &HF0F0
LINE (0, 75)-(600, 75), , , &HF0F0
LINE (0, 50)-(600, 50), , , &HF0F0
LINE (0, 25)-(600, 25), , , &HF0F0
FOR mx = 100 TO 600 STEP 100
LINE (mx, 0)-(mx, 125), , , &HF0F0
NEXT mx
END IF
WINDOW
VIEW
END IF
'---------------<<Show PID-controller>>----------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
IF b$ = "c" THEN
VIEW (10, 10)-(220, 120), , 1
WINDOW (0, 0)-(250, 120)
IF h(n,n1)<>h(n-1,n1) THEN LINE(x,h(n,n1))-(x,h(n-1,n1))
PSET (x, h(n, n1))
IF x >= 250 THEN CLS
LOCATE 7, 30: PRINT USING "###.##"; h(n, n1)
WINDOW
VIEW
VIEW (10, 130)-(220, 250), , 1
WINDOW (0, 0)-(250, 120)
IF Vn <> Vn1 THEN LINE (x, Vn)-(x, Vn1)
PSET (x, Vn)
IF x >= 250 THEN CLS
LOCATE 16, 30: PRINT USING "###.##"; Vn
WINDOW
VIEW
VIEW (410, 72)-(480, 220)
WINDOW (0, 0)-(71, 100)
IF sv < osv THEN LINE (1, sv)-(11, osv), 0, BF
LINE (1, 0)-(11, sv), 1, BF
IF h(n, n1)<h(n-1,n1)THEN LINE(60,h(n,n1))-(70,h(n-1,n1)), 0, BF
LINE (60, 0)-(70, h(n, n1)), 1, BF
WINDOW
VIEW
IF x >= 250 THEN x = 0
x = x + 1
END IF
cb300:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
IF tt$ = "s" THEN GOTO k1100
IF tt$ = "t" THEN GOTO k2100
IF tt$ = "b" THEN GOTO k3100
IF tt$ = "a" THEN GOTO k4100
IF tt$ = "x" THEN GOTO k10000
IF tt$ = "p" THEN GOTO kp110
IF tt$ = "i" THEN GOTO ki110
IF tt$ = "d" THEN GOTO kd110
m$ = INKEY$
IF m$ = "e" OR m$ = "E" THEN cal = 0
IF m$ = "s" OR m$ = "S" THEN GOTO k1000
IF m$ = "t" OR m$ = "T" THEN GOTO k2000
IF m$ = "b" OR m$ = "B" THEN GOTO k3000
IF m$ = "a" OR m$ = "A" THEN GOTO k4000
IF m$ = "c" OR m$ = "C" THEN GOTO k5000
IF m$ = "p" OR m$ = "P" THEN GOTO kp100
IF m$ = "i" OR m$ = "I" THEN GOTO ki100
IF m$ = "d" OR m$ = "D" THEN GOTO kd100
k200:
n = n + 1
IF n >= 99 THEN
FOR k = 1 TO nl
h(0, k) = h(n - 1, k)
NEXT k
n = 1
END IF
k300:
IF TIME$ < "00:00:01" THEN GOTO k300
IF pid$ = "i" THEN
Vn1 = Vn: En2 = En1: En1 = En
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
END IF
WEND
n = 0
FOR k = 1 TO nl
h(0, k) = 0
NEXT k
RETURN
END
k1000:
tt$ = "s"
k1100:
osv = sv
GOSUB k10000
IF ss > 100 THEN ss = 100
IF ss <= 0 THEN ss = 0
sv = ss
IF b$ = "c" THEN


LOCATE 7, 66: PRINT USING "###.##"; sv
END IF
tt$ = "": GOTO k200
k2000:
tt$ = m$
b = 0
k2100:
bm$ = INKEY$
IF LEN(bm$) <> 0 THEN
g$(b) = bm$: b = b + 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
END IF
IF bm$ <> "s" THEN GOTO k200
bb = b - 2: b = 0: bk = 0
FOR be = 0 TO bb
bk = bk + VAL(g$(be)) * 10 ^ (bb - be)
NEXT be
k2200:
CLS
LOCATE 23, 1
PRINT " Select : Bargraph Timegraph Setpoint dAta Controller Exit ";
LOCATE 23, 65: PRINT name$
LINE (49, 50)-(651, 301), 1, B
LOCATE 5, 8
PRINT "Time graph #    Input          Sp          Level"
LINE (50, 80)-(650, 80): LINE (200, 50)-(200, 80)
LINE (230, 75)-(300, 75)
LINE (370, 75)-(440, 75), , , &HFFF0
LINE (500, 75)-(570, 75)
VIEW (50, 81)-(650, 300), , 1
WINDOW (0, 0)-(600, 125)
LINE (0, 100)-(600, 100), , , &HF0F0
LINE (0, 75)-(600, 75), , , &HF0F0
LINE (0, 50)-(600, 50), , , &HF0F0
LINE (0, 25)-(600, 25), , , &HF0F0
FOR mx = 100 TO 600 STEP 100
LINE (mx, 0)-(mx, 125), , , &HF0F0
NEXT mx
WINDOW
VIEW
LOCATE 10, 74: PRINT "100"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LOCATE 13, 74: PRINT "75" : LOCATE 16, 74: PRINT "50"
LOCATE 19, 74: PRINT "25" : LOCATE 22, 74: PRINT "0"
IF bk >= nl THEN bk = nl : LOCATE 5, 20: PRINT bk
b$ = "t": tt$ = "": x = 1
FOR k = 1 TO nl
hc(k) = h(n, k)
NEXT k
GOTO k200
k3000:
IF tt$ = m$ THEN GOTO k200
tt$ = m$
b = 0
k3100:
bm$ = INKEY$
IF LEN(bm$) <> 0 THEN
g$(b) = bm$: b = b + 1
END IF
IF bm$ <> "s" THEN GOTO k200
bb = b - 2: b = 0: bk = 0
FOR be = 0 TO bb
bk = bk + VAL(g$(be)) * 10 ^ (bb - be)
NEXT be
b$ = "b": tt$ = ""
IF bk >= page THEN bk = page
p = bk
CLS
LOCATE 23, 1
PRINT " Select : Bargraph Timegraph Setpoint dAta Controller Exit   "
LOCATE 23, 65: PRINT name$
