



ปีการศึกษา 2534

ระบบการควบคุมแบบแยกส่วน



อาจารย์ที่ปรึกษา

อาจารย์อภินิษฐา แซ่คัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไปว่ากรณีใดที่ขึ้น ลึกขึ้นห้วงมิให้คัดลอกไปเผยแพร่ และต้องแจ้งถึงเจ้าของเอกสารทุกครั้งที่มีวางไปใช้

ปริญญาโทปีการศึกษา 2533

ภาควิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

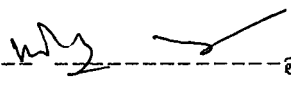
เรื่อง ระบบการควบคุมแบบแยกส่วน (DISTRIBUTED CONTROL SYSTEM)

จัดทำโดย

นางสาวสุวิมล ชรรณกลางกูร รหัส 311377

นางสาวโสภาวดี จันทระประทีน รหัส 311388

นายอดิสรณ์ อธิรานพัฒนา รหัส 311391


-----อาจารย์ที่ปรึกษา

(อาจารย์บัณฑิต แท้ตั้ง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบการควบคุมแบบแยกส่วน

(DISTRIBUTED CONTROL SYSTEM)

นางสาว สุวิมล ธรรมกุลกลางกูร 31.1377

นางสาว รสภาวดี จันทร์ประทีน 31.1388

นาย อภิสิทธิ์ อีรานุกุลนา 31.1391

อาจารย์ที่ปรึกษา อาจารย์ ชนิษฐา แซ่ตั้ง

ปีการศึกษา 2534

บทคัดย่อ

ในภาวะปัจจุบัน การอุตสาหกรรมได้รับการพัฒนาให้ก้าวหน้าไปอย่างรวดเร็ว ดังนั้น ระบบการควบคุมจึงมีความสำคัญมากขึ้น ระบบการควบคุมแบบแยกส่วน (Distributed Control System) ซึ่งเป็นระบบควบคุมที่สามารถใช้ควบคุมหน่วยควบคุมย่อยๆ ซึ่งจะเหมาะสมกับการควบคุมในระยะทางที่ไกล หรือในสภาพแวดล้อมที่ใหม่ เหมาะสมกับการที่สิ่งมีชีวิตจะเข้าไปอยู่ในสภาพนั้นได้ การควบคุมแบบแยกส่วนจะใช้ ไมโครโพรเซสเซอร์ (Microprocessor) เป็นส่วนประมวลผลข้อมูลเพื่อการควบคุม และใช้ส่วนคอมพิวเตอร์ (Computer) เป็นตัวรับคำสั่งจากผู้ควบคุม (User) ทำหน้าที่เป็นตัวกลางในการส่งคำสั่งต่อไปยังส่วนขบวนการ การควบคุมจะให้ผลการทำงานที่มีความละเอียดและแม่นยำ เพราะส่วนที่ถูกควบคุมจะมีการป้อนกลับเพื่อนำมา เปรียบเทียบและชดเชย เพื่อให้ข้อมูลมีความถูกต้องทำให้การทางานของระบบเป็นไปอย่างมีประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DISTRIBUTED CONTROL SYSTEM

MS.SUVIMOL THAMMAKURANGKUL 311377

MS.SOPAWADEE CHANTARAPRATIN 311388

MR.ADISON THEERANUPATTANA 311391

MS. KANITTA SAETANG ADVISOR

ABSTRACT

Recently, Industrial have vastly improved and developed so that control system is considered very important. Distributed Control System is a kind of control used for control many individual controllers that accommodate with remote control of subsistence for humankind. Distributed Control System will allow many microprocessors for control processes and receive command from user by computer. Due to computer, its main function is transmit data to process. Control system must be works with reliability because controlled unit would be allowed for comparison and compensation. So output data will be precisely and process will works with effeiciency.

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานโดยไม่ได้รับอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎี	3
บทที่ 3 การสร้างระบบควบคุมแบบแยกส่วน	36
บทที่ 4 การทดลองและผลการทดลอง	52
บทที่ 5 บทสรุปและวิจารณ์	58

ภาคผนวก

กิตติกรรมประกาศ

หนังสืออ้างอิง

บทที่ 1

บทนำ

ในภาวะปัจจุบันโลกเรากำลังรุกหน้าไปด้วยเทคโนโลยี (Technology) ที่ทันสมัย หน่วยงานอุตสาหกรรมก็ เป็นแหล่งสำคัญ ต่อการพัฒนาความก้าวหน้าของประเทศไทย จะเห็นได้ว่าระบบอุตสาหกรรมทั่วไป จะต้องมีการควบคุมการทำงานทุกขั้นตอนขั้นตอนนี้ เพื่อที่จะ สามารถทราบการทำงานของหน่วยการทำงานย่อยต่างวของระบบ เพื่อลดความผิดพลาด ของผลงานให้น้อยที่สุด

เนื่องจากการที่คอมพิวเตอร์ (Computer) เป็นอุปกรณ์ช่วยในการคำนวณ ประมวลผล และวิเคราะห์ข้อมูลได้อย่างถูกต้องรวดเร็ว ทำให้คอมพิวเตอร์เข้ามามีบทบาท มากขึ้นทุกทีในปัจจุบัน สำหรับหน่วยงานที่มีความซับซ้อน และยุ่งยาก การเลือกใช้คอมพิวเตอร์มีทางกันแทนจึง เป็นความเหมาะสมอย่างยิ่ง

ในการควบคุมการทำงานส่วนต่างวของระบบ จะมีไมโครคอนโทรลเลอร์ (Micro Controller) ทำหน้าที่ในการควบคุมการทำงาน แต่เนื่องจากผู้ควบคุมไม่สามารถติดต่อกับตัวคอนโทรลเลอร์ได้โดยตรง จึงได้นำหน่วยคอมพิวเตอร์มาทำการอินเทอร์เฟส (Interface) กับตัวคอนโทรลเลอร์และระบบ ทำให้ผู้ทำการควบคุมมีความสะดวกในการทำงานมากขึ้น และเพื่อให้ระบบทำงานได้อย่างมีประสิทธิภาพมีค่าผิดพลาดน้อยที่สุดเท่าที่จะเป็นไปได้ จากสิ่งเหล่านี้ทำให้คณะผู้จัดทำมีแนวความคิดดังนี้คือ

1. ใช้คอมพิวเตอร์เป็นตัวกำหนดความเร็ว และประมวลผลเพื่อแก้ไขความผิดพลาดที่เกิดจากการทำงานของดีซีเอ็มเอเตอร์
2. ใช้ไมโครคอนโทรลเลอร์ เบอร์ 8031 ช่วยในการควบคุมดีซีเอ็มเอเตอร์และติดต่อกับคอมพิวเตอร์

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของคณะผู้จัดทำ หากมีการนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตจากคณะผู้จัดทำ ถือว่าผิดกฎหมาย และต้องรับผิดชอบต่อเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เพื่อให้เกิดประโยชน์สูงสุด จะสามารถใช้คอมพิวเตอร์ร่วมกับตัวคอนโทรลเลอร์หลายตัวได้ จะสามารถติดต่อกับทุกหน่วยโดยใช้การแบ่ง เวลา

ที่กล่าวมาทั้งหมดนี้เป็นแนวคิดหลักในการหาโครงการนี้ ภายใต้โครงการที่ชื่อว่า "ระบบการควบคุมแบบแยกส่วน" และมีชื่อเป็นภาษาอังกฤษว่า "DISTRIBUTED CONTROL SYSTEM" คณะผู้จัดทำได้พยายามที่พัฒนาระบบให้ใช้งานได้สะดวก และมีเสถียรภาพมากที่สุด และ เนื่องจากการติดต่อกับตัวคอนโทรลเลอร์ที่มีจำนวนหลายตัว จะต้องมีการกำหนดรหัสของแต่ละตัวไว้ด้วย

คณะผู้จัดทำหวัง เป็นอย่างยิ่งว่า โครงการนี้จะสามารถก่อให้เกิดประโยชน์ต่อการใช้งานมากที่สุดเท่าที่จะเป็นได้



บทที่ 2

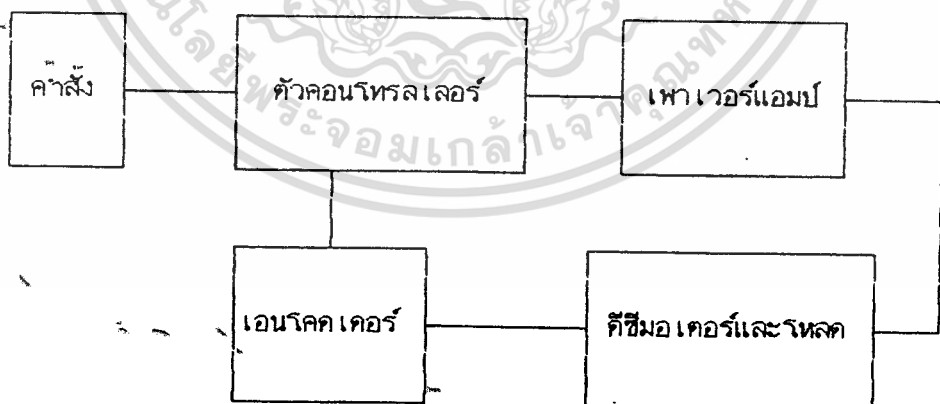
ทฤษฎี

ทฤษฎีพื้นฐานที่จะถูกนำมาใช้ในการอธิบาย การทำงานของโครงงานนี้ สามารถแยกอธิบายออกเป็นเรื่องต่างๆ ได้ดังต่อไปนี้

2.1 ส่วนประกอบของการควบคุมมอเตอร์กระแสตรง

การควบคุมมอเตอร์กระแสตรง ประกอบด้วยส่วนประกอบที่สำคัญ 4 ส่วน คือ

1. ตัวควบคุม (Controller)
2. วงจรขับ (Driver) หรือ วงจรขยายกำลัง (Power Amplifier)
3. ฟีดแบ็คทรานสดิวเซอร์ หรือ เอนโคคเตอร์ (Transducer or Encoder)
4. มอเตอร์กระแสตรงและโหลด



รูปที่ 2.1 ระบายการควบคุมมอเตอร์กระแสตรงแบบพื้นฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 การสื่อสารแบบอนุกรม

การสื่อสารในคอมพิวเตอร์จะอาศัยอยู่ในรูปของโวลท์หรืออนกิลส์ส การที่จะส่งข้อมูลแบบขนานจะต้องส่งข้อมูลไปพร้อมกันทั้ง 8 บิต ต้องใช้สายในการคิดต่อกถึง 9 เส้น (เป็นสัญญาณอ้างอิง 1 เส้น) และต้องใช้วงจรแปลงระดับสัญญาณเพื่ทำให้ถึง 8 วงจรทำให้เกิดความสิ้นเปลืองไม่เหมาะที่จะนำมาใช้งานในระยะทางไกล ส่วนการสื่อสารแบบอนุกรมจะใช้การทยอยส่งทีละบิต แบ่งออกเป็น 2 รูปแบบคือการส่งแบบอะซิงโครนัส (Asynchronous) และแบบซิงโครนัส (Synchronous)

การส่งแบบอะซิงโครนัส จะเป็นการส่งทีละอักขระไม่จำเป็นต้องต่อเนื่อง และช่วงเวลาาระยะอักขระไม่กำหนดแน่นอน แต่สัญญาณที่ส่งออกไปจะเป็นตัวบอกคือ มีบิตเริ่ม (Start Bit) และบิตหยุด (Stop Bit) เพื่อบอกให้ฝ่ายรับทราบขอบเขตของการแยกอักขระได้ มาตรฐานของสมาคมอุตสาหกรรมอิเล็กทรอนิกส์ (EIA) กำหนดมาตรฐาน RS 232C ไว้ดังนี้คือ

ข้อมูลที่เป็น 1 ให้แทนด้วย -15 V.

ข้อมูลที่เป็น 0 ให้แทนด้วย +15 V.

บิตเริ่ม (Start Bit) เริ่มต้นเมื่อสัญญาณเปลี่ยนจาก 0 (+15 V.) ไปค่า (-15 V.)

บิตหยุด (Stop Bit) จะมีสัญญาณแบบ 0 (+15 V.) เป็นระยะเวลา 1 สัญญาณนาฬิกา หรือ 1.5 สัญญาณนาฬิกา

เพราะฉะนั้น จึงต้องมีการกำหนดพารามิเตอร์สำหรับการสื่อสาร ซึ่งประกอบด้วย

1. ความเร็วของการส่ง (Baud Rate)
2. ขนาดของข้อมูลที่รวมเป็น 1 อักขระ (Data)
3. ชนิดของการตรวจสอบ (Parity)

4. ความยาวของบิตหยุด (Stop Bit)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่งแบบชิงโครนัส จะไม่มีทั้งบทเริ่มและบทหยุด อักษรทุกอักขระ
ที่ส่งจะต้องส่งมาอย่างต่อเนื่อง จนครบตามที่ตกลงกันไว้ การกำหนดขอบเขตของอักขระ
ขึ้นกับส่วนปรกติคอล (Protocol)

ไอซี เบอร์ 8250จะเป็นคำทหาหน้าส่งโดยเปลี่ยนระบบการสื่อสาร
จากแบบขนาน ไปเป็นแบบอนุกรม และทหาหน้ารับข้อมูลโดยเปลี่ยนระบบการสื่อสารจาก
แบบอนุกรมไปเป็นแบบขนาน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 การโปรแกรม 8250

ช่องการสื่อสาร	รีจิสเตอร์	การใช้งาน
XF8	RBR, DLL	-ใช้สำหรับที่หักข้อมูลทั้งรับทั้งส่ง -ใช้ เป็นตัวหารความถี่ที่มีค่าน้อย
XF9	DLM, IER	-ใช้ เป็นตัวหารความถี่ที่มีค่าสูง -ใช้สำหรับควบคุมการอินเทอร์รัพท์
XFA	IIR	-ใช้สำหรับบอกชนิดของการอินเทอร์รัพท์
XFB	LCR	-ใช้สำหรับควบคุมพารามิเตอร์การรับส่ง -สำหรับการบ่งบอกการใช้งานของ DLL และ DLM
XFC	MCR	-ใช้สำหรับควบคุมรีโม้เต็ม
XFD	LSR	-ใช้สำหรับบอกสถานะภาพของการรับส่ง
XFE	MSR	-ใช้สำหรับบอกสถานะภาพของรีโม้เต็ม

ตารางที่ 2.1 แสดงช่องทางเข้าออกและการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ของรีจิสเตอร์ 8250 ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BAUD	ตัวหาร	
	ฐาน 10	ฐาน 16
300	384	180
600	192	0C0
1200	96	060
2400	48	030
4800	24	018
9600	12	00C

ตารางที่ 2.2 แสดงตัวหาร 1.8243 MHz. เพื่อห้

ได้ความถี่ 16 เท่าของ BAUD RATE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิทที่ 0	สร้างสัญญาณอินเตอร์รัพท์เมื่อรับข้อมูลได้ครบ 1 อักขระ
บิทที่ 1	สร้างสัญญาณอินเตอร์รัพท์เมื่อรีจิสเตอร์สำหรับส่งว่างลง
บิทที่ 2	สร้างสัญญาณอินเตอร์รัพท์เมื่อมีข้อผิดพลาดเกิดขึ้นจากการรับ
บิทที่ 3	สร้างสัญญาณอินเตอร์รัพท์เมื่อมีสัญญาณ Modem Status

ตารางที่ 2.3 แสดงการคุมการกำเนิดสัญญาณอินเตอร์รัพท์
โดยการเขียนข้อมูลเข้าในช่องทางเข้าออกหมายเลข XF9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

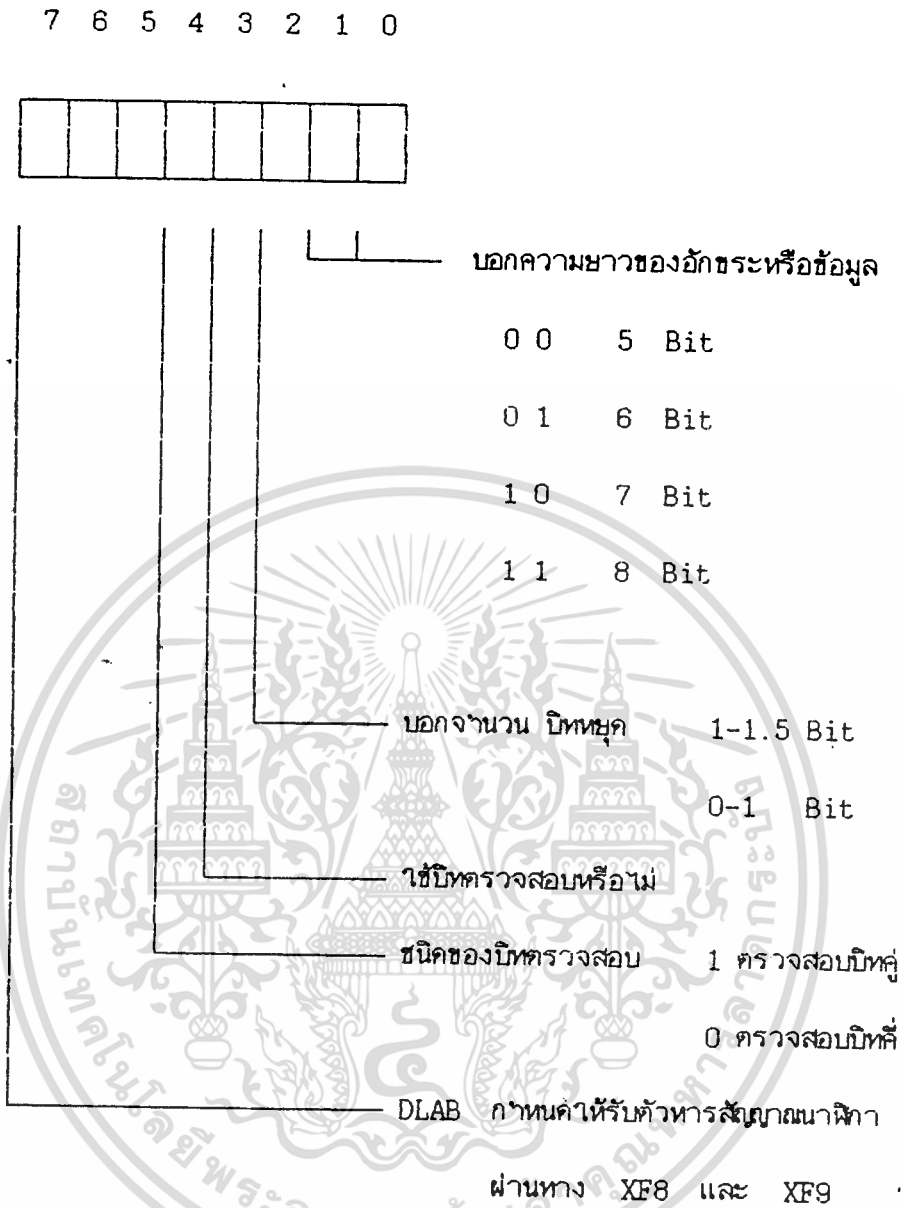


ลำดับ ความสำคัญ	คำที่อ่าน ไว้ด้าน IIR	สาเหตุของ การอินเทอร์รัพท์	ขบวนการทำให้สัญญาณ อินเทอร์รัพท์หาย
1	0000110	-เกิดการเขียนทับข้อมูลเก่า -ตรวจสอบความผิดพลาด ของช่วงอักขระ	อ่านรีจิสเตอร์ LSR ที่หมายเลขช่องทาง XFD
2	0000100	-เมื่อรับข้อมูลครบอักขระ	อ่านข้อมูลจาก RBR ที่หมายเลขช่องทาง XF8
3	0000010	-เมื่อส่งข้อมูลออกไปหมด	เขียนข้อมูลลง RBR