DEF SEG = &HB000
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
k$ = RIGHT$(STR$(p), 1)
BLOAD d$ + f$ + k$ + ".pic", 0
LOCATE 23, 1
PRINT " Select : Bargraph Timegraph Setpoint dAta Controller Exit    "
LOCATE 23, 65: PRINT name$
GOSUB k3500
GOSUB k3550
GOSUB k3600
GOTO k200
k4000:
tt$ = m$
DIM dat(5125)
GET (0, 0)-(400, 100), dat
LINE (0, 0)-(400, 100), 0, BF
LOCATE 23, 1: PRINT " Select : Continue Exit                    "
LOCATE 1, 1
PRINT "Data block          #"
PRINT "Resistance          ="
PRINT "Time constant       ="
PRINT "Init or Casd input  ="
PRINT "No.input port       ="
PRINT "Input from block    ="
dk = 1
LOCATE 1, 27: PRINT USING "###"; dk
LOCATE 2, 26: PRINT USING "####.##"; r(dk)
LOCATE 3, 27: PRINT USING "###.##"; t(dk)
LOCATE 4, 29: PRINT m$(k)
LOCATE 5, 27: PRINT USING "###"; ip(dk)
LOCATE 6, 27: PRINT USING "###"; n(dk)
GOTO k200
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
k4100:
m$ = INKEY$
IF m$ = "e" OR m$ = "E" THEN
LOCATE 23, 10
PRINT " Select : Bargraph Timegraph Setpoint dAta Controller Exit   "
LINE (0, 0)-(400, 100), 0, BF
PUT (0, 0), dat
tt$ = ""
IF b$ = "t" OR b$ = "T" THEN GOTO k2200
END IF
IF m$ = "c" OR m$ = "C" THEN
dk = dk + 1
IF dk > n1 THEN dk = 1
LOCATE 1, 27: PRINT USING "###"; dk
LOCATE 2, 26: PRINT USING "####.##"; r(dk)
LOCATE 3, 27: PRINT USING "###.##"; t(dk)
LOCATE 4, 29: PRINT m$(dk)
LOCATE 5, 27: PRINT USING "###.##"; qi(dk)
LOCATE 6, 27: PRINT USING "###"; n(dk)
END IF
GOTO k200
k10000:
ss = 0: de = 0: ye = 0: ttt = 0: tttt = 0
a$ = INKEY$
IF LEN(a$) <> 0 THEN
f$(e) = a$: e = e + 1
IF a$ <> "." THEN jkk = jkk + 1 ELSE jk = jkk
END IF
IF a$ <> "s" THEN GOTO k200
nn = e - 2: e = 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
FOR fe = 0 TO nn
IF f$(fe) = "." THEN
ye = 1: ge = 1: GOTO k10200
END IF
IF ge = 0 THEN
IF jk >= 1 THEN
ttt = ttt + VAL(f$(fe)) * 10 ^ (jk - 1 - fe)
ELSE
ttt = ttt + VAL(f$(fe)) * 10 ^ (nn - fe)
END IF
END IF
IF ge = 1 THEN
tttt = tttt + VAL(f$(fe)) * 10 ^ (-ye)
ye = ye + 1
END IF
ss = ttt + tttt
k10200:
NEXT fe
k10300:
jk = 0: jkk = 0: ge = 0
RETURN
'----------------------Write bar-------------------------------
k3500:
RESTORE
DIM a%(10)
a%(0) = 8: a%(1) = 8
a%(8) = 0: a%(9) = 0
FOR vm = 1 TO tn!(p)
LINE (x(p, vm), y(p, vm))-(x(p, vm) + 105, y(p, vm) + 80), 1, B
x = x(p, vm) + 15: y = y(p, vm) + 85: s = 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
FOR k = 1 TO 9
FOR i = 2 TO 7
READ a%(i)
NEXT i
s = s + 1
PUT (x, y), a%
IF s = 2 OR s = 5 THEN x = x + 16 ELSE x = x + 8
NEXT k
RESTORE
NEXT vm
RESTORE k3570
DATA -30608,8256, -30704,112,56,56
DATA 0,17592,30788,16448,56,56
DATA -31624 ,-31612,-29548,636,56,56
DATA 32,8224,8224,32,56,56
DATA 0,-15176,-31612,132,56,56
DATA -31624,-31612,-29548,636,56,56
DATA 0,-31624,-31612,120,56,56
DATA 0,-30584 ,-26488,108,56,56
DATA 8224,8440,11296,56,56,56
RETURN
'-----------------<<draw valve>>-------------------------------
k3550:
a%(0) = 8
a%(1) = 8
a%(6) = 0: a%(7) = 0: a%(8) = 0: a%(9) = 0: a%(10) = 0
FOR vm = 1 TO tnl(p)
x = x(p, vm) + 107: y = y(p, vm) - 4: jj = 1
FOR k = 2 TO 5
READ a%(k)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
NEXT k
PUT (x, y + 80), a%
FOR j = 1 TO 3
FOR i = 0 TO 1
FOR k = 2 TO 5
READ a%(k)
NEXT k
PUT (x + i * 9, y + 20 * j), a%
NEXT i
NEXT j
FOR i = 0 TO 2
FOR k = 2 TO 5
READ a%(k)
NEXT k
PUT (x + i * 9, y), a%
NEXT i
RESTORE k3570
NEXT vm
k3570:
DATA -29576,-19300,-15132,120
DATA -31492,2052,4112,16
DATA -32516,1272,-31740,120
DATA -32516,1272,-31740,120
DATA -29576,-19300,-15132,120
DATA -31624,14340,-32704,252
DATA -32516,1272,-31740,120
DATA 24608,8224,8224,112
DATA -29576,-19300,-15132,120
DATA -29576,-19300,-15132,120
RETURN
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
k3600:
FOR vm = 1 TO tnl(p)
y = x(p, vm) + 80: x = y(p, vm) + 93
posX(vm) = x ■ 14: posY(vm) = y ■ 9
xxx = (x / 14) - posX(vm)
yyy = (y / 9) - posY(vm)
IF xxx > 0 THEN posX(vm) = posX(vm) + 2 ELSE posX(vm) = posX(vm) + 1
IF yyy > 0 THEN porY(vm) = posY(vm) + 1
NEXT vm
RETURN
'----------------------<<End>>-----------------------------------
'----------------------<<Main menu>>-----------------------------
ik10:
CLS
SCREEN 0
FOR y = 28 TO 50
LOCATE 4, y: PRINT CHR$(178)
LOCATE 20, y: PRINT CHR$(178)
NEXT y
FOR x = 5 TO 19
LOCATE x, 29: PRINT CHR$(186)
LOCATE x, 49: PRINT CHR$(186)
NEXT x
LOCATE 5, 35: COLOR 9, 0: PRINT "Main menu"
LOCATE 7, 32: PRINT "L"; : COLOR 7, 0: PRINT "oad process "
LOCATE 9, 32:COLOR 9,0:PRINT "N";:COLOR 7, 0:PRINT "ew process"
LOCATE 11, 32: COLOR 9, 0:PRINT "R";: COLOR 7,0:PRINT "un process"
LOCATE 13, 32: COLOR 9,0:PRINT "F";:COLOR 7, 0: PRINT "ile name"
LOCATE 15, 32: COLOR 9,0:PRINT "E";:COLOR 7, 0: PRINT "dit data"
LOCATE 17, 32: COLOR 9,0:PRINT "S";:COLOR 7, 0: PRINT "ave data"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LOCATE 19, 32: COLOR 9, 0: PRINT "Q"; : COLOR 7, 0: PRINT "uit "
LOCATE 23, 32: COLOR 9, 0: PRINT.d$ + f$: COLOR 7, 0
RETURN
'---------------<<PID-controller>>----------------------
k5000:
IF b$ = "c" THEN GOTO k200
RESTORE k5500
k5500:
a%(0) = 8
a%(1) = 8
a%(6) = 0: a%(7) = 0: a%(8) = 0: a%(9) = 0
b$ = "c": y = 65: CLS : x = 0
LOCATE 19, 11: PRINT "Sv +   En              Vn                Pv"
LOCATE 20, 25: PRINT "Controller          Process"
LOCATE 21, 20: PRINT "-"
DRAW "bm 100,272 r30 ng5 h5 bm 170,272 r35 ng5 h5"
CIRCLE (150, 272), 20
DRAW "bm 205,260 r110d25l110u25 bm315,272 r63ng5nh5nd13u12r80d25l80"
DRAW "bm 458,272 r60ng5nh5 bm 480,272 d30 l330 u16 ng5 f5"
LOCATE 17, 47: PRINT "SV % PV"
LOCATE 7, 57: PRINT "SV"
LOCATE 11, 57: PRINT "PID"
LOCATE 13, 57: PRINT "PI"
LOCATE 15, 57: PRINT "P"
LOCATE 17, 57: PRINT "mOde"
DRAW "bm 500,62 r30 d20 l30 nu20 bm 515,78 u12 ng5 f5"
DRAW "bm 500,98 r30 d20 l30 nu20 bm 515,102 d12 nh5 e5"
LINE (500, 135)-(532, 156), 1, B
LINE (500, 164)-(532, 182), 1, B
LINE (500, 192)-(532, 210), 1, B
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LINE (400, 60)-(542, 250), 1, B
LINE (410, 71)-(432, 220), 1, B
LINE (458, 71)-(480, 220), 1, B
LINE (490, 60)-(490, 250)
VIEW (410, 71)-(480, 220)
WINDOW (0, 0)-(70, 150)
LINE (12, 0)-(12, 150)
LINE (58, 0)-(58, 150)
FOR i = 1 TO 10
LINE (12, 15 * i)-(21, 15 * i)
NEXT i
FOR i = 1 TO 10
LINE (49, 15 * i)-(58, 15 * i)
NEXT i
WINDOW
VIEW
FOR k = 2 TO 5
READ a%(k)
NEXT k
PUT (434, 65), a%
DATA 24608,8224,8224,112
FOR i = 1 TO 5
FOR k = 2 TO 5
READ a%(k)
NEXT k
PUT (442, y), a%
y = y + 30
NEXT i
DATA -29576,-19300,-15132,120
DATA 16956,15426,16962,60
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DATA 16440,-1920,-31612,120
DATA 10264,-30648,2296,28
DATA -31624,14340,-32704,252
FOR k = 2 TO 5
READ a%(k)
NEXT k
DATA -29576,-19300,-15132,120
y = 65
FOR i = 1 TO 6
PUT (450, y), a%
y = y + 30
NEXT i
LOCATE 15, 62: PRINT "Kp ="; Kp
LOCATE 13, 62: PRINT "Ki ="; Ki
LOCATE 11, 62: PRINT "Kd ="; Kd
LOCATE 7, 62: PRINT "Sv ="; sv
LOCATE 7, 26: PRINT "Pv ="
LOCATE 16, 26: PRINT "Vn ="
LOCATE 23, 1
PRINT " Select :Bargraph Timegraph Setpoint dAta Controller Exit "
GOTO k200
kp100:
tt$ = "p"
kp110:
GOSUB k10000
IF ss > 100 THEN ss = 100
IF ss <= 0 THEN ss = 0
Kp = ss
IF b$ = "c" THEN
LOCATE 15, 66: PRINT USING "###.##"; Kp
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
END IF
tt$ = "": GOTO k200
ki100:
tt$ = "i"
ki110:
GOSUB k10000
IF ss > 100 THEN ss = 100
IF ss <= 0 THEN ss = 0
Ki = ss
IF b$ = "c" THEN
LOCATE 13, 66: PRINT USING "###.##"; Ki
END IF
tt$ = "": GOTO k200
kd100:
tt$ = "d"
kd110:
GOSUB k10000
IF ss > 100 THEN ss = 100
IF ss <= 0 THEN ss = 0
Kd = ss
IF b$ = "c" THEN
LOCATE 11, 66: PRINT USING "###.##"; Kd