ตารางที่ 2.4 แสดงรีจิสเตอร์ IIR แสดงชนิดของ

อินเทอร์รัพท์ที่เกิดขึ้น

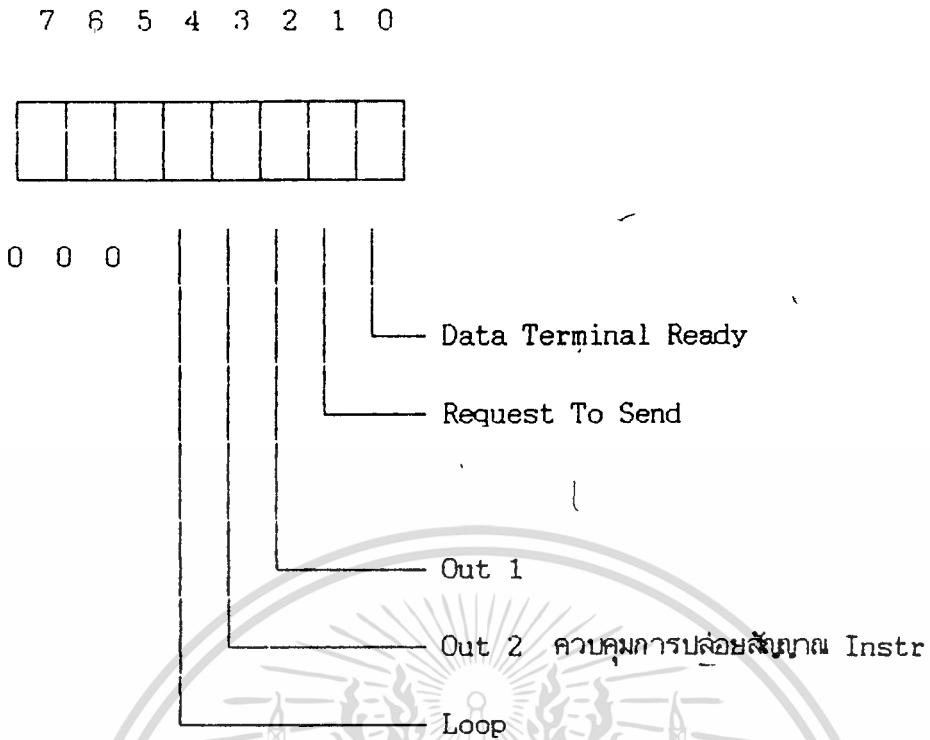
008494



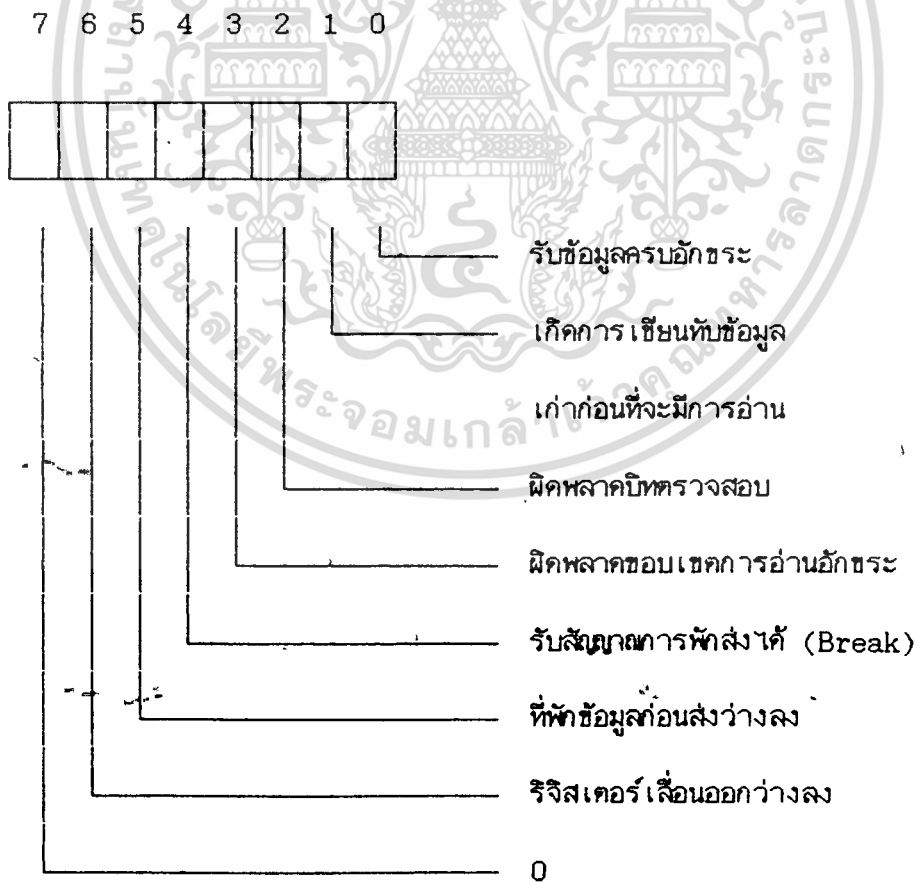
รูปที่ 2.2 การตั้งค่าพารามิเตอร์ของการสื่อสาร

ที่หมายเลขช่องทางเข้าออก XF8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 การตั้งค่าควบคุมเพิ่มเติม XFC

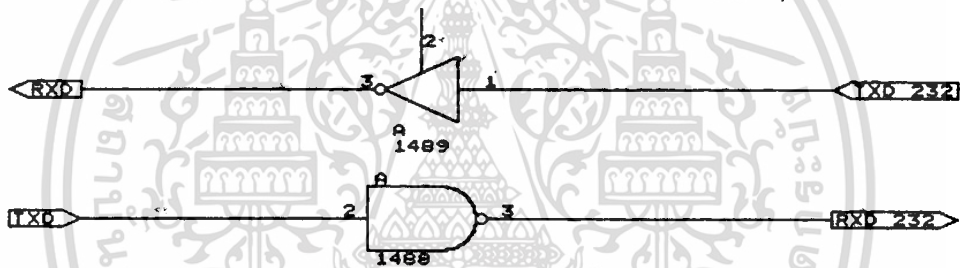


รูปที่ 2.4 สถานภาพของการรับส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า (หมายเหตุช่องทางเข้าออกที่ XFD)
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามเผยแพร่ลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 การอินเทอร์เฟสกับ 8031

รูปแบบของข้อมูลที่ 8031ใช้ในการติดต่อกับรับส่งข้อมูลทางพอร์ตอนุกรม (Serial Port) อยู่ในโหมด 8 BIT UART ซึ่งมี 8 บิตข้อมูล (Data Bit), 1 บิตเริ่ม (Start Bit) และ 1 บิตหยุด (Stop Bit) และไม่มีพาริตีบิต (Parity Bit) พอร์ตอนุกรมของ 8031 เป็นสัญญาณแบบทีแอล(TTL) ไม่สามารถต่อเข้าโดยตรงกับพอร์ตอนุกรมมาตรฐาน RS-232C จะต้องมีวงจรมินิอินเทอร์เฟส (Interface) เพื่อแปลงสัญญาณ TTL เป็นสัญญาณ EIA ตามมาตรฐาน RS-232C ดังแสดงในรูป



รูปที่ 2.5 แสดงวงจรมินิอินเทอร์เฟสสำหรับ

แปลงสัญญาณ TTL เป็น EIA และ EIA เป็น TTL

2.5 ชิปเกิลซีพ ไมโครคอนโทรลเลอร์ (Single Chip Microcontroller)

ชิปเกิลซีพ เบอร์ 8031 เป็นไมโครคอนโทรลเลอร์ในตระกูล MCS-51ซึ่งไม่มีหน่วยความจำชนิด รอม อยู่บน ชิป แต่มีลักษณะเด่น ดังนี้

1. ใช้เอชเอ็มอส (HMOS) เทคโนโลยีและทำงานได้ด้วยแหล่งจ่ายไฟขนาด 5 V.
2. มีหน่วยประมวลผลกลาง ขนาด 8 บิต
3. มีวงจรรอสซิลเลเตอร์บนชิป
4. มีชุดรีจิสเตอร์ (Register) 4 ชุด (Bank)
5. มีตัวตั้งเวลา และตัวนับ (Timer/ Counter) ขนาด 16 บิต 2 ตัว
6. มีพอร์ตขนาน อินพุต/เอาต์พุต แบบ 2 ทิศทาง (Bidirectional I/O Port) 4 พอร์ต
7. มีพอร์ตการสื่อสารแบบอนุกรม ที่สามารถโปรแกรมการรับส่งแบบ 2 ทิศทางสมบูรณ์ (Full Duplex) ที่มีความเร็วในการสื่อสารสูง
8. มีซอฟต์แวร์แฟลก (Software Flag) สำหรับผู้ใช้ที่สามารถกำหนดเองได้ ถึง 128 คาบแทนบิต
9. สามารถใช้งานแบบบูลีน (Boolean) ได้
10. มีคำสั่งกวาดค่าทางคณิตศาสตร์ และสามารถใช้ได้ทั้งตัวเลขแบบฐาน 10 และฐาน 2
11. การกำหนดพื้นที่สแตค (Stack) สำหรับโปรแกรมย่อยต่างๆ ทำได้กว้าง

โครงสร้างทั่วไปของไมโครคอนโทรลเลอร์ MCS-51 มีดังนี้

(1) โครงสร้างภายนอกของ 8031

จากรูปที่ 2.6 เป็นการจัดขาตามลักษณะภายนอกของ 8031ซึ่งมีรายละเอียดดังนี้

- ขา V_{SS} (ขา 20) เป็นขาสำหรับต่อลงกราวด์
- ขา V_{CC} (ขา 40) เป็นขาสำหรับต่อกับไฟเลี้ยง (Supply)
- พอร์ต 0 (ขา 32 ถึง 39) เป็น อินพุท/เอาต์พุท 8 บิต แบบ 2 ทิศทาง ที่สามารถขับโหลด TTL ได้สูงถึง 8 ตัว และ พอร์ตนี้ ยังเป็น การรวมสัญญาณระหว่างแอดเดรสเบ็ดเตล็ด กับ ข้อมูลในการอ้างถึงหน่วยความจำภายนอกด้วย
- พอร์ต 1(ขา 1-8) เป็นพอร์ต อินพุท/เอาต์พุท 8 บิต แบบ 2 ทิศทาง ที่สามารถ ขับโหลด TTL ได้สูงสุด 4 ตัว
- พอร์ต 2 (ขา 21-28)เป็นพอร์ต อินพุท/เอาต์พุท 8 บิต แบบ 2 ทิศทาง ที่สามารถขับ โหลด TTL ได้สูงสุด 4 ตัว พอร์ตนี้จะถูกใช้งานเป็นตัวส่งแอดเดรสเบ็ดเตล็ดสูงด้วย เมื่อใช้งานร่วมกับหน่วยความจำภายนอก เพื่อจะได้อ้างแอดเดรส ได้ถึง 16 บิต
- พอร์ต 3 (ขา 10-17) เป็นพอร์ต อินพุท/เอาต์พุท 8 บิต แบบ 2 ทิศทาง สามารถ ขับโหลด TTL ได้ถึง 4 ตัว และยังสามารถใช้งานตามฟังก์ชันพิเศษได้อีก
- ขา RST (ขา 9) ต้องคงสถานะสูง เป็นเวลาอย่างน้อย 2 ไมโครวินาที เมชขึ้น ขณะที่ต้องการรีเซ็ตทั้งระบบงาน โดยจะต่อตัวความต้านทานพูลดาวน์ (Pull Down) จากขารีเซ็ตลงกราวด์ และเพื่อที่ตัวชิพ สามารถรีเซ็ตตัวเองได้ โดยอัตโนมัติ ขณะเปิดไฟจะใช้ตัวเก็บประจุต่อคร่อมระหว่างขา RST กับ ขาไฟเลี้ยง
- ขา ALE/PROG (ขา 30) เป็นขาแอดเดรส แลทช์ อีเนเบิล (Address Latch Enable) ใช้ สำหรับแลทช์ค่าแอดเดรสเบ็ดเตล็ดจาก พอร์ต 0 ในระ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้เฉพาะเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หว่างการเข้าถึงหน่วย ความจำภายนอก.

- ขา PSEN (ขา 29) เป็นสัญญาณสโตรบ (Strobe) การอ่านข้อมูลจาก ปรแกรม หน่วยความจำภายนอก ขา PSEN จะไม่มีพัลส์ ส่งออก เมื่อ ซีพียู ทำงาน ด้วยปรแกรมหน่วยความจำภายใน
- ขา EA/V_{pp} (ขา 31) ถ้าขานี้มีสถานะสูง หน่วยประมวลผลกลางจะทำงานตามปรแกรม ที่อยู่ในหน่วยความจำภายใน แต่ถ้าขานี้มีสถานะต่ำหน่วย ประมวลผลกลางก็จะทำงานตามปรแกรม ที่อยู่ในหน่วยความจำภายนอก
- ขา XTAL1 (ขา 19) ใช้เป็นขาอินพุต เข้าสู่ตัวออสซิลเลเตอร์ ในซีพียู

(2) การจัดการทางสถาปัตยกรรม

จากรูปที่ 2.7 เป็นบล็อก ที่ถูกแบ่งตามลักษณะงานในการจัดการภายในของ MCS-51 โดยซึ่งเกิลซีพียู แต่ละตัวของตระกูลนี้จะประกอบด้วยหน่วยประมวลผลกลาง (CPU) หน่วยความจำ, พอร์ทอินพุต, พอร์ทเอาต์พุต, รีจิสเตอร์สถานะและข้อมูล และ ส่วนวงจรตรรก ในการส่งที่จำเป็นสำหรับตัวแปรของฟังก์ชัน การติดต่อท่วงส่วนต่างที่กล่าวมานี้ จะติดต่อกันด้วยบัสข้อมูลขนาด 8บิต และ จะมีบัฟเฟอร์ สำหรับการติดต่อกับข้อมูลกับภายนอกผ่าน พอร์ทอินพุต/เอาต์พุต

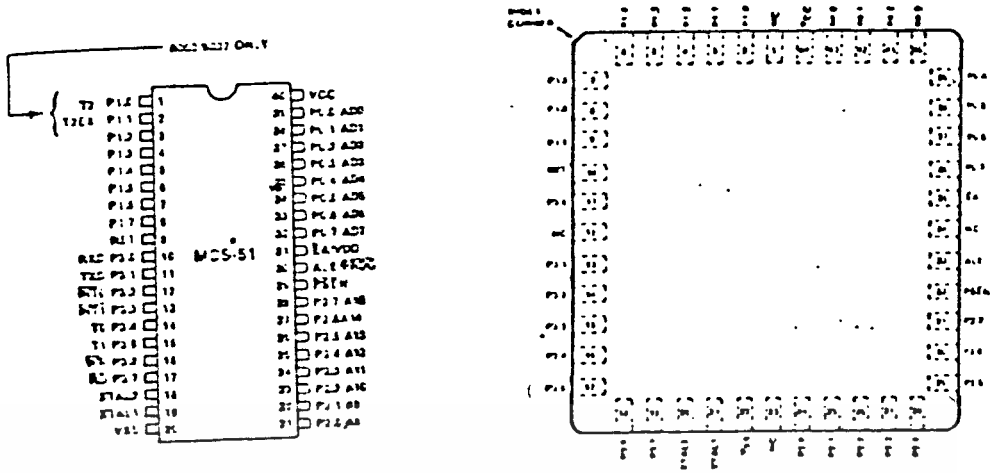
2.6 โครงสร้างทวงฮาร์ดแวร์ ของหน่วยประมวลผลกลาง