END IF
tt$ = "": GOTO k200
END
SUB bargraph (p, tnl, x(), y())
DIM bag(1590)
GET (0, 0)-(200, 60), bag
LINE (0, 0)-(200, 60), 1, B
LOCATE 23, 10
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PRINT " Continue Edit                                    "
bg10:
LOCATE 1, 1: INPUT "X pos =", x(p, tn1)
bg20:
LOCATE 2, 1: INPUT "Y pos =", y(p, tn1)
IF x + 105 > 710 THEN GOTO bg10
IF y + 90 > 300 THEN GOTO bg20
LINE (x(p, tn1), y(p, tn1))-(x(p, tn1) + 105, y(p, tn1) + 90), 1, B
LOCATE 3, 1: INPUT " Continue or Edit ", m$
IF m$ = "e" OR m$ = "E" THEN
LINE (x(p, tn1), y(p, tn1))-(x(p, tn1) + 105, y(p, tn1) + 90), 0, B
GOTO bg10
END IF
IF m$ = "c" OR m$ = "C" THEN
LINE (0, 0)-(200, 60), 0, BF
LINE (x(p, tn1), y(p, tn1))-(x(p, tn1) + 105, y(p, tn1) + 90), 0, B
END IF
PUT (0, 0), bag
END SUB
SUB box (x, y, x1, y1, c())
bb10:
CALL crs(x, y, c())
x2 = x: y2 = y
IF x2 > 710 THEN GOTO bb10
IF y2 > 305 THEN GOTO bb10
LINE (x1, y1)-(x2, y2), 1, B
END SUB

SUB Ccle (x1, y1)
cpcess = 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
IF x1 = 0 OR y1 = 0 THEN GOTO cb10
IF x1 + 15 > 710 THEN GOTO cb10
IF y1 + 15 > 305 THEN GOTO cb10
CIRCLE (x1, y1), 15
cb10:
END SUB
SUB crs (x, y, c())
cpcess = 1
VIEW (1, 307)-(710, 322), , 1
CLS
LOCATE 23, 2: PRINT "Choose:"
DRAW "bm82,315 r15 nh5 ng5 bm107,315 ne5 nf5 r15"
DRAW "bm142,320 u10 ng5 nf5 bm162,320 nu10 nh5 e5"
LOCATE 23, 21: PRINT "Stop"
VIEW
WHILE cpcess
m$ = INKEY$
IF m$ <> "" THEN
PUT (x, y), c, XOR
mm = ASC(RIGHT$(m$, 1))
IF mm = 77 THEN x = x + 1
IF mm = 75 THEN x = x - 1
IF mm = 80 THEN y = y + 1
IF mm = 72 THEN y = y - 1
IF m$ = "s" OR m$ = "S" THEN cpcess = 0
IF x <= 0 THEN x = 0
IF x + 5 >= 710 THEN x = 710 - 5
IF y <= 0 THEN y = 0
IF y + 5 > 305 THEN y = 305 - 5
PUT (x, y), c
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LOCATE 23, 70: PRINT x; y
END IF
WEND
END SUB
SUB del (x, y, x1, y1, c())
dpcess = 1
db10:
CALL crs(x, y, c())
x2 = x: y2 = y
xx = ABS(x2 - x1): yy = ABS(y2 - y1)
size = 4 + INT((xx + 8) / 8) * (yy + 1)
DIM d(size)
GET (x1, y1)-(x2, y2), d
IF x1 > x2 THEN x1 = x2
IF y1 > y2 THEN y1 = y2
LOCATE 23, 10
PRINT "Look Del Exit                    "
WHILE dpcess
m$ = INKEY$
IF m$ = "l" OR m$ = "L" THEN GOSUB db20
IF m$ = "d" OR m$ = "D" THEN GOTO db30
IF m$ = "e" OR m$ = "E" THEN
dpcess = 0: GOTO db40
END IF
WEND
db20:
PUT (x1, y1), d, PRESET
PUT (x1, y1), d, PSET
RETURN
db30:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LINE (x1, y1)-(x1 + 4, y1 + 4), 0, BF
PUT (x1, y1), d, XOR
db40:
END SUB

SUB Joint (x, y)
SCREEN 3
x$ = STR$(x): y$ = STR$(y)
DIM jj(5000)
GET (0, 0)-(85, 90), jj
LINE (0, 0)-(85, 90), 1, BF
DRAW "c0 bm10,20 u10 e5 r20 bd5 l14 g3 d7"
DRAW "c0 bm70,20 u7 h3 l14 bu5 r20 f5 d10"
DRAW "c0 bm10,50 d10 f5 r20 bu5 l14 h3 u7"
DRAW "c0 bm70,50 d7 g3 l14 bd5 r20 e5 u10"
LOCATE 2, 4: PRINT "1"
LOCATE 2, 7: PRINT "2"
LOCATE 4, 4: PRINT "3"
LOCATE 4, 7: PRINT "4"
LOCATE 6, 3: PRINT "Select"
kj = 1
WHILE kj
j$ = INKEY$
IF VAL(j$) > 0 AND VAL(j$) <= 4 THEN
LINE (0, 0)-(85, 90), 0, BF
IF j$="1"THEN DRAW"c1 bm"+x$+","+y$+"u10 e5 r20 bd5 l14 g3 d7"
IF j$="2"THEN DRAW"c1 bm" +x$+","+y$+"u7 h3 l14 bu5 r20 f5 d10"
IF j$="3"THEN DRAW"c1 bm"+x$+","+y$+" d10 f5 r20 bu5 l14 h3 u7"
IF j$="4"THEN DRAW"c1 bm"+x$+","+ y$+"d7 g3 l14 bd5 r20 e5 u10"
kj = 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PUT (0, 0), jj
END IF
WEND
LOCATE 23, 2
PRINT "Select:Valve Pipe Tank Box pUmp Joint Circle Line Del Exit"
END SUB
SUB Lne (x, y, x1, y1, c())
lpcess = 1
VIEW (1, 307)-(700, 322), , 1
CLS
LOCATE 23, 2: PRINT "Choose:"
DRAW "bm82,315 r15 nh5 ng5 bm107,315 ne5 nf5 r15"
DRAW "bm142,320 u10 ng5 nf5 bm162,320 nu10 nh5 e5"
LOCATE 23, 21: PRINT "Stop"
VIEW
WHILE lpcess
m$ = INKEY$
IF m$ <> "" THEN
PUT (x1, y1), c, XOR
PSET (x1, y1)
mm = ASC(RIGHT$(m$, 1))
IF mm = 77 THEN x1 = x1 + 1
IF mm = 75 THEN x1 = x1 - 1
IF mm = 80 THEN y1 = y1 + 1
IF mm = 72 THEN y1 = y1 - 1
IF m$ = "s" OR m$ = "S" THEN lpcess = 0
IF x1 <= 0 THEN x1 = 0
IF x1 >= 710 THEN x1 = 710
IF y1 <= 0 THEN y1 = 0
IF y1 > 305 THEN y1 = 305
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PUT (x1, y1), c
LOCATE 23, 67: PRINT x1; y1
END IF
WEND
x = x1: y = y1
END SUB
SUB move (x, y, x1, y1, c())
mpcess = 1
CALL crs(x, y, c())
x2 = x: y2 = y
xx = ABS(x2 - x1): yy = ABS(y2 - y1)
PUT (x, y), c, XOR
size = 4 + INT((xx + 8) / 8) * (yy + 1)
DIM m(size)
GET (x1, y1)-(x2, y2), m
IF x1 > x2 THEN x1 = x2
IF y1 > y2 THEN y1 = y2
LOCATE 23, 10
PRINT "Look Move Exit                "
WHILE mpcess
m$ = INKEY$
IF m$ = "l" OR m$ = "L" THEN
PUT (x1, y1), m, PRESET
PUT (x1, y1), m, PSET
END IF
IF m$ = "m" OR m$ = "M" THEN GOTO mb30
IF m$ = "e" OR m$ = "E" THEN
mpcess = 0: GOTO mb40
END IF
WEND
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
mb30:
VIEW (1, 307)-(710, 322), , 1
CLS
LOCATE 23, 2: PRINT "Choose:"
DRAW "bm82,315 r15 nh5 ng5 bm107,315 ne5 nf5 r15"
DRAW "bm142,320 u10 ng5 nf5 bm162,320 nu10 nh5 e5"
LOCATE 23, 21: PRINT "Stop"
VIEW
WHILE mpcess
m$ = INKEY$
IF m$ <> "" THEN
PUT (x1, y1), m, XOR
mm = ASC(RIGHT$(m$, 1))
IF mm = 77 THEN x1 = x1 + 1
IF mm = 75 THEN x1 = x1 - 1
IF mm = 80 THEN y1 = y1 + 1
IF mm = 72 THEN y1 = y1 - 1
IF m$ = "s" OR m$ = "S" THEN mpcess = 0
IF x1 <= 0 THEN x1 = 0
IF x1 + xx >= 710 THEN x1 = 710 - xx
IF y1 <= 0 THEN y1 = 0
IF y1 + yy >= 305 THEN y1 = 305 - yy
PUT (x1, y1), m
LOCATE 23, 60: PRINT x1; y1
END IF
WEND
mb40:
x = x1: y = y1
PUT (x, y), c
END SUB
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SUB pipe (x, y, x1, y1, c())
ppcess = 1
pb10:
CALL crs(x, y, c())
x2 = x: y2 = y
LOCATE 23, 26
INPUT "Hor or Ver ", pa$
px$ = STR$(x1): py$ = STR$(y1)
IF pa$ = "h" OR pa$ = "H" THEN
pl$ = STR$(x2 - x1)
DRAW "bm" + px$ + "," + py$
DRAW "c1" + "r" + pl$ + "bd5"
DRAW "l" + pl$ + "bu5"
END IF
IF pa$ = "v" OR pa$ = "V" THEN
pl$ = STR$(y2 - y1)
DRAW "bm" + px$ + "," + py$
DRAW "c1" + "d" + pl$ + "br7"
DRAW "u" + pl$ + "bl7"
END IF
LOCATE 23, 26: PRINT "                "
END SUB
SUB pump (x, y, x1, y1, c())
IF x1 = 0 OR y1 = 0 THEN GOTO ub10
IF x1 + 10 >= 710 THEN GOTO ub10
IF y1 + 18 >= 305 THEN GOTO ub10
LINE (x1, y1)-(x1 + 5, y1 + 5), 0, BF
CIRCLE (x1, y1), 3
CIRCLE (x1, y1), 10
pp$ = STR$(x1 + 4)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ppp$ = STR$(y1 + 7)
DRAW "bm" + pp$ + "," + ppp$
DRAW "f7 l21 e7"
x = x1 - 5: y = y1 - 5
PUT (x, y), c
ub10:
END SUB
SUB tank(p,x(),y(),z$(),tnl,x1,y1,tx(),ty(),r(),t(),ip(),n(),m$())
IF x1 + 110 >= 710 THEN GOTO tb10
IF y1 + 125 >= 305 THEN GOTO tb10
LOCATE 23, 10
PRINT "                                                    "
LOCATE 23, 10: INPUT " Tank num # ", k
tnl = tnl + 1
tx(p, tnl) = x1: ty(p, tnl) = y1
tx$ = STR$(x1): ty$ = STR$(y1)
DRAW "bm" + tx$ + "," + ty$
DRAW "c1d125f5r100e5nl110nu125r30u5nl30u2nd10f10u10g5ng5u10nl5r5"
z$(k) = "t"
CALL Tdata(k, r(), t(), ip(), n(), m$())
CALL bargraph(p, tnl, x(), y())
tb10:
END SUB
SUB Tdata (k, r(), t(), ip(), n(), m$())
DIM tdat(5155)
GET (0, 0)-(400, 100), tdat
LOCATE 1, 1
PRINT "Data block                "
PRINT "Resistance                "
PRINT "Time constance            "
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PRINT "Init or Casd input       "
PRINT "No.input port <<1-16>>   "
PRINT "Input from block         "
c10:
cx =-1: ccx = 1
LOCATE 1, 30: PRINT "#"; k
c15:
LOCATE 2, 30: INPUT "= ", r$
r(k) = VAL(r$)
IF r(k) <= 0 OR r(k) > 10000 THEN GOTO c15
c17:
LOCATE 3, 30: INPUT "= ", t$
t(k) = VAL(t$)
IF t(k) <= 0 OR t(k) > 100 THEN GOTO c17
WHILE ccx
m$ = INKEY$
IF m$ = "i" OR m$ = "I" OR m$ = "c" OR m$ = "C" THEN
m$(k) = m$: ccx = 0