กับหน่วยความจำภายนอก

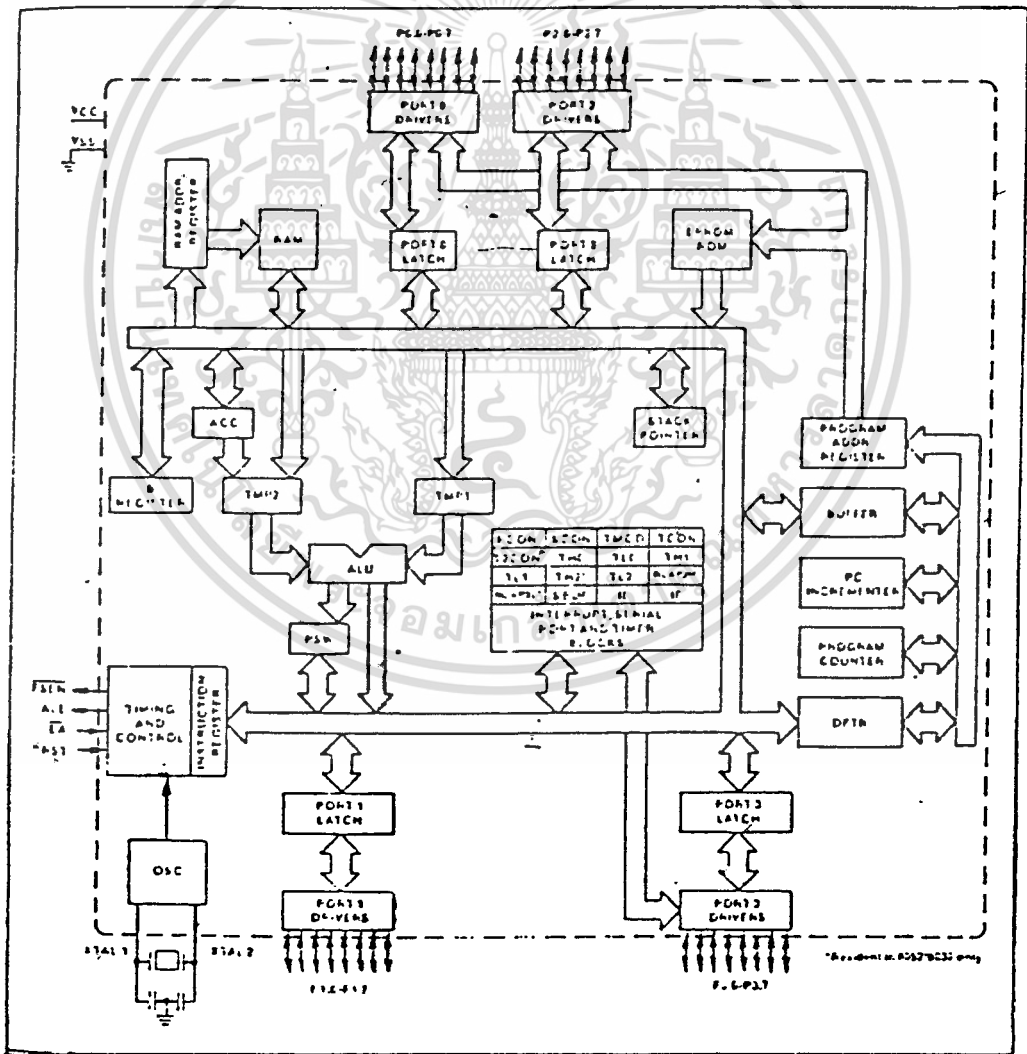
เนื่องจากไมโครคอนโทรลเลอร์ เบอร์ 8031 มีแรมภายใน เพียง 128 ไบท์ และไม่มีรอมภายใน ดังนั้น ในการนำมาใช้งานจึงต้องต่อหน่วยความจำเพิ่มเติม โดยมี วงจรดังรูปที่ 2.8

จากรูปวงจร จะเห็นว่า แบ่งเป็น 3 ส่วนคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับส่วนวงจรสัญญาณนาฬิกาของหน่วยประมวลผลกลางไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.6 การจลัษายภายนอกของ 8031



รูป 2.7 สถาปัตยกรรมของ 8031

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ส่วนวงจรรีเซ็ต ของหน่วยประมวลผลกลาง

3. ส่วนวงจรที่หน่วยประมวลผลกลาง ต่อกับหน่วยความจำภายนอก

ส่วนวงจรสัญญาณนาฬิกาของ หน่วยประมวลผลกลาง

จากรูปวงจร จะเห็นว่าวงจรส่วนนี้จะประกอบด้วยตัวเก็บประจุ C_1 และ C_2 ซึ่งมีค่า 30 pF. และคริสตอล ขนาด .11.0592 MHz. ซึ่งใช้เป็นตัวสร้างสัญญาณนาฬิกา และใน 1 แมกซีน ไซเคิล จะประกอบด้วย 6 สเตท และ ในแต่ละสเตท จะประกอบด้วย 2 เฟส แต่ละเฟสจะใช้เวลาเท่ากับ 1 สัญญาณนาฬิกา และใน 1 แมกซีน ไซเคิล จะมีการส่งสัญญาณ ALE ออกมา 2 ลูก เพื่อนำมาเข้าในการอีเนเบิล IC เบอร์ 74LS373 ซึ่งเป็นตัวแลตช์ ค่าแอดเดรสแบบที่ค่าไว้

ส่วนวงจรรีเซ็ต ของหน่วยประมวลผลกลาง

การรีเซ็ต 8031 จะต้องให้สัญญาณอินพุท ที่เข้ามาที่ขารีเซ็ต คงสถานะสูงเป็นเวลา 2 แมกซีน ไซเคิล ในที่นี้เราใช้วงจร RC เป็นวงจรส่วนรีเซ็ต ซึ่งวงจรนี้ประกอบด้วย ตัวเก็บประจุ C_3 ซึ่งมีค่า 10 μ F. และตัวความต้านทาน R_4 และ R_5 ซึ่งมีค่า 8.2 กิโลโห์ม และ 5.1 กิโลโห์ม ตามลำดับ

การทำงานของวงจร RC นี้ก็คือ วงจรนี้จะจ่ายแรงดันไฟเลี้ยง (V_{CC}) ให้แก่ขา รีเซ็ต จนกว่าตัวเก็บประจุ จะเก็บประจุ (Charge) จนมีแรงดัน 2.5 V. ซึ่งที่ค่าแรงดันนี้ ขารีเซ็ต จะมีสัญญาณอินพุท เข้ามาเป็นลอจิก 0 และ ระบบจะเริ่มทำงาน

สวิตช์รีเซ็ต คือเอาไว้เพื่อให้ผู้ใช้สามารถรีเซ็ต ระบบได้ตามต้องการ

ส่วนวงจรที่หน่วยประมวลผลกลาง ต่อกับหน่วยความจำภายนอก

ในโครงงานนี้ เราใช้อีพรอม เบอร์ 2764 และ RAM 6264 ซึ่งมีหน่วยความจำเท่ากับ 8 กิโลไบต์ และใช้แอดเดรสบัส 13 เส้น คือ A0-A12

เอกสารนี้เป็นเนื่องจากพอร์ต 0 เป็นการรวมสัญญาณระหว่าง แอดเดรสบัส (A0-A7) กับข้อมูลบัส (D0-D7) ไม่ว่าการณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มูล ดังนั้นการติดต่อของ หน่วยประมวลผลกลาง กับ หน่วยความจำภายนอกทำได้โดย
 ส่งค่าผ่าน พอร์ต 0 ออกไปเลขที่ 1677 IC เบอร์ 74LS373 ก่อน เมื่อ CPU ต้องการส่ง
 แอคเครส ก็จะมีการส่งสัญญาณ ALE เป็น ลอจิก 1 ออกไปอีเนเบิล 1677 74LS373 ทำ
 งาน จึงจะมีการส่งแอกเครส ผ่านไปได้ ส่วนพอร์ท 2 1677 เป็นแอกเครสไบต์สูง (A8-
 A15) 1677 เลข เพราะไม่มีการรวมสัญญาณระหว่างแอกเครส กับข้อมูล

จากรูปวงจร จะเห็นว่า เมื่อหน่วยประมวลผลกลาง ต้องการจะติดต่อกับหน่วย
 ความจำส่วนที่เป็นอีพรม จะต้องส่งสัญญาณ PSEN ไปยังขา OE ของอีพรม และ เมื่อ
 ต้องการติดต่อกับหน่วยความจำส่วนที่เป็นแรม ก็ต้องส่งสัญญาณ RD (Read) หรือ WR
 (Write) ไปยังขา OE ของแรม เนื่องจากสัญญาณ PSEN และ RD หรือ WR จะไม่ทำ
 งาน พร้อมกัน ดังนั้น เราจึงใช้ แอกเครส ของอีพรม และ แรม ได้ตั้งแต่ 0000H ถึง
 2000H

2.7 โครงสร้างทางซอฟต์แวร์ ของ 8031

เป็นส่วนประกอบที่สำคัญอีกส่วนหนึ่งของโครงงานนี้ การทำงานของส่วนซอฟต์แวร์
 เริ่มโดยการอ่านไบรแกรมซึ่งถูกแปลง เป็นซอร์สโค้ด (Source Code) เก็บไว้ในอีพรม
 แล้วจึงส่งค่าเข้ามาในหน่วยประมวลผลกลาง 8031 และทำการประมวลผลตามไบร
 แกรมที่อ่านเข้ามา แล้วส่ง ผลไปแสดงผ่านทางส่วนฮาร์ดแวร์ โดยมีไบรแกรม เคาน์
 เตอร์ เป็นตัวแสดงว่าต้องอ่านไบรแกรมที่ตำแหน่งที่เท่าไร ความเร็วในการทำงานของ
 ส่วนซอฟต์แวร์ ขึ้นอยู่กับสัญญาณนาฬิกาที่ส่งให้ หน่วยประมวลผลกลาง ทำงาน ดังนั้น จะ
 เห็นว่าในการทำงานที่จะหาความเข้าใจถึงการออกแบบ และการทำงานของระบบควบคุม
 คุมนี้ ต้องเข้าใจถึงส่วนประกอบทางฮาร์ดแวร์ และขั้นตอนการทำงานของส่วน ซอฟต์แวร์
 1677 ละ เอียค เลียคก่อน ซึ่งรายละเอียดของส่วนซอฟต์แวร์ มีดังนี้

1. โปรแกรมควบคุมการทำงานของระบบควบคุม

ในการทำงานของโปรแกรมส่วนนี้ จะเป็นการทำงานของการจัดระบบต่างๆ ตั้งแต่ เซ็ทค่ารีจิสเตอร์ควบคุม (Control Register) หรือ สเปเชียล ฟังก์ชัน รีจิสเตอร์ (Special Function Register) รับค่าพารามิเตอร์ในโหมดควบคุม กำหนดขั้นตอนการทำงานการรับข้อมูลผ่านทาง ADC การส่งข้อมูลที่ประมวลผลได้ ผ่านทาง DAC การรับค่าข้อมูลจากผู้ใช้ผ่านทางส่วนคอมพิวเตอร์ การจัดส่วนหน่วยความจำข้อมูลภายนอก (RAM เบอร์ 6264) ในการเก็บข้อมูลต่างๆ ในการเขียนโปรแกรมส่วนนี้ จะต้องทำความเข้าใจถึงขั้นตอนการทำงานของวงจรส่วนฮาร์ดแวร์ ให้ละเอียดว่ามีการทำงานอย่างไรในระหว่างการทำงานของโปรแกรม

2. โปรแกรมทางคณิตศาสตร์

ในการทำงานของโปรแกรมส่วนนี้ จะเป็นการทำงานโดยนำข้อมูลที่มีอยู่มาประมวลผล ซึ่งได้แก่การบวก ลบ คูณ หาร ซึ่งขั้นตอนการทำงานในการประมวลผลจะมีส่วนสำคัญมาก เพราะ ถ้าโปรแกรมที่เขียนนั้น ทำงานได้รวดเร็วก็จะส่งผลให้การทำงานของระบบควบคุมเร็วขึ้นได้

3. โปรแกรมฟังก์ชันการควบคุมแบบพีไอดี (PID)

โปรแกรมส่วนนี้มีหลักการทางาน คือ รับสัญญาณอนาล็อก จากกระบวนการมาประมวลผลด้วย สมการดิครีท (Discrete) เพื่อที่ 8031 ประมวลผลได้ แล้วจึงแปลงเป็นสัญญาณอนาล็อก เป็นเอาต์พุต ของตัวควบคุม

2.8 ลิเนียร์ แอมพลิฟายเออร์ (Linear Amplifier)

ลิเนียร์ แอมพลิฟายเออร์ สร้างขึ้นเพื่อใช้เป็นอุปกรณ์ขยายแบบลิเนียร์ เช่น เกี่ยวกันกับออปแอมป์ หรือ ดิฟเฟอเรนเชียลแอมป์ (Differential Amplifier) ที่ใช้ป้อนระดับภาคกำลัง ให้กับมอเตอร์ ภาคกำลัง จะมีการป้อนกลับแบบลูปปิด (Closed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับคนทำงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
Loop) เพื่อให้พารามิเตอร์ที่ต้องการ ได้แก่ แรงดัน กระแสของมอเตอร์ มีความเป็น
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลิเนียร์ ลิเนียร์แอมพลิไฟร์จะเป็นตัวควบคุมแรงดัน หรือ กระแสของมอดูเลเตอร์โดยการบังคับขนาดแรงดัน ที่จ่ายให้กับมอดูเลเตอร์ ดังนั้นมันจะเป็นตัวที่ลดแรงดันเอาไว้ ซึ่งเท่ากับผลต่างของแหล่งจ่ายแรงดัน และ แรงดันที่มอดูเลเตอร์ใช้งานจริง ในขณะที่มีกระแสที่เหมาะสมไหลผ่าน กำลังงานจะสูญเสียใน เอาท์พุททรานซิสเตอร์ เป็นจำนวนมากโดยเฉพาะ ขณะมอดูเลเตอร์วิ่งที่ความเร็วค่าแรงบิดสูง เพราะขณะนั้นโวลต์เตจย้อนกลับของมอดูเลเตอร์ค่ากระแสจึงสูง นี่จึงเป็นความแตกต่างที่เห็นได้ชัดเมื่อเทียบกับสวิตชิ่งแอมพลิไฟร์

ลิเนียร์แอมพลิไฟร์ นิยมใช้ในระบบที่มีแบนด์วิดท์กว้าง และกำลังงานต่ำ (น้อยกว่า 200-300 W.) และมักจะใช้สำหรับนาบับเคลื่อนขดลวดหมุนของมอดูเลเตอร์ ที่มีความเฉื่อยต่ำทำให้เพิ่มกระแสด้วยอัตราเร่งสูงในช่วงเวลาอันสั้น (2-3 ms.) ซึ่งกระแสที่กระทำบนคาว์ของ เอาท์พุททรานซิสเตอร์อาจจะยังทำงานได้ดี

ลิเนียร์แอมพลิไฟร์ อาจจะตามหน้าที่ได้ 2 ชนิด

1. แอมพลิไฟร์ แบบทิศทางเดียว (Unidirectional Amplifier)
2. แอมพลิไฟร์ แบบ 2 ทิศทาง (Bidirectional Amplifier)

หรืออาจจะแบ่งตามโครงสร้างได้เป็นแบบไบโพลาร์ (Bipolar) และ บริดจ์ (Bridge) หรือ อาจจะแบ่งตามสมการของทรานส์เฟอ์ ฟังก์ชัน ได้เป็น แรงดัน และกระแส แอมพลิไฟร์

1. แอมพลิไฟร์ แบบทิศทางเดียว จะใช้ในระบบที่มอดูเลเตอร์ถูกขับให้หมุนได้ในทิศทางเดียว จะไม่สามารถให้แรงดัน หรือ กระแสไหลกลับทิศทางได้ จะแสดงคุณสมบัติการทำงานโดยสามารถเพิ่มอัตราเร่งได้อย่างรวดเร็ว แต่สามารถลดความเร็วได้อย่างเชื่องช้า แต่อย่างไรก็ตามเราสามารถแก้ไขข้อจำกัดบางส่วนนี้ได้โดย เพิ่มส่วนไดนามิค เบรค (Dynamic Break) เพื่อห้ามมอดูเลเตอร์ยังคงเป็นทิศทางเดียวแต่สามารถกลับทิศทางการทำงาน

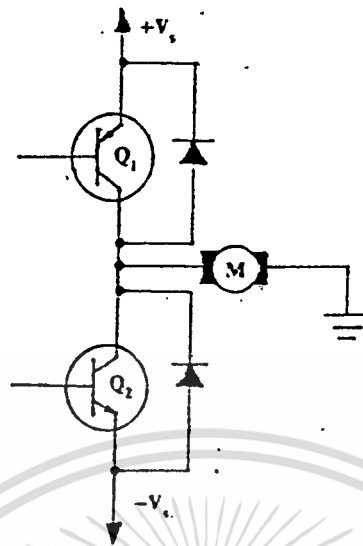
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับจอร์จใช้งานเพื่อการศึกษานั้น ไม่นอนเอาตให้ไปใช้ประโยชน์ด้านการค้า
 1. หลีกเลี่ยงการกระทำที่ผิดกฎหมาย โดยเฉพาะการละเมิดลิขสิทธิ์ของผู้อื่น และให้ทางเดินกระแส
 2. ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอสซิง เกิดจากแรงดันย้อนกลับของมอเตอร์ (Back EMF) ในทิศทางตรงกันข้าม

2. แอมบลิเอร์แบบ 2 ทิศทาง ทาให้มอเตอร์ทำงานได้ทั้ง 2 ทิศทาง แบบเรครันแนลแอมบลิเฟ จะแบ่งได้ 2 แบบ คือ ลิเนียร์ไบโพลาร์ และ ลิเนียร์บริดจ์ ในที่นี้จะเลือกใช้แบบ ลิเนียร์ไบโพลาร์ เพราะว่าสามารถป้อนแรงดัน และกระแสทั้งเข้าบวกและลบให้กับมอเตอร์ได้ การทำงานจะเป็นไปได้ 2 ลักษณะ คือ สำหรับการทำงานแบบฟอร์เวิร์ด และแบบรีเวอร์ส โวลต์เคจที่ป้อนให้มอเตอร์ เท่ากับ ผลต่างระหว่างซัพพลายโวลต์เคจ และโวลต์เคจคกคร่อมคอลเลคเตอร์ - อิมิตเตอร์ ของเอาต์พุททรานซิสเตอร์ทั้ง 2 ตัว พบว่าลิเนียร์แอมบลิเฟ จะมีประสิทธิภาพการทานานไม่ค่อนัก โดยเฉพาะอย่างยิ่งที่สภาวะความเร็วต่ำ แรงบิดสูง กระแสจะไหลมาก และโวลต์เคจคกคร่อมเอาต์พุททรานซิสเตอร์สูงมาก ทาให้ต้องใช้ตัวระบายความร้อนขนาดใหญ่

สามารถสรุป ข้อดีและข้อเสียของลิเนียร์ไบโพลาร์ได้ดังนี้

1. เพาเวอร์ซัพพลาย ต้องใช้ทั้งแบบบวกและลบ
2. $V_{ce(sat)}$ ของเพาเวอร์ทรานซิสเตอร์ ต้องสูงกว่าผลบวกของซัพพลายโวลต์เคจทั้งสอง ($2 V_S$)
3. ลิเนียร์ไบโพลาร์จะค่อนข้างมีปัญหา (Secondary Breakdown) ในช่วงการลดความเร็วลงของโหลดที่มีความเฉื่อยสูง เมื่อลดลงจากความเร็วสูงๆ



รูปที่ 2.9 ภาคเอาต์พุตของดีเบียร์โมเตอร์

การออกแบบเพาเวอร์แอมพลิฟายเออร์

1. การเลือกอุปกรณ์เอาต์พุต พิจารณาคงนี้คือ

- กระแส และ โวลต์เทจสูงสุด ที่มอเตอร์ต้องการ
- กระแสที่ลดลง ของการเพิ่ม หรือลดความเร็วของมอเตอร์ ต้องใช้ ช่วง

เวลานานเท่าไร

- แบนคิวิตีของ เพาเวอร์แอมพลิฟายเออร์ จะถูกกำหนดได้โดยตัวจำกัดกระแสใน

ช่วงความเร็วที่ต้องการ

- ออกแบบตามความต้องการว่าจะ เป็นแบบเบรคเกอร์ หรือ แบบบริคจ์ แบบ
- บริคจ์จะต้องการ $V_{ce0}/2$ สำหรับ พอร์เวอร์ตเซ็คชันดารี เบรคคาวน์

2. การออกแบบส่วนเอาต์พุตทรานซิสเตอร์ จะต้องมีคุณสมบัติการทำงานที่สม

มาตรกัน โดยเฉพะอย่างยิ่งอัตราขยายและแบนคิวิตี เอาต์พุตทรานซิสเตอร์จะต้องมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

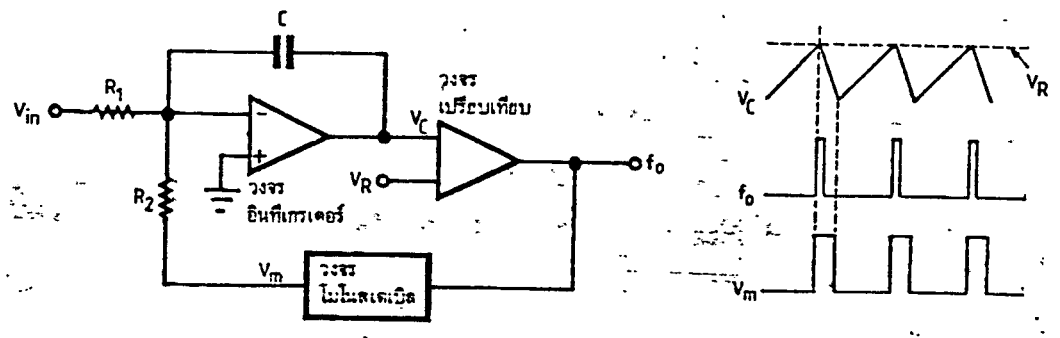
ลักษณะเป็นคู่ (Complementary Pair)

3. อัตราการขยายแบนด์วิดท์ ต้องพิจารณาถึงอัตราการขยายที่เปลี่ยนแปลงไปกับความถี่ โดยพิจารณาถึงสมการของทรานส์เฟอร์ฟังก์ชัน อัตราการขยายสามารถบังคับได้ด้วยการป้อนกลับ เพื่อรักษาคุณสมบัติการขยายของโวลต์เคจหรือกระแส ให้อัตราขยายที่ทรานซิสเตอร์ภาคเอาต์พุตจะถูกต่อกันในลักษณะของ เคอร์เรนท์ มีเรอร์ (Current Mirror) เพื่อกำจัด ริปเปิล (Ripple) ในเพาเวอร์ซัพพลาย และทำให้กระแสเอาต์พุตเป็นฟังก์ชันกับกระแสของภาคขับ ไม่ขึ้นกับค่า v ของทรานซิสเตอร์ และซัพพลาย

4. ครอสโอเวอร์ ดิสทอร์ชัน (Cross Over Distortion) เกิดจากสาเหตุเอาต์พุตทรานซิสเตอร์นำกระแสได้ค่า หรือสูญเสียการนำกระแส เนื่องจาก V_{be} ของเอาต์พุตทรานซิสเตอร์มีค่ามากเสียก่อนจึงจะเริ่มนำกระแสได้ การผิดเพี้ยนเช่นนี้จะทำให้เกิดปัญหาในการบังคับตำแหน่งของมอเตอร์ การที่จะขจัดผิดผลาดนี้ได้จะใช้ไดร็อคเป็นตัวอย่าง ออกเพิ่วโวลต์เคจ

5. การจำกัดกระแส ต้องมีคุณสมบัติดังต่อไปนี้

- ต้องรักษาคุณสมบัติการลิมิตได้ราบเรียบ แม้ว่าจะเกิดโอเวอร์โวลท์ครอสโอเวอร์จำนวนมาก
- ต้องมีผลตอบสนองที่รวดเร็วพอที่จะป้องกันกระแสโอเวอร์ชูท เมื่อมอเตอร์ลัดอัตราความเร็วลง
- ควรต่ออยู่ในลักษณะลูบิก เพื่อป้องกันค่าแรงดันผิดผลาด (Error Voltage) ที่ได้จากส่วนซั้มมิง แอมพลิฟาย์ (Summing Amplifier)
- ส่วนจำกัดกระแส ควรอยู่ใกล้เอาต์พุตทรานซิสเตอร์มากที่สุด



รูปที่ 2.11 วงจรเปลี่ยนแรงดันเป็นความถี่เป็นความถี่เบื้องต้น

จากรูป 2.11 เป็นวงจรพื้นฐานของการเปลี่ยนแรงดันเป็นความถี่ ประกอบด้วย วงจรอินทิเกรเตอร์ กาเน็คสัญญาณกระแสรูปฟันเลื่อย วงจรเปรียบเทียบ และ วงจรมอสเตเบิล ลักษณะการทำงานจะเป็นแบบคอสเมื่อมีแรงดันอินพุตเข้ามา ตัวเก็บประจุ (C) จะประจุแรงดันที่ V_0 ให้สูงขึ้น ขณะเดียวกันขาอินพุตของวงจรเปรียบเทียบจะมีแรงดันอ้างอิง (V_R) ให้อยู่ เมื่อแรงดัน V_0 สูงเท่ากับ V_R วงจรเปรียบเทียบจะทำงาน ทำให้เกิดพัลส์ f_0 ออกมาที่เอาท์พุท และพัลส์นี้จะไปทริกวงจรมอสเตเบิลทำงาน เพื่อทำการคายประจุ ตัวเก็บประจุ C

สังเกตความสัมพันธ์ของสัญญาณทั้ง 3 จุด จะเห็นว่า f_0 ซึ่งเป็นเอาท์พุทที่ต้องการ จะขึ้นอยู่กับระดับแรงดันของอินพุต หากแรงดันอินพุตสูงค่าความถี่ที่ได้ออกก็จะสูงตาม

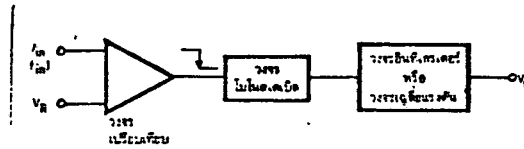
การกำหนดย่านความถี่ สามารถกำหนดค่าได้ที่ตัวต้านทาน อินพุต และตัวเก็บประจุ ในวงจรมอสเตเบิล ความถี่ตรงของวงจรขึ้นอยู่กับวงจรเปรียบเทียบ และ ความถี่เป็นเชิงเส้นของกระแสรูปฟันเลื่อย วงจรเปลี่ยนแรงดันเป็นความถี่ถูกนำมาใช้ในงานวงจรเปลี่ยนสัญญาณอนาล็อก เป็นดิจิทัล

การเปลี่ยนความถี่เป็นแรงดัน

มักถูกนำมาใช้ในภาคแรกๆของ เครื่องมือวัด หรือวงจรตรวจจับสัญญาณความถี่ก่อนจะ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้ใช้เพื่อประโยชน์ทางการศึกษาเท่านั้น การนำเอกสารนี้ไปใช้โดยไม่ผ่านการอนุญาตจากเจ้าของเอกสารถือว่าผิดกฎหมาย และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปลี่ยนเป็นแรงดัน เพื่อนำไปประมวลผลต่อไป วงจรเปลี่ยนความถี่เป็นแรงดัน แสดงไว้ดัง
รูปที่ 2.12 ประกอบด้วย วงจรเปรียบเทียบ วงจรอินทิเกรเตอร์ หรือ วงจรเฉลี่ยแรงดัน



รูปที่ 2.12 บล็อกไดอะแกรมวงจรเปลี่ยนความถี่เป็นแรงดัน

ความถี่อินพุต (f_{in}) จะถูกป้อนเข้าที่ขาอินพุตขาหนึ่งของวงจรเปรียบเทียบ ขาอินพุตอีกขาหนึ่งคืออยู่กับแรงดันอ้างอิง (V_R) เอาท์พุทจากวงจรเปรียบเทียบ จะไปทริกวงจรมานัสเตเบิล เข้าวงจรเฉลี่ยแรงดัน เพื่อให้ได้แรงดันเอาท์พุท V_o ออกมา จะสังเกตเห็นว่า การทำงานจะกลับกันกับวงจรเปลี่ยนแรงดันเป็นความถี่ ความแม่นยำของวงจรจะขึ้นกับความไวของวงจรเปรียบเทียบ และ ความเที่ยงตรงของวงจรเฉลี่ยแรงดัน

2.10 มอเตอร์กระแสตรง

มอเตอร์กระแสตรงจะสามารถพบในอุปกรณ์ใช้งานได้โดยทั่วไป ในโรงงานอุตสาหกรรมที่ได้นำมาใช้กันอย่างกว้างขวาง ตั้งแต่ในเครื่องคอมพิวเตอร์ขนาดเล็กที่ใช้ในเครื่องมือช่าง จนถึงแบบที่มีกำลังเป็นพันวัตต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

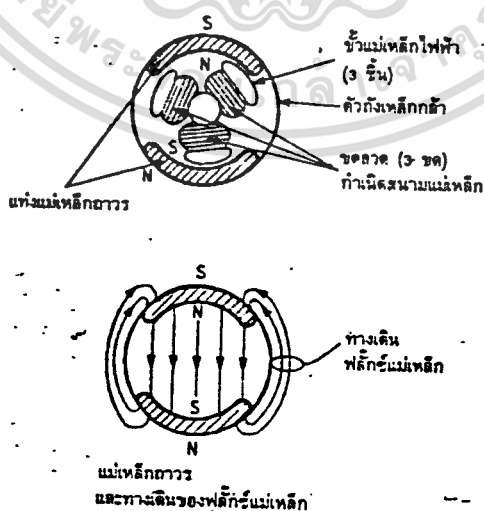
หลักการทางาน

ถึงแม้การช่างานมีขอบเขตอย่างกว้างขวาง แต่เมื่อเคอร์กระแสดรทรงทุกชนิดมีหลักการทางานเดียวกัน โดยการผ่านกระแสไฟฟ้าให้ขดลวดในสนามแม่เหล็ก ซึ่งจะทาให้เกิดแรงแม่เหล็ก สักส่วนของแรงนี้ขึ้นอยู่กับกระแสและกำลังของสนามแม่เหล็ก แรงจะเกิดขึ้นเป็นมุมฉากกับกระแสและสนามแม่เหล็ก ขณะที่ทิศทางของแรงจะกลับตรงกันข้าม ถ้ากระแสของสนามแม่เหล็กไหลย้อนกลับ การเปลี่ยนแปลงของสนามแม่เหล็กและกระแส จะเป็นผลให้ทิศทางของแรง เปลี่ยนแปลง เช่นกัน ด้วยคุณสมบัติเช่นนี้ทาให้เมื่อเคอร์กระแสตรงกลับทิศทางการหมุนได้

พิจารณาถึงส่วนของแท่งแม่เหล็กถาวร สนามแม่เหล็กของมอเตอร์ที่เกิดขึ้นส่วนหนึ่ง จะขึ้นอยู่กับจำนวนของแท่งแม่เหล็กถาวร ซึ่งจะถูกยึดติดกับแผ่นเหล็ก หรือขึ้นส่วนที่เป็นเหล็กกล้า และบางแบบก็จะทาด้วยเหล็ก โดยปกติส่วนนี้จะเป็นส่วนที่อยู่กับที่ของมอเตอร์ ขดลวดเหนี่ยวนำจะถูกพันอยู่บนส่วนที่เป็นแกนหมุนของมอเตอร์

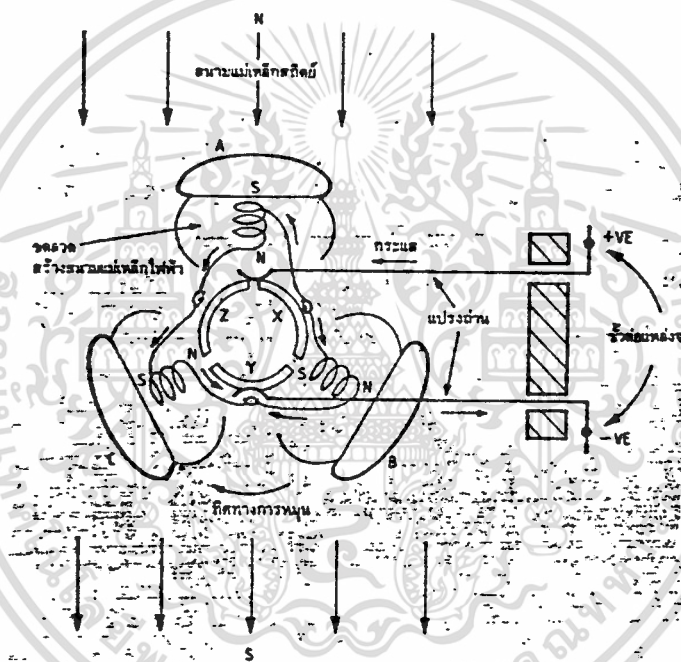
โครงสร้างโดยทั่วๆไปของมอเตอร์กระแสตรง สามารถพิจารณาได้เป็น 3 ส่วน ดัง

รูปที่ 2.13



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์โดย บริษัท อีซีเอส จำกัด หากมีข้อผิดพลาดประการใด ขออภัยไว้ก่อน และขอสงวนสิทธิ์ในเนื้อหา ไม่สามารถนำเนื้อหาไปใช้ประโยชน์ด้านการค้า
รูปที่ 2.13 โครงสร้างเบื้องต้นของมอเตอร์กระแสตรง
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป แสดงให้เห็นถึงโครงสร้างมอเตอร์ขนาดเล็ก ทางเดินของหลักแม่เหล็ก และ สนามแม่เหล็ก ซึ่งเกิดจากแท่งแม่เหล็กเพอร์มาลิต 2 ชิ้น ขึ้นเป็นแบบคั้ง ยึดกับค้ำคัง ใต้หอคี้ เพื่อให้เส้นแรงแม่เหล็กวิ่งเข้าสู่ใจกลางของมอเตอร์ ความเข้มสนามแม่เหล็กจะ ขึ้นอยู่กับความหนาของแม่เหล็กค้ำขย หลักแม่เหล็กจะวิ่งแบบค้ำคัง กระแสไฟฟ้าในขดลวด ที่พันกับทุ่นโรเตอร์ จะทำให้เกิดสนามแม่เหล็กที่มีแรงมากกว่า กระแสจะผ่านไปยังส่วนโร เตอร์โดยผ่านแปรงถ่าน ซึ่งสัมผัสกับแหวนตัวนำบนส่วนโรเตอร์ (คอมมิวเตเตอร์) ถูก แบ่งเป็น 3 เซกเมนต์ (Segment) ทาหน้าที่ นำกระแสเข้าสู่ขดลวด



รูปที่ 2.14 แสดงหลักการทางกายภาพของมอเตอร์กระแสตรง

จากรูปที่ 2.14 แสดงให้เห็นการทำงานของมอเตอร์ กระแสในขดลวด B จะมีทิศทางตรงข้ามกับกระแสในขดลวด A และขดลวด C โดยที่ขดลวดทั้ง 3 ต่อกันเป็นลักษณะอนุกรม ทิศทางการไหลของกระแสในขดลวด B ทำให้ขั้วแม่เหล็ก B มีสภาพเป็นขั้วเหนือถูกดูดเข้าทางขั้วใต้ของสนามแม่เหล็กสถิตย์ ส่วนขดลวด A และ C มีสภาพเป็นขั้วใต้ จึงถูกดูดเข้าทางขั้วเหนือ แรงดึงดูดแบบนี้จะทำให้ส่วนโรเตอร์เกิดแรงบิดมีทิศทางเข็มนา

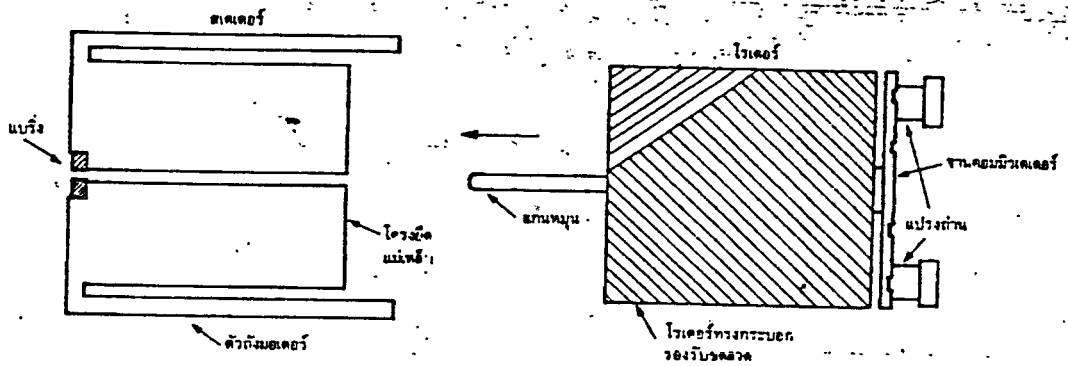
ฬิกา เมื่อโรเตอร์หมุนไปได้เล็กน้อย แปรงถ่านจะสัมผัสกับเซกเมนต์ Z ของคอมมิวเต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงแหล่งที่มาที่ใช้