LOCATE 4, 30: PRINT ": "; m$
END IF
WEND
IF m$(k) = "i" OR m$(k) = "I" THEN
c20:
LOCATE 5, 30: INPUT "= ", ip$
ip(k) = VAL(ip$)
IF ip(k) <= 0 OR ip(k) > 16 THEN GOTO c20
END IF
IF m$(k) = "c" OR m$(k) = "C" THEN
c25:
LOCATE 6, 30: INPUT "= ", n$
```

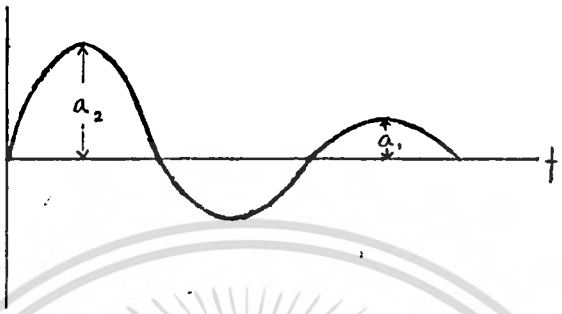
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
n(k) = VAL(n$)
IF n(k) <= 0 OR n(k) > 99 THEN GOTO c25
END IF
LOCATE 23, 1: PRINT " Select : Edit Continue                    "
WHILE cx
m$ = INKEY$
IF m$ = "e" OR m$ = "E" THEN GOTO c10
IF m$ = "c" OR m$ = "C" THEN cx = 0
WEND
LINE (0, 0)-(400, 100), 0, BF
PUT (0, 0), tdat
END SUB
SUB valve (p,z$() ,vnl,x,y,x1,y1,c(),r(),t(),ip(),n(),m$())
IF x1 + 10 >= 710 THEN GOTO vb10
IF y1 + 10 >= 305 THEN GOTO vb10
LINE (x1, y1)-(x1 + 5, y1 + 5), 0, BF
LOCATE 23, 10
PRINT "                                                    "
LOCATE 23, 10: INPUT " Valve num # ", k
vv$ = STR$(x1): vvv$ = STR$(y1)
DRAW "bm" + vv$ + "," + vvv$
DRAW "c1 r5 nr5 d10 nf5 h5 d10 e10 d10 "
x = x1 - 5: y = y1 - 5
PUT (x, y), c
CALL Tdata(k, r(), t(), ip(), n(), m$())
vnl = vnl + 1
z$(k) = "v"
vb10:
END SUB
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

หลักการในการปรับค่า Kp , Ti และ Td ของ PI-controller มีดังนี้คือ เราจะพิจรณาที่ Damping ratio criterion โดยที่ค่าอัตราส่วน a1/a2 มีค่าเท่ากับ 1/4 ดังรูปที่ 20



รูปที่ 20

การปรับจะใช้วิธี ลองผิดลองถูก (Trial and error) ซึ่งมีขั้นตอนดังนี้คือ

1. ปรับตัวควบคุมไปที่ M (manaul mode)
2. ปรับค่าอัตราขยาย (proportional band) ไปสูงสุด และ ค่าอินติกรัลไทม์ (Ti) สูงสุด , ค่าดิริวิทีฟไทม์ (Td) ต่ำสุด
3. ปรับค่าเป้าหมาย (setpoint) ไปที่ค่าที่ต้องการ
4. ปรับการควบคุมด้วยมือ (manaul control) จนกระทั่งตัวแปรกระบวนการมีค่าเท่ากับค่าเป้าหมาย
5. ปรับการควบคุมไปที่การทำงานแบบอัตโนมัติ (Automatic mode)
6. เปลี่ยนค่าเป้าหมายไปเล็กน้อย เมื่อค่าที่วัดได้เริ่มเปลี่ยน ให้ลดค่าเป้าหมายมาที่เดิม
7. ลดค่าอัตราการขยายลงมาและทำตามขั้นตอนที่ 1-6 ใหม่โดยสังเกตผลตอบสนองของตัวแปรกระบวนการ
8. ทำขั้นที่ 6 และ 7 หลายๆครั้ง จนได้ค่าอัตราส่วน a1/a2 = 1/4=25%
9. ลดค่าอินติกรัลไทม์ จนกระทั่งค่าผิดพลาด (offset) หายไป
10. เพิ่มค่าอินติกรัลไทม์ ถ้าเกิดการแกว่ง
11. ทำตามขั้นตอน 10 และ 11 จนได้อัตราส่วน a1/a2 = 1/4 =25%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

จากผลการทดลองในบทที่ 4 นั้น ได้มาจากการทดลองเชื่อมต่อกระบวนการที่ทำการจำลองขึ้นกับตัวควบคุมภายนอก ซึ่งเป็นตัวควบคุมแบบดิจิตอลของบริษัท ฟูจิ อิเล็คทริค จำกัด รุ่น FC series compact Controller F ซึ่งเป็นตัวควบคุมแบบลูปเดียวบนพื้นฐานไมโครโปรเซสเซอร์ และ ถูกออกแบบมาเพื่อใช้ในลักษณะเดียวกันกับตัวควบคุมแบบอนาลอคทั่วๆไป แต่ตัวควบคุมแบบนี้จะมีสมรรถนะที่สูงกว่า มีความยืดหยุ่นมากกว่า และประกอบไปด้วยฟังค์ชั่นในการคำนวณ และ ในการควบคุมมากมาย ซึ่งในส่วนภาคผนวกนี้จะขอกล่าวถึง คุณลักษณะ ข้อกำหนดต่างๆ ที่เป็นประโยชน์ในการใช้ตัวควบคุมได้เป็นอย่างดี ในหน้าถัดไป

ADDRESS	00	21	ST
	01	64	ST
	02	00	ST
	03	14	ST
	04	22	ST
	05	04	ST
	06	06	ST
	07	06	ST
	08	23	ST
	09	08	ST
	0A	00	ST
	0B	00	ST
	0C	02	ST
	0D	0C	ST
	0E	00	ST
	0F	00	ST
	10	67	ST
	11	9A	ST
	12	9B	ST
	13	9B	ST
	14	01	ST
	15	04	ST
	16	05	ST
	17	00	ST

โปรแกรมการใช้ตัวควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียด (Specification)

สัญญาณอินพุท	ตัวแปรกระบวนการ	1 ถึง 5 โวลต์ ดีซี
	ค่าเป้าหมายจากภายนอก	1 ถึง 5 โวลต์ ดีซี
	พัลส์วิดท์อินพุท	1 ชุด (เพิ่ม และ ลด)
	อนาลอคอินพุทช่วย	1 ถึง 5 โวลต์ดีซี 5 จุด
	ดิจิตอลอินพุท	5 จุด
	ดิจิตอลอินพุทช่วย	5 จุด
	ตัวแปรที่ใช้ในการควบคุม (ที่อ่านค่ากลับมา)	1 จุด
สัญญาณเอาท์พุท	ตัวแปรที่ใช้ในการควบคุม	4 ถึง 20 มิลลิแอมป์ดีซี 1 ถึง 5 โวลต์ดีซี หรือ พัลส์วิดท์เอาท์พุท
	ตัวแปรกระบวนการที่ชดเชยแล้ว	1 ถึง 5 โวลต์ดีซี
	ค่าเป้าหมายเอาท์พุท	1 ถึง 5 โวลต์
	อนาลอคเอาท์พุทช่วย	1 ถึง 5 โวลต์ดีซี 2จุด
	ดิจิตอลเอาท์พุท	8 จุด
	ดิจิตอลเอาท์พุทช่วย	8จุด
ตัวแสดงค่า	ตัวแปรกระบวนการและค่าเป้าหมาย	แสดงด้วยพลาสม่า : ความละเอียด 0.5%
		ความยาวสเกล 100 มม.