ส่วนทิศทางของกระแสในขดลวดที่เหลืออีก 2 ขด ทิศทางเดิม ขั้ว A ก็จะกลายเป็นขั้วเหนือและถูกผลักออกจากขั้วเหนือของสนามแม่เหล็กสถิตย์ แบรงก่านขั้วลบจะเปลี่ยนจากอามาเจอร์เซกเมนต์ (Amature Segment) Y เป็นเซกเมนต์ X กระแสในขดลวด B มีทิศทางกันข้ามกับทิศทางเดิม และขั้ว B ซึ่งเป็นขั้วใต้จะถูกผลักออกจากขั้วใต้ของสนามแม่เหล็กสถิตย์ การหมุนจะเป็นวัฏจักรในทิศทางนี้ไปเรื่อย ๆ จนกว่ากระแสภายนอกจะกลับทิศทาง

ลักษณะการทำงานของมอเตอร์กระแสตรงที่ได้กล่าวมาข้างต้น เป็นแบบมีขดลวด 3 ขด ซึ่งเป็นโครงสร้างแบบง่าย ๆ ซึ่งจะพบว่าในการทำงานที่ความเร็วสูงจะไม่มีปัญหาเกี่ยวกับแรงบิด (Torque) เพราะความเฉื่อยของมันสามารถชดเชยได้แต่การใช้งานที่ความเร็วต่ำ ๆ จะพบปัญหาอันเนื่องมาจากขั้วแม่เหล็กอยู่ห่างกันทำให้มีอาการกระตุกขณะหมุนซึ่งปัญหาเหล่านี้จะสามารถแก้ไขได้โดยใช้โรเตอร์ที่มีจำนวนขั้วแม่เหล็กมาก ๆ และจำนวนเซกเมนต์ของคอมมิวเตเตอร์มากตามไปด้วย การออกแบบในส่วนของสเตเตอร์ (Stator) ก็จะต้องมีความละเอียด เพื่อให้สนามแม่เหล็กกระจายตัวได้ดีขึ้น ระบบการควบคุมแบบเซอร์โวก็จะเป็นอย่างหนึ่งของการใช้งานมอเตอร์ให้มีประสิทธิภาพดีที่ความเร็วต่ำและการตอบสนองรวดเร็ว

มอเตอร์กระแสตรงอีกชนิดหนึ่งที่พบอยู่บ่อย ๆ คือ มอเตอร์ชนิดที่โรเตอร์ ไม่มีแผ่นเหล็กอยู่เลย จะมีก็แค่ขดลวดพันกันเป็นรูปทรงกระบอก ปลายข้างหนึ่งต่อกับคอมมิวเตเตอร์ที่มีรูปร่างเป็นแผ่นกลม สนามแม่เหล็กที่ทำให้ส่วนโรเตอร์หมุนคืออยู่ระหว่างแกนเหล็กศูนย์กลาง กับ เบล็อกนอกของทรงกระบอกดังรูปที่ 2.15 ใต้แสดงโครงสร้างของมอเตอร์ลักษณะนี้

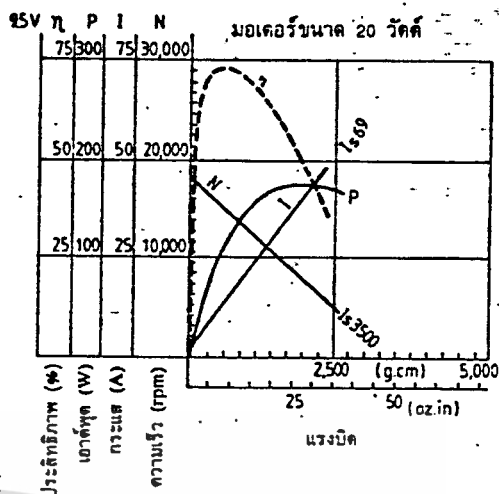
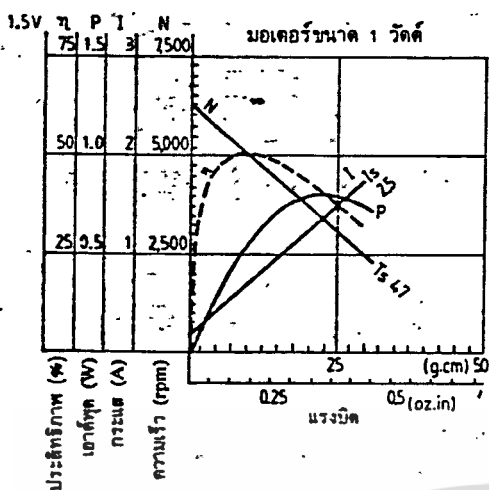


รูปที่ 2.15 แสดงส่วนประกอบของมอเตอร์กระแสตรงทั้งส่วนที่เคลื่อนที่ได้ และ เคลื่อนที่ไม่ได้

มอเตอร์ที่กล่าวมาข้างต้นมีการสร้างสนามแม่เหล็กโดยแม่เหล็กถาวร แต่ยังมีมอเตอร์อีกกลุ่มหนึ่งที่มีการสร้างสนามแม่เหล็กด้วยไฟฟ้า แต่โดยทั่วไปแล้วจะเป็นมอเตอร์แบบใด ลักษณะการต่อขดลวดจะมี 3 ลักษณะ คือ อนุกรม ขนาด และผสม การเปลี่ยนแปลงแรงดันเหนี่ยวนำของสนามแม่เหล็ก จะสามารถควบคุมได้เท่าที่ขดลวดในส่วนโรเตอร์จะหนาได้ มอเตอร์จึงมีบทบาทมากในระบบอุตสาหกรรม เพราะ เนื่องจาก เป็นอุปกรณ์ที่ง่ายต่อการควบคุมนั่นเอง

คุณสมบัติของมอเตอร์

จากที่กล่าวมาแล้วว่าแรงที่เกิดจากกระแสที่ผ่านคานาเข้าไปยังสนามแม่เหล็ก ต้องมีสภาวะที่เหมาะสมนั้น ทาให้เกิดข้อจำกัดของแรงบิดในคานมอเตอร์ รูปที่ 2.16 ได้แสดงกราฟคุณสมบัติของมอเตอร์ขนาดเล็กที่มีกำลังประมาณ 1 วัตต์ และ อีกรูปหนึ่งเป็นกราฟของมอเตอร์ขนาดใหญ่ที่มีกำลังประมาณ 80 วัตต์ เปรียบเทียบกันโดยให้แรงดันคงที่



รูปที่ 2.16 กราฟแสดงคุณสมบัติของมอเตอร์ ขนาด 1 วัตต์และ 80 วัตต์

จากรูป จะสังเกตเห็นว่าความสัมพันธ์ระหว่างกระแส กับ แรงบิด เป็นเส้นตรง ถ้า
 นำค่านี้ถึงแรงดันที่แน่นอน และความเร็วในการหมุน จะพบว่าอัตราส่วนแรงบิด และกระแส
 (T/I) ทุกจุดจะเท่ากัน ทั้งนี้ขึ้นอยู่กับความเร็วรอบ, กำลังของแม่เหล็ก, ชนิดและ
 จำนวนของแผ่นเหล็กในส่วนโรเตอร์ สเตเตอร์ และ ช่องว่างระหว่างโรเตอร์กับสเต
 เตอร์ ความสัมพันธ์ที่สองที่สังเกตได้จากกราฟ คือ ความเร็วเปรียบเทียบกับแรงบิด ซึ่ง
 ความเร็วขณะไม่มีโหลดจะมีกราฟเป็นเส้นตรง เพื่อที่จะอธิบายคุณสมบัติของมอเตอร์ให้ได้
 จะเขี่ยคัมมิ่งขึ้น ต้องพิจารณาแรงดันที่แน่นอนและความต้านทานของโรเตอร์ด้วย วงจรภายใน
 มอเตอร์เขียนได้ดังรูปที่ 2.17 โดยสมมติให้ส่วนโรเตอร์มีความต้านทานอยู่เลข
 อนุกรมกับความต้านทาน ซึ่งก็คือ ความต้านทานของขดลวดนั่นเอง แรงดันที่หัวต่อสาย
 ของมอเตอร์ คือ ผลบวกระหว่างแรงดันที่ส่วนโรเตอร์ (V_A) และ แรงดัน
 ตกคร่อมความต้านทานขดลวด (V_R) แรงดัน V_A ถูกเรียกว่า แรงเคลื่อนเหนี่ยวนำ
 บวกกลับ (Back EMF) ซึ่งเกิดขึ้นในขดลวดโรเตอร์ขณะที่หมุนแรงดันที่เกิดขึ้นนี้เป็นไปตาม
 กฎการเหนี่ยวนำแม่เหล็กไฟฟ้า จากการเคลื่อนที่ของตัวนำในสนามแม่เหล็ก สัมพันธ์กับ
 แรงเคลื่อนเหนี่ยวนำแม่เหล็ก และความเร็วในการเคลื่อนที่ของตัวนำ แรงดันที่เกิดขึ้นจะ
 มีขั้วตรงกันข้ามกับแรงดันที่ป้อนให้มอเตอร์

เอกสารประกอบการเรียน การศึกษา และแปรผันตรงกับความเร็วของการหมุน

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลบวกของแรงดันที่ส่วนรีเคอร์ (V_A) และแรงดันตกคร่อมขดลวด (V_R) ต้องเท่ากับแรงดันที่อินพุตให้มอเตอร์ (V)

$$V = V_A + V_R \quad (V)$$

พิจารณาตั้งแต่มอเตอร์หยุดนิ่ง ความเร็วมีค่าเป็นศูนย์ ดังนั้น $V_A = 0 \text{ V}$, $V_R = V$ กระแสที่ไหลในมอเตอร์หาได้จาก

$$I = V_R / R \quad (A)$$

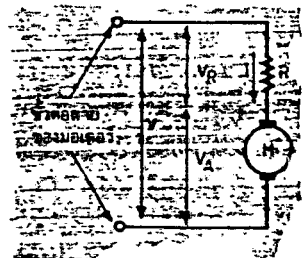
เมื่อมอเตอร์เริ่มหมุน จะมีความเร็ว และ V_A เพิ่มขึ้นเป็นเส้นตรงตามความเร็ว V_R ซึ่งมีค่าเท่ากับ ความตกต่างระหว่าง V_A และ V จะเริ่มลดลง กระแส I ก็เริ่มลดลง เช่นกัน

ขณะที่มอเตอร์ยังมีความเร็วอยู่ ความเร็วจะเพิ่มขึ้น แรงบิดจะลดลงจนกว่าจะถึงจุดซึ่งแรงบิดของมอเตอร์รับภาระโหลดได้สมกู่พอดี ขณะที่มอเตอร์ไม่มีโหลดและหมุนได้อย่างอิสระจะมีเพียงแต่ค่าความฝืดของแบร์ริง และ แรงต้านอากาศ ทำให้ค่า V_A เกือบจะเท่ากับค่า V ในรูปที่ 2.17 แรงดันที่อินพุตของมอเตอร์ ก็คือ ผลบวกของ V_A และ V_R ที่ทุก ๆ ความเร็ว และ V_A จะแปรผันตรงกับความเร็ว ส่วนกระแสและแรงบิด จะแปรผันตรงกับ V_R ความสัมพันธ์ระหว่างแรงบิด และความเร็ว เมื่อเขียนเป็นเปอร์เซ็นต์ ขณะที่แรงดันจากภายนอกที่ จะเป็นเส้นตรงเช่นกัน สามารถสรุปได้ว่า

$$\text{ความเร็ว} = 1 - \text{แรงบิด}$$

การควบคุมขั้นพื้นฐาน

1. การควบคุมโดยวิธีควบคุมตำแหน่ง
2. การควบคุมโดยวิธีเปลี่ยนค่าแรงดัน



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และ **รูปที่ 2.17 แสดงวงจรภายนอกของมอเตอร์**

ตัวจับเวลา/ตัวนับ (Timer /Counter)

MCS-31 มี 16 ไท ตัวจับเวลา/ตัวนับ 2 ตัว คือ Timer/Counter 0 และ Timer/Counter 1 ขณะนี้แต่ละ ตัวจับเวลา/ตัวนับ (Timer/Counter) สามารถที่จะ กำหนดค่าให้ทำงานได้เป็นตัวจับเวลา หรือ ตัวนับ

ตัว Timer/Counter 0 และ Timer/Counter 1

แต่ละตัวจะถูกกำหนดค่าให้ทำงานเป็น ตัวจับเวลา หรือตัวนับ ได้ด้วยการเซ็ท หรือ เคลียร์ ที่ตัวควบคุมในรีจิสเตอร์ TMOD ในกลุ่ม SFR

ในฟังก์ชันตัวจับเวลา ตัวรีจิสเตอร์จะเพิ่มค่าทุก ๆ 1 ในวัฏจักรแมชชีน (Machine Cycle) ดังนั้น ตัวเลขในรีจิสเตอร์จะเป็นจำนวนของวัฏจักรแมชชีน เนื่องจากแต่ละวัฏจักรแมชชีน ประกอบด้วย 12 คาบของออสซิลเลเตอร์ อัตราการนับแต่ละครั้งจะกินเวลา เป็น 1/12 ของความถี่ออสซิลเลเตอร์

ในฟังก์ชันตัวนับ รีจิสเตอร์จะเพิ่มค่าทุกครั้งที่มีการเปลี่ยนแปลงสถานะ จาก "1" เป็น "0" ที่เข้ามาที่ขา T0 หรือ T1 ในฟังก์ชันนี้ สัญญาณภายนอกที่เข้ามาจะถูกแซมปลิง (Sampling) ระหว่างช่วง S5P2 ของทุกวัฏจักรแมชชีน โดยถ้าการแซมปลิงสัญญาณเข้า เป็นระดับสูงในวัฏจักรหนึ่ง ดังนั้น ถ้าวัฏจักรตัวต่อมาของสัญญาณเข้าเป็นระดับต่ำ รีจิสเตอร์จะนับเพิ่มหนึ่งค่า โดยที่ค่าใหม่ของตัวนับจะปรากฏที่รีจิสเตอร์ ช่วง S3P1 ของวัฏจักร ซึ่งค่าหนึ่งที่ได้รับเข้าจะใช้เวลา 2 วัฏจักรแมชชีน (เท่ากับ 24 คาบ) ในการรับ ค่าช่วงการเปลี่ยนแปลง "1" เป็น "0" ดังนั้น ค่าสูงสุดในการนับจะมีอัตรา 1/24 ของความถี่ออสซิลเลเตอร์ และสัญญาณอินพุตที่นับนั้น จะไม่มีช่วงระยะห่างที่แน่นอนของ ivity ไซเคิล (Duty Cycle) แต่จะถูกนับเมื่อระดับแรงดันที่ถูกแซมปลิง ในแต่ละครั้งจะต้องมีช่วงคงที่อย่างน้อย 1 วัฏจักรแมชชีน ก่อนที่จะเปลี่ยนค่าระดับแรงดันใหม่

ในการเลือกทำงานระหว่าง ตัวนับเวลา และ ตัวจับเวลา จะเลือกได้ 4 โหมด

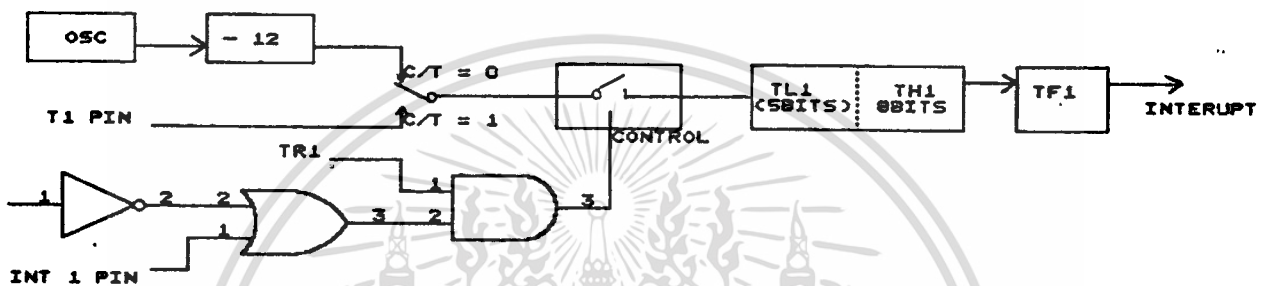
คือ โหมด 0,1 และ 2 เลือกได้ทั้งสองตัวของ ตัวจับเวลา/ตัวนับ ส่วนโหมด 3 จะทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

งานเทคโนโลยี ไม่ว่ากรรมใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมด 0

การใช้ตัวจับเวลา/ตัวนับเวลา 0 หรือ 1 ให้อยู่ในโหมด 3 จะทำงานคล้ายกับของ MCS-48 โดยตัวจับเวลาของ MCS-48 มีขนาด 8 บิต มีตัวหารเสถียร (Prescaler) เป็นตัวหาร 12 ดังรูปที่ 2.18 แสดงการทำงานในโหมด 0 ของตัวจับเวลา/ตัวนับเวลา



รูปที่ 2.18 แสดงการทำงานในโหมด 0 ของตัวจับเวลา/ตัวนับ 1 ขนาด 13 บิต

ในโหมดนี้ รีจิสเตอร์ตัวจับเวลาถูกกำหนดค่าให้มี 13 บิต ด้วยการนับขึ้น เมื่อเป็น "1" หมดทุกบิต จะกลับมากที่ "0" ทุกบิตใหม่ เมื่อกลับเป็น "0" ทุกบิตจะเป็นการเกิดรีจิสเตอร์เฟลว บทความให้เฟลกอินเทอร์รัพท์ TF1 ปรับเป็น "1" การควบคุมให้เริ่มนับตัวอินพุตจะควบคุมด้วยการอีเนเบิล $TR = 1$; $GATE = 0$; และ $INT1 = 1$ การปรับ $GATE = 1$ เป็นการตั้งตัวนับให้ถูกควบคุมด้วยสัญญาณจากภายนอกเข้า INT1 TR1 จะเป็นบิตควบคุมในรีจิสเตอร์ TMOD ของ SFR

รีจิสเตอร์ตัวนับ จะมี 13 บิต ประกอบด้วย TH1 8 Bit และ TL1 อีก 5 บิตอื่น

คัมค่า ส่วนอีก 3 บิต ที่เหลืออันอันดับสูงของ TL1 จะมาใช้ การปรับเฟลท TR1 ให้ทำ

งานจะไม่มีเคอร์ค่านรีจิสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

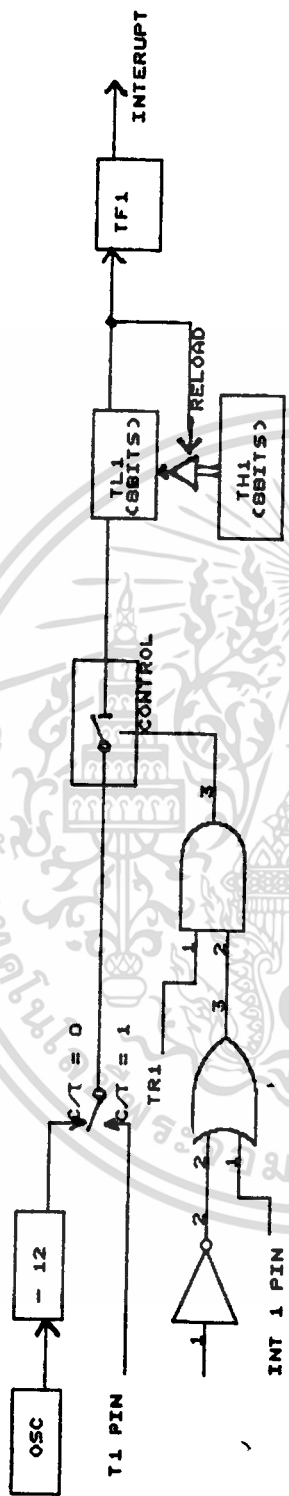
การทำงานในโหมด 0 ในตัวจับเวลา/ตัวนับ จะทำงานเหมือนกับตัวจับเวลา/ตัวนับ 1 โดยใช้ TRO และ INTO รวมกันควบคุมแทนสัญญาณต่าง ๆ ในรูปที่ 2.18 มีความแตกต่างในการควบคุม คือ โทของเกททั้งสอง ตัวหนึ่งจะแทนตัวจับเวลา/ตัวนับ 1 (TMOD.7) และอีกตัวจะแทนตัวจับเวลา/ตัวนับ 0 (TMOD.3)

โหมด 1

โหมด 1 ทำงานเหมือนกับ โหมด 0 ต่างกันแต่เฉพาะการใช้รีจิสเตอร์ ตัวจับเวลา/ตัวนับ จะทำงานนับด้วยขนาด 16 บิต

โหมด 2

โหมด 2 มีการทำงานโดยการกำหนดค่าให้ตัวนับ 8 บิต ของ TL1 และจะไหลคาบหม่โดยอัตโนมัติทุกครั้ง เมื่อมีการโอเวอร์ฟลว (Overflow) จาก TL1 ดังรูปที่ 2.19 แม้เพียงแต่ TF1 จะปรับเป็น "1" แต่ TL1 จะถูกไหลโดยอัตโนมัติจากค่าที่ตั้งไว้บน TH1 สามารถจะตั้งค่าได้ด้วยซอฟต์แวร์ และบรรจุเข้าบิตใหม่ที่ TL1 ทุกครั้งที่เกิด โอเวอร์ฟลว TH0 และ TFO จะเป็นตัวร่วมการทำงานในโหมดนี้



Size	Document Number	REV
A		
Date:	March 22, 1992 Sheet of	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 2.19 ไทม์เมอร์ / เคาน์เตอร์ 1 โหมด 2: 8 บิต ออโต-รีโหลด

บทที่ 3

การสร้างระบบควบคุมแบบแยกส่วน

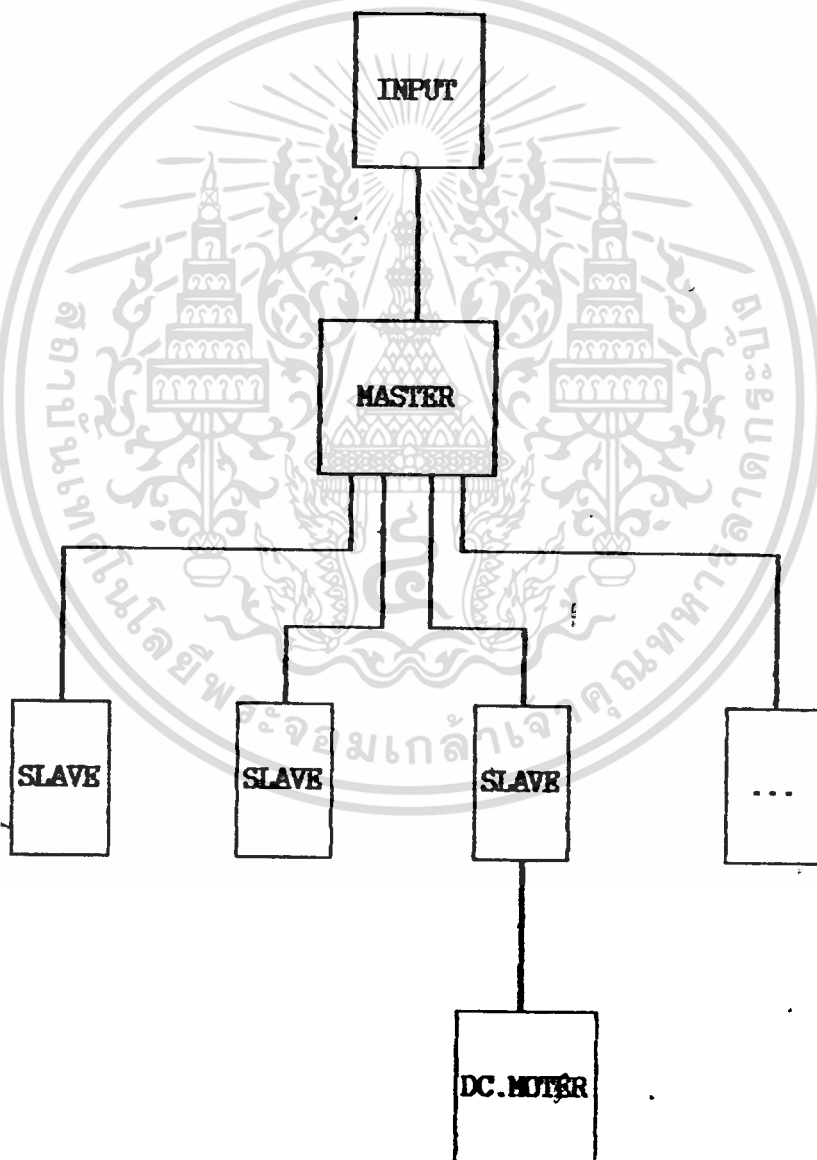
3.1 วัตถุประสงค์ของระบบควบคุมแบบแยกส่วน

- 1) เนื่องจากเพื่อความสะดวก และให้การพัฒนาในระบบควบคุมเป็นไปได้อย่างต่อเนื่องจึงเลือกใช้งานไมโครคอนโทรลเลอร์ (Microcontroller) และในระบบการควบคุมงานที่นี้ จะเลือกชิปเดี่ยว (Single Chip) เบอร์ 8031 เป็น หน่วยประมวลผลกลาง (Central Processing Unit) ซึ่งจะทำหน้าที่เป็นศูนย์กลางในการรับส่งข้อมูล ประมวลผลเพื่อการตัดสินใจ และเป็นตัวชี้แนวทางแก้ไข เมื่อมีปัญหาเกิดขึ้นในระบบ
- 2) เพราะเหตุที่ในสภาวะแวดล้อมบางสภาวะนั้น ไม่มีความเหมาะสมที่สิ่งมีชีวิตจะอยู่ได้ ดังนั้นระบบการควบคุมแบบแยกส่วน ลักษณะนี้จะช่วยอำนวยความสะดวกให้แก่ผู้ทำการควบคุม (User) ที่จะสามารถรับทราบข้อมูล จากส่วนอินพุทเทอร์มินัล (Input Port), เอาท์พุทเทอร์มินัล (Output Port) ที่ให้ทำการแสดงผลออกมา และเมื่อผู้ทำการควบคุม (User) ต้องการให้ระบบ (System) นั้นมีการตอบสนอง (Response) ที่เปลี่ยนแปลงไปจากเดิม ก็จะสามารถจะส่งคำสั่งผ่านส่วนคอมพิวเตอร์เพื่อไปสั่งให้ส่วนขบวนการ (Process) ที่มี ไมโครคอนโทรลเลอร์ เป็นตัวควบคุมให้ทำงานต่อไป
- 3) ระบบการควบคุมแบบแยกส่วน จะให้การตอบสนองต่อการเปลี่ยนแปลงของอินพุทให้เป็นไปอย่างรวดเร็ว ด้วยเพราะว่าการควบคุมแบบแยกส่วนนี้จะทำงานในลักษณะของการควบคุมแบบมีอนาคกลับ (Feedback Control) ที่เป็นลูปปิด (Closed Loop)
- 4) สำหรับขบวนการ (Process) ที่เลือกมาใช้งานในการควบคุมแบบแยกส่วนนี้ จะเป็นการควบคุมมอเตอร์ ซึ่งการทำงานของมอเตอร์ได้ถูกนำมาประยุกต์และใช้งานในอุตสาหกรรมได้อย่างมากมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) เพื่อเป็นการลดค่าใช้จ่ายในการทำงาน เพราะในสภาวะแวดล้อมที่ไม่เหมาะสมสามารถนำระบบการควบคุมแบบแยกส่วนเข้ามาควบคุมการทำงานแทนได้

3.2 บล็อกไดอะแกรม (Block Diagram) ของระบบควบคุมแบบแยกส่วน



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ 3.1 บล็อกไดอะแกรมของระบบควบคุมแบบแยกส่วน ถ้าไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การทำงานระบบควบคุมแบบแยกส่วน

การทำงานโดยทั่วไปของระบบการควบคุมแบบแยกส่วน จะมีหน่วยควบคุมหลัก (Master) ซึ่งมีหน้าที่คอยทำการควบคุมระบบทั้งหมดที่ประกอบด้วยระบบย่อยที่ถูกควบคุม โดยหน่วยควบคุมย่อยหรือ สลาฟ (Slave) ในที่นี้ส่วนหน่วยควบคุมหลักจะใช้ เครื่องไมโครคอมพิวเตอร์ (Microcomputer) ทำหน้าที่ในการรับคำสั่งและแสดงผลที่เกิดขึ้นในกระบวนการ (Process) เนื่องจากในภาวะปัจจุบัน เครื่องไมโครคอมพิวเตอร์เป็นที่นิยมใช้กัน อย่างแพร่หลาย และมีราคาไม่แพง ส่วนหน่วยควบคุมย่อย ได้ใช้ไมโครคอนโทรลเลอร์ เบอร์ 8031 เป็นตัวประมวลผล ด้วยเพราะว่า 8031 มี พอร์ตแบบอนุกรม (Serial Port) อยู่ในตัว ทำให้เหมาะสำหรับควบคุมและติดต่อในระยะไกล ส่วนหน่วยควบคุมย่อย จะทำหน้าที่ควบคุมกระบวนการย่อยทั้งหมดให้เป็นไปตามที่ส่วนหน่วยควบคุมหลักกำหนด และทำการรับผลที่เกิดขึ้นเพื่อแจ้งกลับมายังส่วนหน่วยควบคุมหลัก

ในที่นี้จะใช้หน่วยควบคุมย่อย ไปทำการควบคุมมอเตอร์กระแสตรง เพราะในปัจจุบันการควบคุมมอเตอร์กระแสตรงจะพบได้ในงานอุตสาหกรรมส่วนมาก ดังนั้นการออกแบบเพื่อการควบคุมมอเตอร์กระแสตรง เป็นเทคนิคใหม่ที่สำคัญและน่าสนใจ เนื่องจากมอเตอร์กระแสตรงเป็นอนาล็อกแอกซีวเอเตอร์ ที่ไม่มีตำแหน่งหยุดที่แน่นอนเช่นสเตปมอเตอร์ ดังนั้นการควบคุมมอเตอร์กระแสตรงโดยทั่วไปจึงมัก เป็นการควบคุมแบบระบบปิด (Closed Loop System) เพื่อให้ได้ตามคุณสมบัติตามที่ต้องการ

พอร์ตขนานของ 8031 ไมโครคอนโทรลเลอร์ จะถูกนำมาเชื่อมต่อกับส่วน แปลงสัญญาณดิจิทัลเป็นอนาล็อก (Digital To Analog Convertor)

เพื่อเปลี่ยนสัญญาณให้เป็นสัญญาณอนาล็อก (Analog) และนำมาเป็นสัญญาณอินพุตของภาคขยาย (Amplifier) จะถูกนำมาขับกระแสและโวลต์เคจ เพื่อที่จะสามารถนำมาขับมอเตอร์ได้ สมการการทำงานของมอเตอร์จะถูกรับรู้ (Detect) โดยส่วนเอนโคดเดอร์ (Encoder) สัญญาณที่วัดได้ก็จะถูกนำมาเปลี่ยนให้เป็นสัญญาณดิจิทัลเพื่อให้นักไมโครคอนโทรลเลอร์รับทราบได้ โดยการนำมาผ่านส่วนแปลงสัญญาณอนาล็อก เป็นดิจิทัล

(Analog To Digital Convertor)

เอกสารนี้เป็นเอกสารตัวอย่างที่จัดทำขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 โครงสร้างทางฮาร์ดแวร์ (Hardware) ของระบบควบคุมแบบแยกส่วน

1. ส่วนคอมพิวเตอร์ทำการแสดงผล (Display) และ ส่งคำสั่งจากผู้ควบคุมไปยังระบบ
2. ส่วนการสื่อสารแบบอนุกรม (Serial Port)
3. ส่วนไมโครคอนโทรลเลอร์ (Micro Controller) ทำการควบคุมโดยตรงกับการทำงานของระบบ
4. ส่วนแปลงสัญญาณจากดิจิทัลให้เป็นอนาล็อก (Digital To Analog Converter)
5. ส่วนขยายสัญญาณ (Amplifier)
6. ส่วนแปลงสัญญาณความถี่ให้เป็นโวลต์เตจ (Frequency To Voltage Converter)
7. ส่วนแปลงสัญญาณอนาล็อกให้เป็นดิจิทัล (Analog To Digital Converter)

1. ส่วนคอมพิวเตอร์ที่ทำการแสดงผล และ ส่งคำสั่ง

เมื่อผู้ใช้งานต้องการควบคุมระบบจะเรียกใช้โปรแกรม DCS.SXX จะขึ้นรายการหลักขึ้นมา ให้เลือกใช้งาน รายการเซตค่าบอดเรท(BOADRAT) รายการทดสอบสลาฟของระบบ (TEST) รายการเซตค่าพารามิเตอร์ต่างและควบคุม(SYSTEM) รายการจัดการเก็บและ โหลดค่าพารามิเตอร์(FILE)

รายการเซตค่าบอดเรท จะมีการเซตค่าบอดเรทที่ใช้ในการติดต่อสื่อสารทางพอร์ตอนุกรมให้เลือกใช้งาน 4 ค่าคือ 9600 4800 2400 1200 เมื่อเลือกใช้รายการส่วนคอมพิวเตอร์จะส่งชุดคำสั่งออกทางพอร์ตอนุกรม 3 ค่าเพื่อให้สลาฟนำไปทำการเซตบอดเรท คือ 60H 80H XXH ถ้าหากว่า

บอดเรทคือ 9600 ค่า XXH คือ 08H บอดเรทคือ 4800 ค่า XXH คือ 04H

บอดเรทคือ 2400 ค่า XXH คือ 02H บอดเรทคือ 1200 ค่า XXH คือ 01H

ในเบื้องต้นของการทำงาน จะทำการเซตค่าบอดเรทไว้ที่ 1200

รายการทดสอบสลาฟของระบบ จะเป็นรายการที่ใช้ทดสอบว่ามีสลาฟตัวใดบ้างกำลังทำงานอยู่และพร้อมรับคำสั่งจากส่วนคอมพิวเตอร์ โดยส่วนคอมพิวเตอร์จะส่งชุดคำสั่ง 3 ค่า คือ 60H 70H XXH โดยที่ XXH คือ เบอร์ของสลาฟที่จะทำการทดสอบ สลาฟจะตอบค่าไปรเซส วาลู กลับมาเพื่อมาทำการคำนวณต่อไป

รายการเซตค่าพารามิเตอร์ต่างและควบคุม จะมีรายการเพื่อเซตค่าเซตพอยท์ และ พารามิเตอร์ นิโอดี เมื่อมาทำการคำนวณต่อไป เมื่อทำการควบคุมจะเข้าการทำงานใน โหมดกราฟิก ส่วนของคอมพิวเตอร์จะทำการส่งค่าสถานะของระบบและชุดคำสั่งทำการเริ่ม การควบคุมออกมา ในส่วนของคอมพิวเตอร์เมื่อมีการกด F1 จะหยุดการทำงานของระบบชั่วคราวเพื่อทำการพิมพ์หน้าจอออกทางเครื่องพิมพ์ F2 จะทำการเคลียร์กราฟหน้าจอ

รายการจัดการเก็บและโหลดค่าพารามิเตอร์ เป็นการเก็บและโหลดค่าพารามิเตอร์

ที่ใช้ในการคำนวณ มีฟังก์ชันการออกจากระบบชั่วคราว(OS SHLL) มีฟังก์ชันเลิกการทำงาน
เอกสารนี้เป็นลิขสิทธิ์ของกรมส่งเสริมการค้าระหว่างประเทศ(อ.ช.สข.ล.) มีฟังก์ชันเลิกการทำงาน
ไม่ว่ากรณีใดๆก็ตาม อีกรหัสห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
งาน(ดูปีT)