การติดตั้ง	สภาวะอุณหภูมิ	0 °ถึง 45 ° C
	แหล่งจ่ายกำลัง	24 โวลต์ดีซี(20-30โวลต์)
	การใช้พลังงาน	ประมาณ 11 วัตต์
	ขนาด	144*72*400 มิลลิเมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การโปรแกรม

ตัวควบคุมนี้ประกอบไปด้วยหน่วยฟังค์ชั่นพื้นฐาน 2 แบบคือ

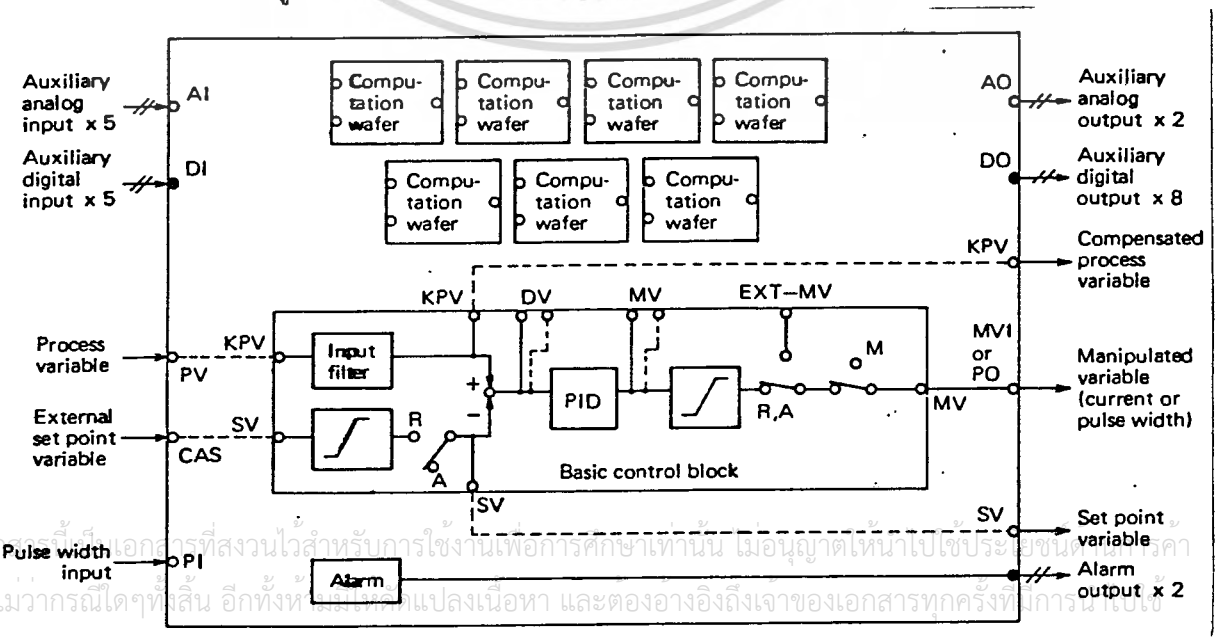
1. บล็อคควบคุม ที่มีแต่ฟังค์ชั่นในการควบคุม
2. บล็อคคำนวณ ที่มีแต่ฟังค์ชั่นในการคำนวณ

ทุกๆฟังค์ชั่นการควบคุมและฟังค์ชั่นการคำนวณ ถูกออกแบบให้เชื่อมต่อกันในลักษณะเป็น ซอฟแวร์แพ็คเกจ ที่ใช้แทนการทำงานของแต่ละฟังค์ชั่นการทำงาน ซึ่งเรียกว่า "เวเฟอร์" (Wafer) โดยที่เราสามารถป้อนค่า หรือ ทำการโปรแกรมการทำงานของตัวควบคุมนี้ ได้ทางหน่วยป้อนข้อมูล (Data entry unit) โดยที่เราไม่จำเป็นต้องมีเครื่องมือใดๆอีก

สำหรับบล็อคการควบคุม จะประกอบไปด้วย บล็อคควบคุมแบบปฐมภูมิ (Primary control block , 1 PID)และ บล็อคควบคุมแบบทุติยภูมิ (Secondary control block, 2 PID) ดังนั้นเราจึงสามารถที่จะรวมบล็อคการควบคุมทั้งสองเข้าด้วยกัน เป็นฟังค์ชั่นการทำงานแบบต่างๆ ดังนี้คือ

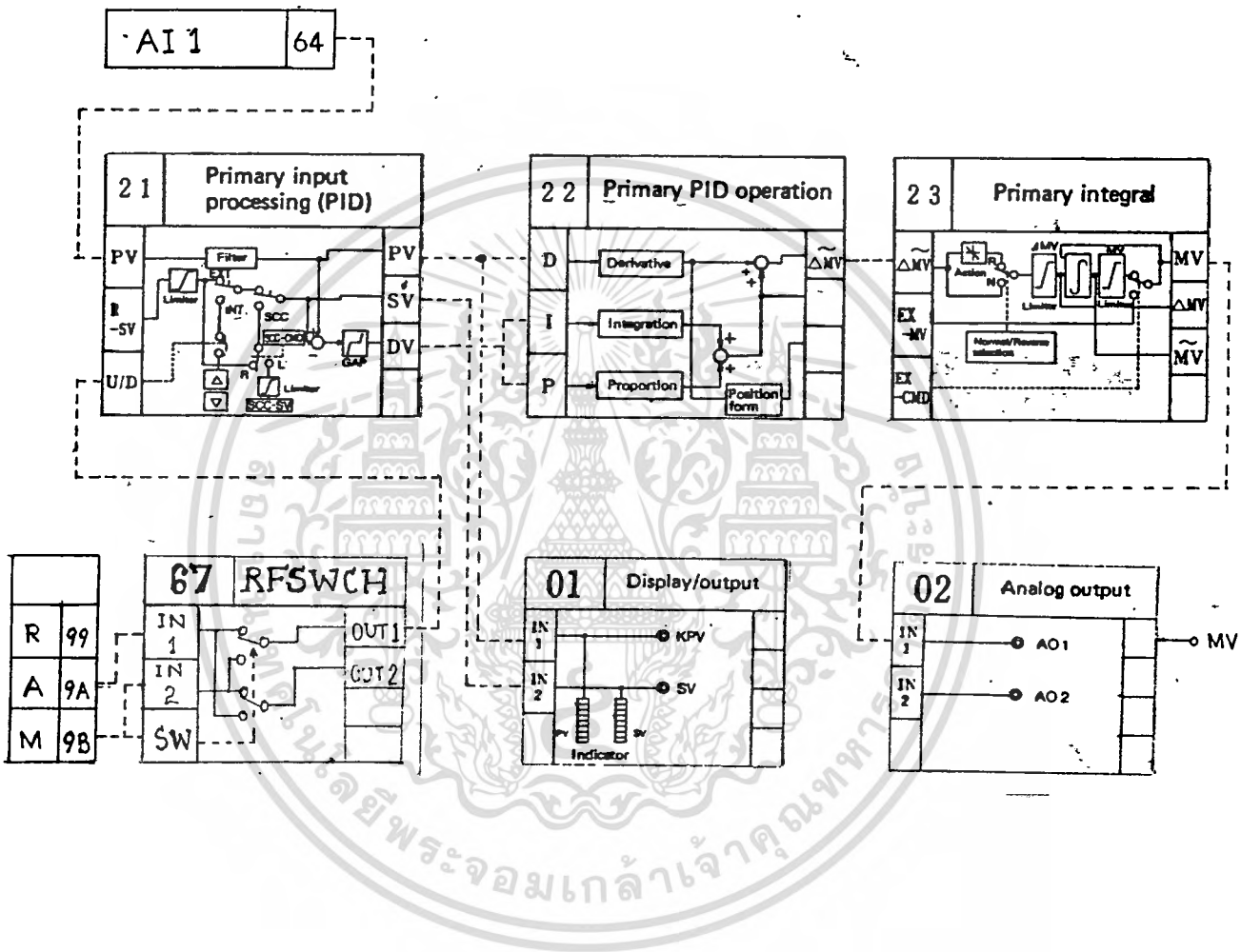
1. ฟังค์ชั่นการควบคุมแบบพื้นฐาน (Basic control function)
2. ฟังค์ชั่นการควบคุมแบบต่อลดหลั่นกัน (Cascade control function, 1PID + 2PID)
3. ฟังค์ชั่นการควบคุมแบบอัตราส่วน (Ratio control function , R + 2PID)
4. ฟังค์ชั่นการควบคุมโปรแกรม (Program control function)

ซึ่งในปริญญานิพนธ์นี้ จะกล่าวถึงฟังค์ชั่นการควบคุมแบบพื้นฐาน ที่ใช้ในการทดลองเท่านั้น ซึ่งจะมีลักษณะการเชื่อมต่อดังรูปที่ 21