2. ส่วนการสื่อสารแบบอนุกรม

การทำงานของส่วนนี้จะเป็นส่วนของการสื่อสารแบบอนุกรมที่ง่ายที่สุดกล่าวคือ ไม่จำเป็นต้องมีการส่งสัญญาณแฮนด์เช็ก (Handshake) กันระหว่างส่วนคอมพิวเตอร์ กับ ไมโครคอนโทรลเลอร์ ซึ่งข้อมูลจากคอมพิวเตอร์จะต้องมีการปรับระดับของสัญญาณให้เป็น ระดับทีทีแอล (TTL) ก่อน วิชาใช้ไอซีเบอร์ MC1489 และเช่นเดียวกัน ข้อมูลจากระบบ ส่วนไมโครคอนโทรลเลอร์ ซึ่งเป็นระดับทีทีแอล (TTL) ก็ต้องถูกแปลงให้อยู่ในระดับ มาตรฐาน RS-232 ก่อน วิชาใช้เบอร์ MC1488

3. ส่วนไมโครคอนโทรลเลอร์ 8031

เมื่อไมโครคอนโทรลเลอร์ 8031 รับส่งสัญญาณกับส่วนคอมพิวเตอร์โดยผ่านทางพอร์ตของการสื่อสารแบบอนุกรม การทำงานส่วนอื่นของไมโครคอนโทรลเลอร์ 8031 ก็จะมีการส่งข้อมูลไปยังส่วนแปลงสัญญาณดิจิทัลเป็นอนาล็อก สำหรับขับมอเตอร์กระแสตรงในทิศทางตามเข็มนาฬิกา ผ่านทางพอร์ตของตัวไมโครคอนโทรลเลอร์ 8031 เอง นอกจากนี้ ได้ทำการขยายพอร์ตโดยการใช้ไอซีเบอร์ 8255 และกำหนดให้แต่ละพอร์ตมีหน้าที่ดังนี้

PORT A ADDRESS 8000H

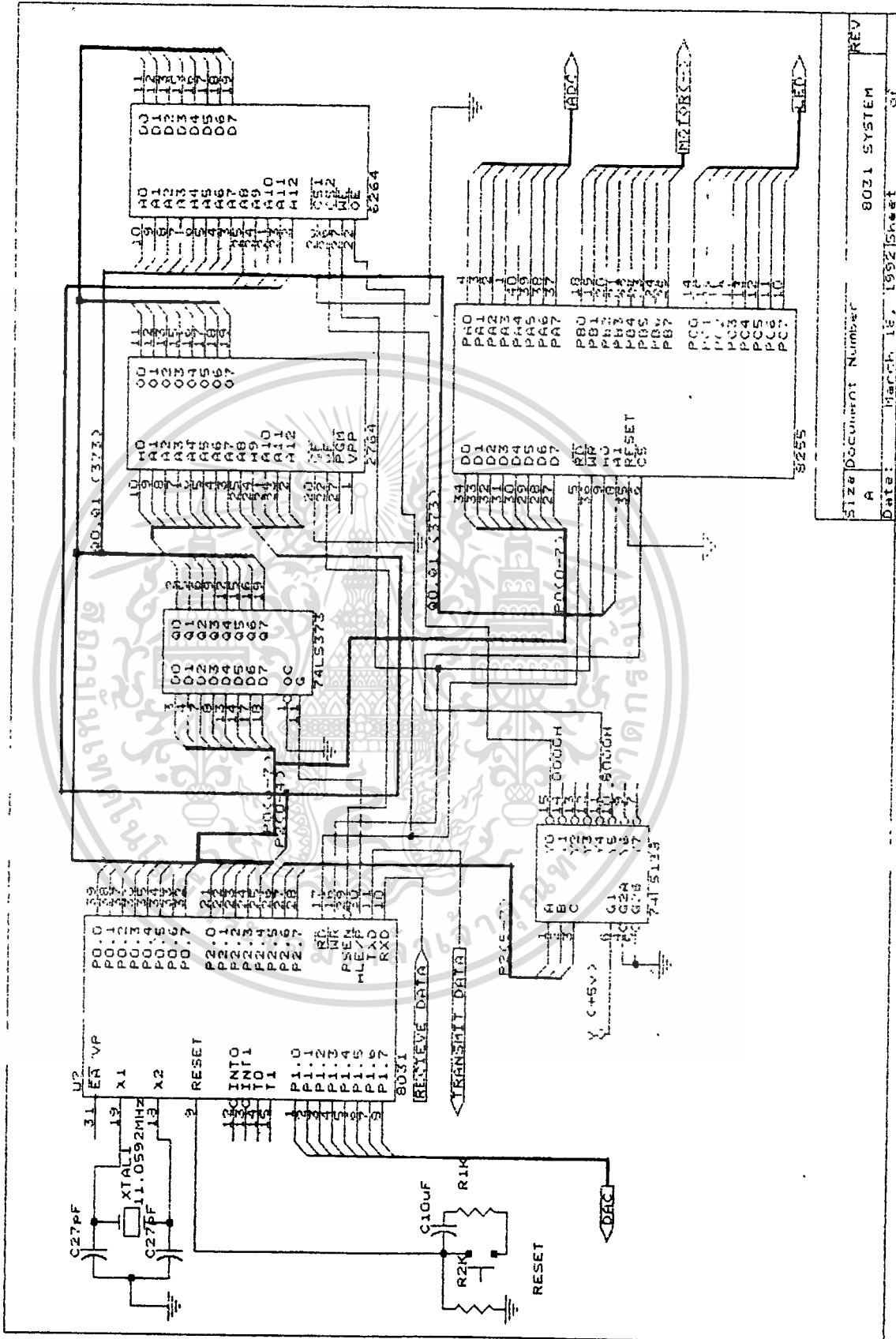
ทำหน้าที่เป็น อินพุทพอร์ต อ่านค่าจากส่วนแปลงสัญญาณ
อนาล็อก เป็นดิจิทัลเข้ามา

PORT B ADDRESS 8001H

ทำหน้าที่เป็น เอาท์พุทพอร์ต ส่งสัญญาณไปยังส่วนแปลงสัญญาณดิจิทัลเป็นอนาล็อก สำหรับขับมอเตอร์กระแสตรงใน
ทิศทางทวนเข็มนาฬิกา

PORT C ADDRESS 8002H

ทำหน้าที่เป็น เอาท์พุทพอร์ต ส่งค่าไปยัง LED เพื่อทำการ



REV	
8031 SYSTEM	
Sheet	of
Date:	March 18, 1992
Size	A
Document Number	

เอกสารนี้เป็นเอกสาร **รูปที่ 3.2** วงจรอินเทอร์เฟสระหว่าง 8031 กับหน่วยความจำ ชนด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
และ พงษ์

4. ส่วนแปลงสัญญาณดิจิทัลเป็นอนาล็อก

วงจรส่วนนี้จะเป็นส่วนที่รับข้อมูลและคำสั่ง ที่ส่งมาจากไมโครคอนโทรลเลอร์ เบอร์ 8031 เนื่องจากสัญญาณเป็นแบบดิจิทัล ยังไม่สามารถนำมาควบคุมการทำงานของระบบในที่นี้คือมอเตอร์กระแสตรงได้ จึงต้องถูก เปลี่ยนให้เป็นสัญญาณอนาล็อก เสียก่อน สำหรับ ฮาร์ดแวร์ในส่วนนี้จะประกอบด้วย 2 ส่วนย่อยที่หาหน้าที่เช่นเดียวกัน เพราะเนื่องจาก การควบคุมการทำงานของมอเตอร์กระแสตรง จะต้องเป็นแบบได้ทั้ง 2 ทิศทาง คือ ทิศทางตามเข็มนาฬิกา และ ทิศทางทวนเข็มนาฬิกา

5. ส่วนขยายสัญญาณ

เนื่องจากสัญญาณดิจิทัลที่ถูกทำการแปลง ให้เป็นสัญญาณอนาล็อกแล้วนั้น ขนาดของสัญญาณยังไม่เพียงพอที่จะนำมาขับมอเตอร์กระแสตรงได้ จึงต้องนำมาขยายสัญญาณให้มีขนาดเท่าที่คองนำมาใช้งานเสียก่อน ทั้งนี้หมายถึงขนาดของแรงดันและกระแส เพื่อให้สามารถนำมาใช้งานกับโหลดต่างวได้

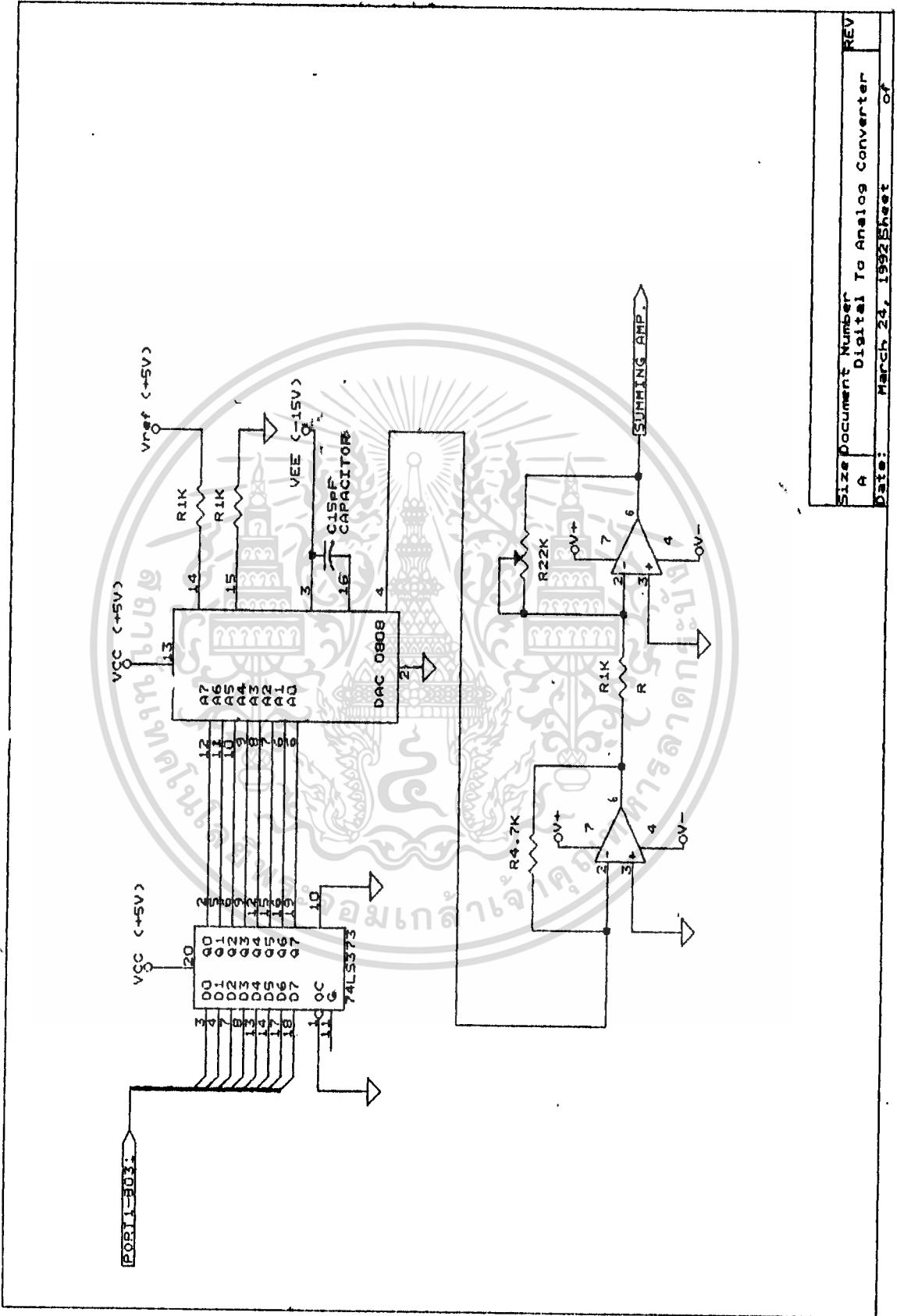
จากการออกแบบตามหลักการ จะได้ว่าวงจรขยายสัญญาณดังรูป 3.3, 3.4

6. ส่วนแปลงสัญญาณความถี่ให้เป็นจวลท์เคจ

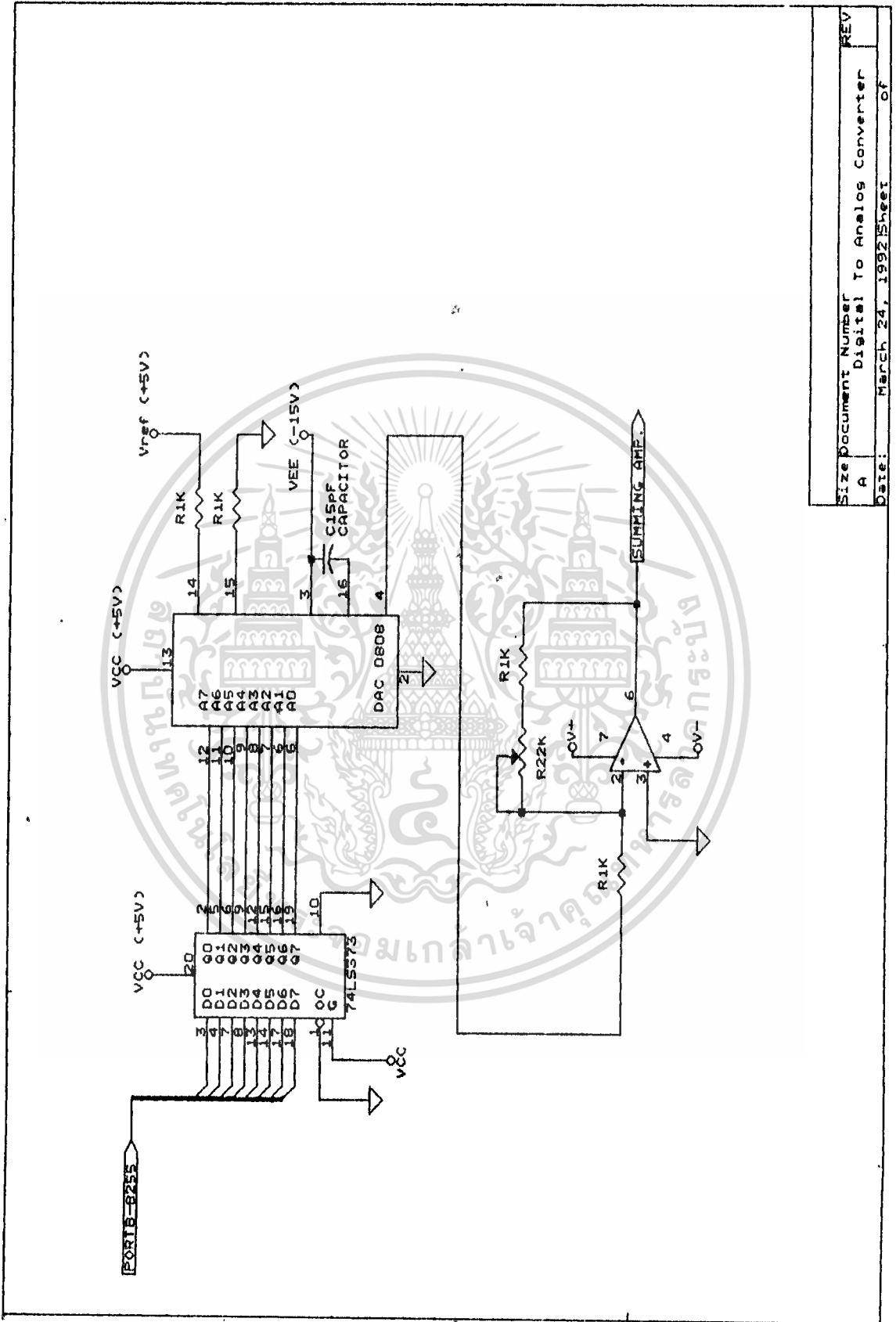
จากการทำงานของมอเตอร์กระแสตรง การควบคุม เพื่อลดความผิดพลาดนั้นจะทำได้โดยการพิจารณาจากความถี่ที่วัดได้ขณะแรงดันใช้งานนั้นๆ และ เนื่องจากสัญญาณที่วัดออกมาได้เป็นค่าของความถี่ จึงต้องนำมาแปลงให้เป็นสัญญาณอนาล็อกที่มีค่า เปลี่ยนไปตามค่าความถี่โดยการใส่ไอซีเบอร์ 2917 และ จากการออกแบบวงจรในส่วนนี้ได้กำหนดให้ค่าแรงดันเอาท์พุทมีค่าเท่ากับ 5 จวลท์ ที่ความถี่มากที่สุด และค่าแรงดันเอาท์พุทมีค่าเท่ากับ 0 จวลท์ ที่ความถี่ต่ำที่สุด

7. ส่วนแปลงสัญญาณอนาล็อกให้เป็นดิจิทัล

เอกสารนี้เป็นเอกสารของฝ่ายวิจัยและพัฒนาของศูนย์วิจัยและพัฒนาเทคโนโลยีพลังงานทดแทน ไม่ควรเผยแพร่ไปใช้ประโยชน์ด้านการค้า
 เมื่อสัญญาณความถี่ถูกแปลงให้เป็นสัญญาณอนาล็อก ดังกล่าวข้างต้นแล้วนั้นสัญญาณนี้
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