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยลักษณะการโปรแกรม หรือ การเชื่อมต่อแต่ละเวเฟอร์. ให้สามารถทำงานตามแผนภาพบล็อคข้างต้น ได้ดังรูปที่ 22



รูปที่ 22

ซึ่งเวเฟอร์ที่ 21 ถึง เวเฟอร์ที่ 23 แทนฟังค์ชั่นการทำงานแบบ การควบคุม PID ปฐมภูมิ

เวเฟอร์ที่ 01 แทนฟังค์ชั่นการแสดงผลเอาท์พุท ในการแสดงค่าเป้าหมาย และ ค่าตัวแปรกระบวนการ

เวเฟอร์ที่ 02 แทนอนาลอคเอาท์พุท เพื่อเป็นพอร์ท (Port) ในการต่อเชื่อมกับวงจรภายนอก โดยเอาท์พุทที่ส่งออกมาจะเป็นค่าตัวแปรที่ใช้ในการควบคุม

เวเฟอร์ที่ 67 แทนสวิทช์เวเฟอร์ ทำให้อินพุท U/D ของเวเฟอร์ที่ 21 มีค่าแอกทีฟ และสามารถปรับค่าเป้าหมายได้จากสวิทช์ค่าเป้าหมายของตัวควบคุมได้โดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

* ผู้เขียนขอขอบพระคุณ อาจารย์สุเธียร เกียรติสุนทร อาจารย์ที่ปรึกษาปริญญานิพนธ์ ที่ได้ช่วยเหลือให้คำแนะนำ ข้อคิดเห็นต่างๆ ตลอดจนกำลังใจในการทำงานอย่างมาก นอกจากนี้ ขอขอบคุณ เพื่อนๆภาควิชา คอมพิวเตอร์ ที่ได้ช่วยเหลือให้คำปรึกษาในด้านโปรแกรม

หากปริญญานิพนธ์นี้พอจะมีประโยชน์อยู่บ้าง ขอมอบความดีนี้ให้ผู้มีพระคุณทั้งหลาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. KATSUHIKO OGATA , "Modern Control Engineering" , Prentice-Hall , 1970

2. F.G. Shinskey , "Process Control System" , McGraw-Hill Book Company ,Inc. 1979

3. Microsoft QuickBasic Compiler 2.0 ,1986

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้