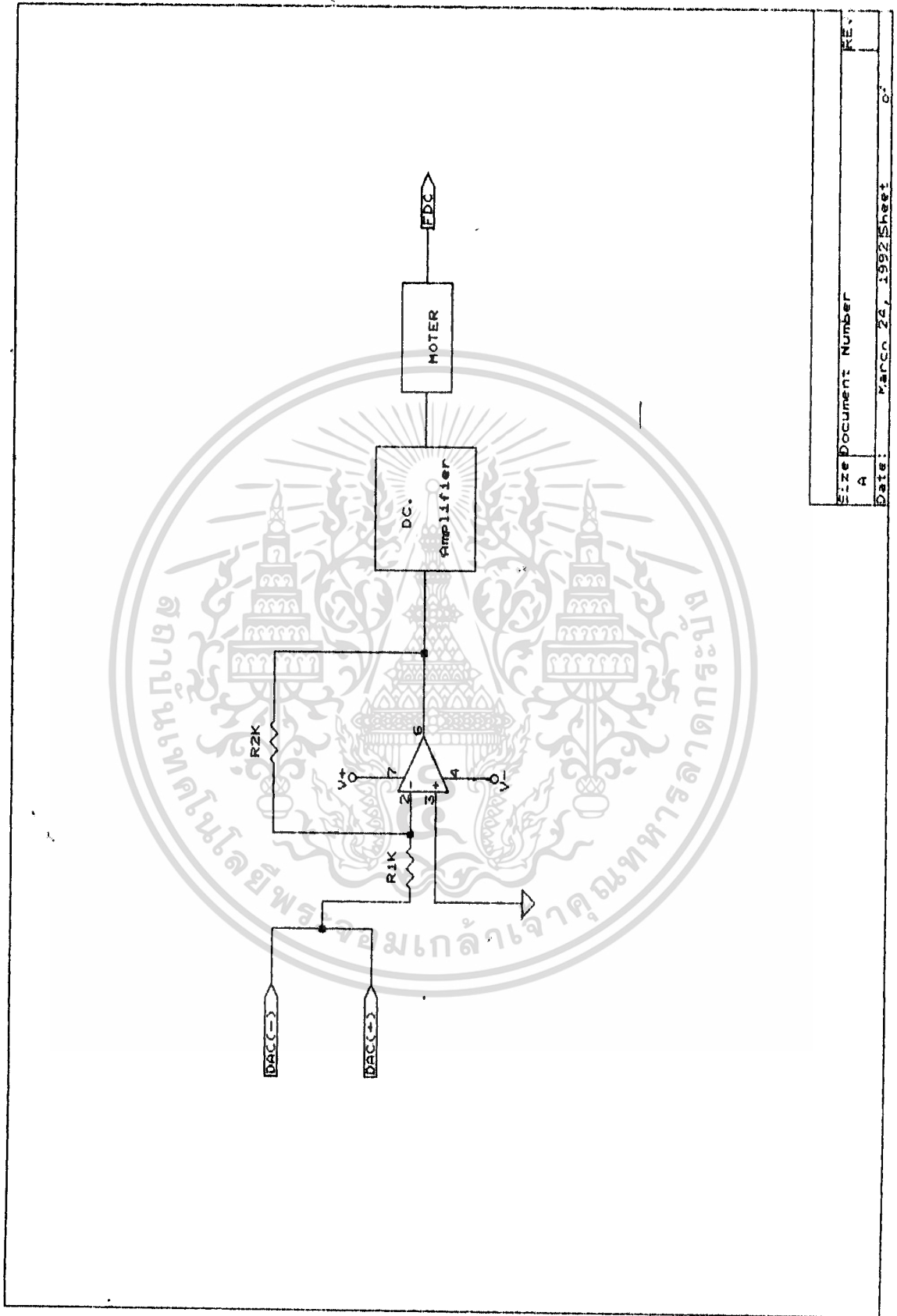


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งนี้ หากมีข้อผิดพลาดประการใดขออภัยเป็นอย่างสูงและขอสงวนสิทธิ์ในการนำไปใช้
วันที่ ๑.๑ วงจรแปลงสัญญาณดิจิทัลเป็นอนาล็อก (ภาคไฟบวก)



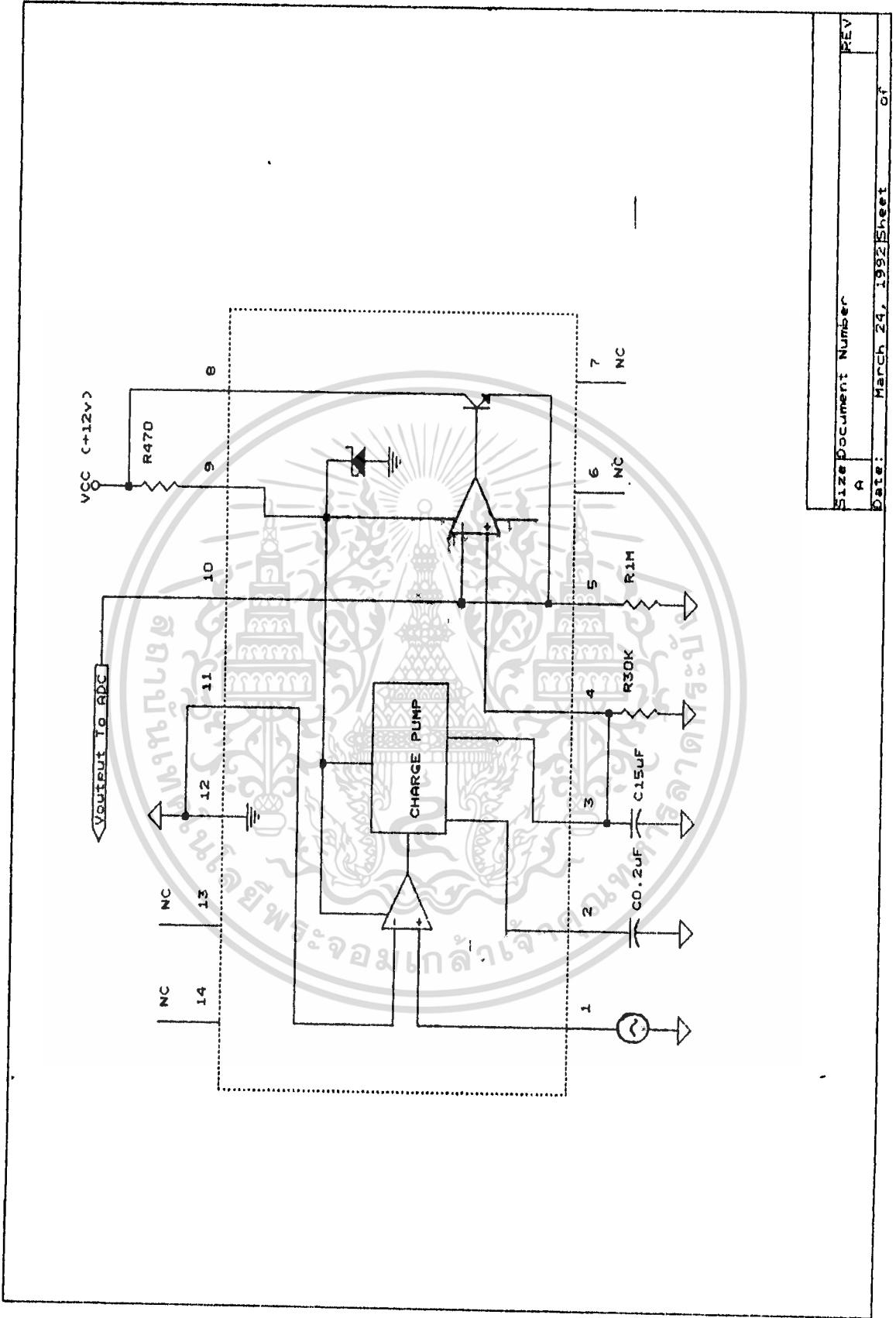
Size	Document Number	REV
A	Digital To Analog Converter	A
Date:	MARCH 24, 1992	Sheet of

เอกสารนี้เป็นเอกสารที่ **รูปที่ 3.4** วงจรแปลงสัญญาณดิจิทัลเป็นอนาล็อก (ภาคไหลย) ทำการคำนวณค่า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Size	Document Number
A	
Date:	March 24, 1992 Sheet 0.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ห้ามมิ้งแอมป์รีไฟร์
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Size	Document Number	REV
A		
Date:	March 24, 1992	Sheet of

เอกสารนี้เป็นเอกสารที่สงวน **รูปที่ ๑.๖ วงจรแปลงสัญญาณความถี่เป็นแรงดัน** ใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก็จะถูกแปลงให้เป็นสัญญาณดิจิทัลอีกครั้งหนึ่ง เพื่อที่จะให้ส่วนของไมโครคอนโทรลเลอร์สามารถนำมาทำการประมวลผล เพื่อควบคุมให้ค่าสัญญาณเอาต์พุตมีค่า เป็นไปตามที่ผู้ทำการควบคุมสั่งได้

3.5 โครงสร้างทางซอฟต์แวร์ (Software) ของระบบควบคุมแบบแยกส่วน

การทำงานของส่วนซอฟต์แวร์ เป็นส่วนประกอบที่สำคัญของระบบการควบคุมแบบแยกส่วน โดยเริ่มจากโปรแกรมที่ทำการส่งค่าจากส่วนคอมพิวเตอร์ กำหนดเงื่อนไขและวิธีการรับส่งข้อมูล ไปยังส่วนไมโครคอนโทรลเลอร์ 8031 ทั้งทำการประมวลผลตามข้อมูลที่ได้รับเข้ามา การส่งผลต่าง ๆ ไปยังส่วนฮาร์ดแวร์ โดยมีโปรแกรมเคาน์เตอร์ (Program Counter) เป็นตัวกำหนดว่าจะต้องอ่านโปรแกรมที่ตำแหน่งเท่าไร ความเร็วในการทำงานของส่วนซอฟต์แวร์ จะขึ้นอยู่กับ สัญญาณนาฬิกาที่ส่งให้ตัวประมวลผลทำงาน ดังนั้นจะเห็นได้ว่า ลำดับการทำงานของซอฟต์แวร์จะต้องมีความสัมพันธ์กับส่วนประกอบทางด้านฮาร์ดแวร์

สำหรับซอฟต์แวร์ชุดควบคุมการทำงานแบบพีเอชดีที่จากรูปขบวนการได้ดังรูปที่ จะมีหลักการคำนวณพื้นฐานโดยการนำค่าผลต่าง ระหว่างค่าที่กำหนด (Set point) กับค่าในขบวนการ (Process Value) มาพิจารณาแล้วจึงการให้ค่าในขบวนการมีค่าเป็นไปตามที่กำหนดไว้ ส่วนแอลกอริทึมสำหรับการคำนวณ จะขึ้นอยู่กับค่าของพารามิเตอร์พีเอชดี ที่สามารถเปลี่ยนแปลงไปตามฟังก์ชันที่ต้องการใช้งานได้ แอลกอริทึมที่ใช้ในการคำนวณมีรูปแบบการดังนี้ คือ

$$\Delta MV_N = \frac{100}{PB} \left\{ \Delta PV_N + \frac{\Delta T}{T_I} E_N + T_D \frac{\Delta(\Delta PV_N)}{\Delta T} \right\}$$

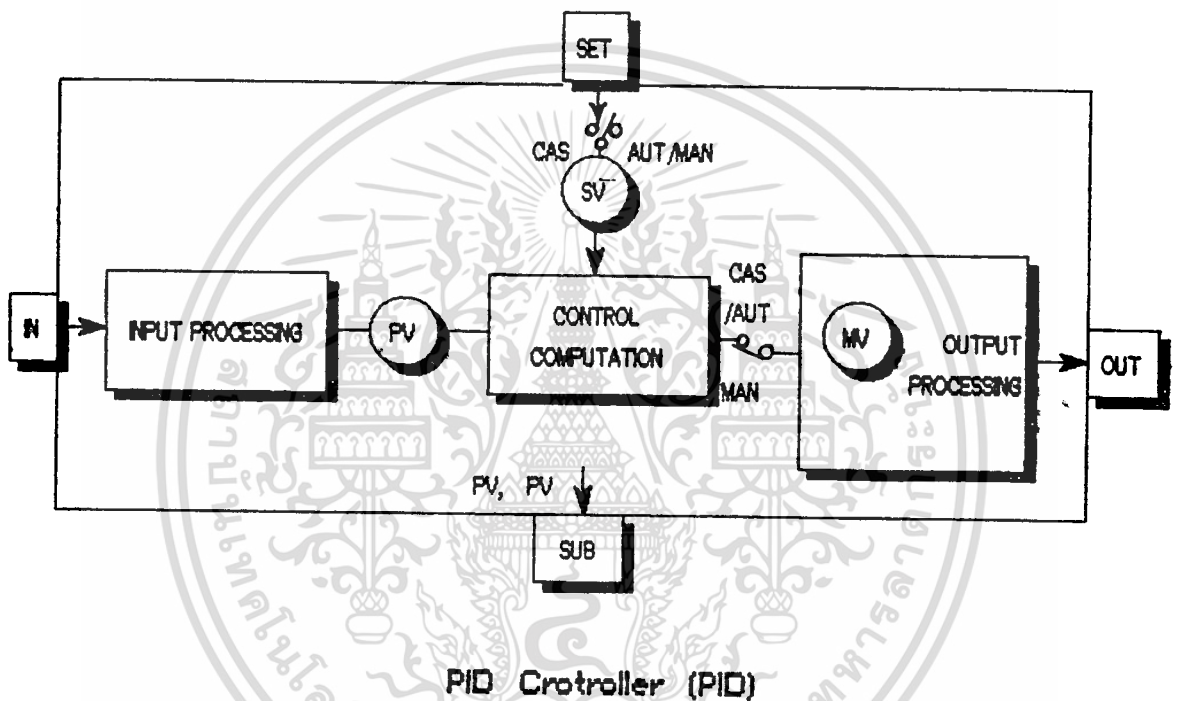
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อกำหนดค่าให้

- ΔMV_N : ค่าเอาต์พุตควบคุมที่เปลี่ยนแปลง
- $E_N = SV_N - PV_N$: ค่าที่ผิดผลาดจากค่าที่กำหนดไว้ (พิจารณาจากการย้อนกลับ)
- $\Delta E_N = E_N - E_{N-1}$: ค่าผิดผลาดที่เปลี่ยนแปลงไป,
- ΔT เป็นคาบเวลาของการควบคุม
- ΔPV_N : ค่าอินพุตที่เปลี่ยนแปลงไป
- PB, T_I, T_D : ค่าพารามิเตอร์พีไอดี ได้แก่ พรอพเพอร์ชันแนลเบนด์,
อินทิกรัลไทม์, แควรี่เวทไทม์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 แสดงบทบาทการควบคุมแบบพีไอดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Baud Rate

Test

System

File

9600
4800
2400
1200

AUD RAT 9600

1 F 2 3 4

รูปที่ 4.1 รายการสำหรับการทดสอบแรก

Baud Rate

Test

System

File

No. 1
No. 2
No. 3
No. 4

AUD RAT 9600

1 F 2 3 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ **รูปที่ 4.2 รายการสำหรับการทดสอบย่อย** จึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Baud Rate

Test

System

File

Setpoint

No.1:	45
No.2:	0
No.3:	0
No.4:	0

ENTER NUMBER OF SET-POINT[0-255] : 89

AUD RAT 9600

1 F 2 3 4

รูปที่ 4.3 รายการสำหรับการตั้งค่าเซตพอยท์

Baud Rate

Test

System

File

Setpoint
Parameter

No.1:Kp	2
:Ti	4
:Td	1
No.2:Kp	0
:Ti	45
:Td	0
No.3:Kp	0
:Ti	0
:Td	0
No.4:Kp	0
:Ti	0
:Td	0

ENTER NUMBER OF PID [0-255] : 56

AUD RAT 9600

1 F 2 3 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 4.4 รายการสำหรับการตั้งค่าพารามิเตอร์พีไอดี
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามเผยแพร่ต่อผู้อื่น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Baud Rate

Test

System

File

Load
Save
OS shell
Quit

ENTER LOAD FILE NAME : odd

BAUD RAT 9600

1 F 2 3 4

รูปที่ 4.5 รายการสำหรับการโหลดไฟล์

Baud Rate

Test

System

File

Load
Save
OS shell
Quit

CANNOT OPEN FILE NAME: odd

BAUD RAT 9600

1 F 2 3 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งนี้ รูปที่ 4.6 รายการเมื่อทำการโหลดไฟล์ไม่สำเร็จ ของเอกสารทุกครั้งที่มีการนำไปใช้

Baud Rate

Test

System

File

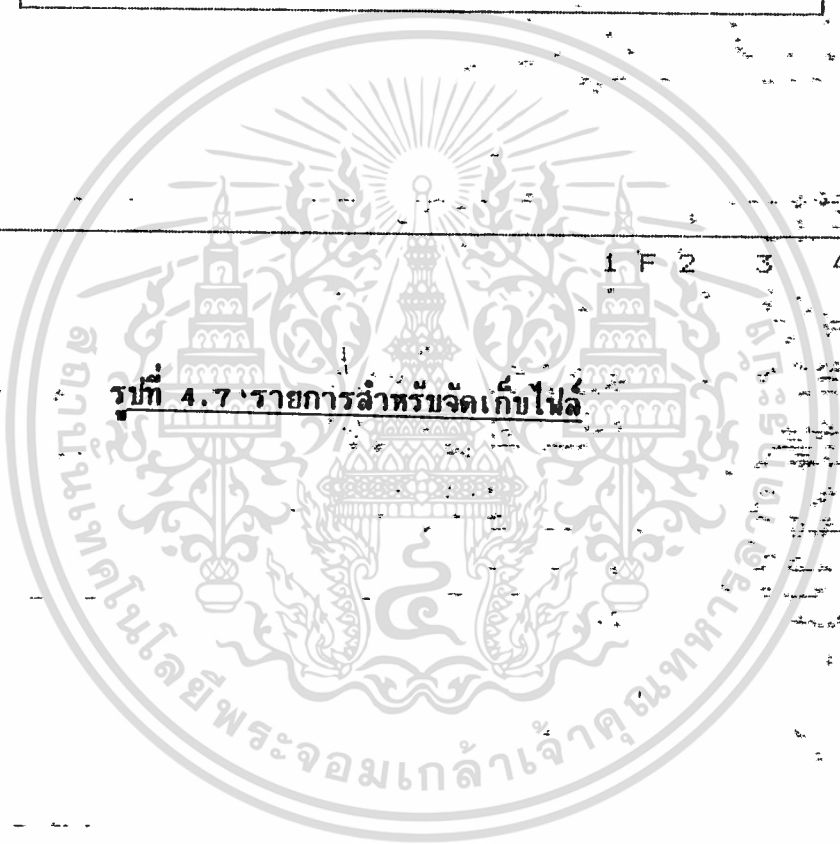
Load
Save
OS shell
Quit

ENTER SAVE FILE NAME : ood

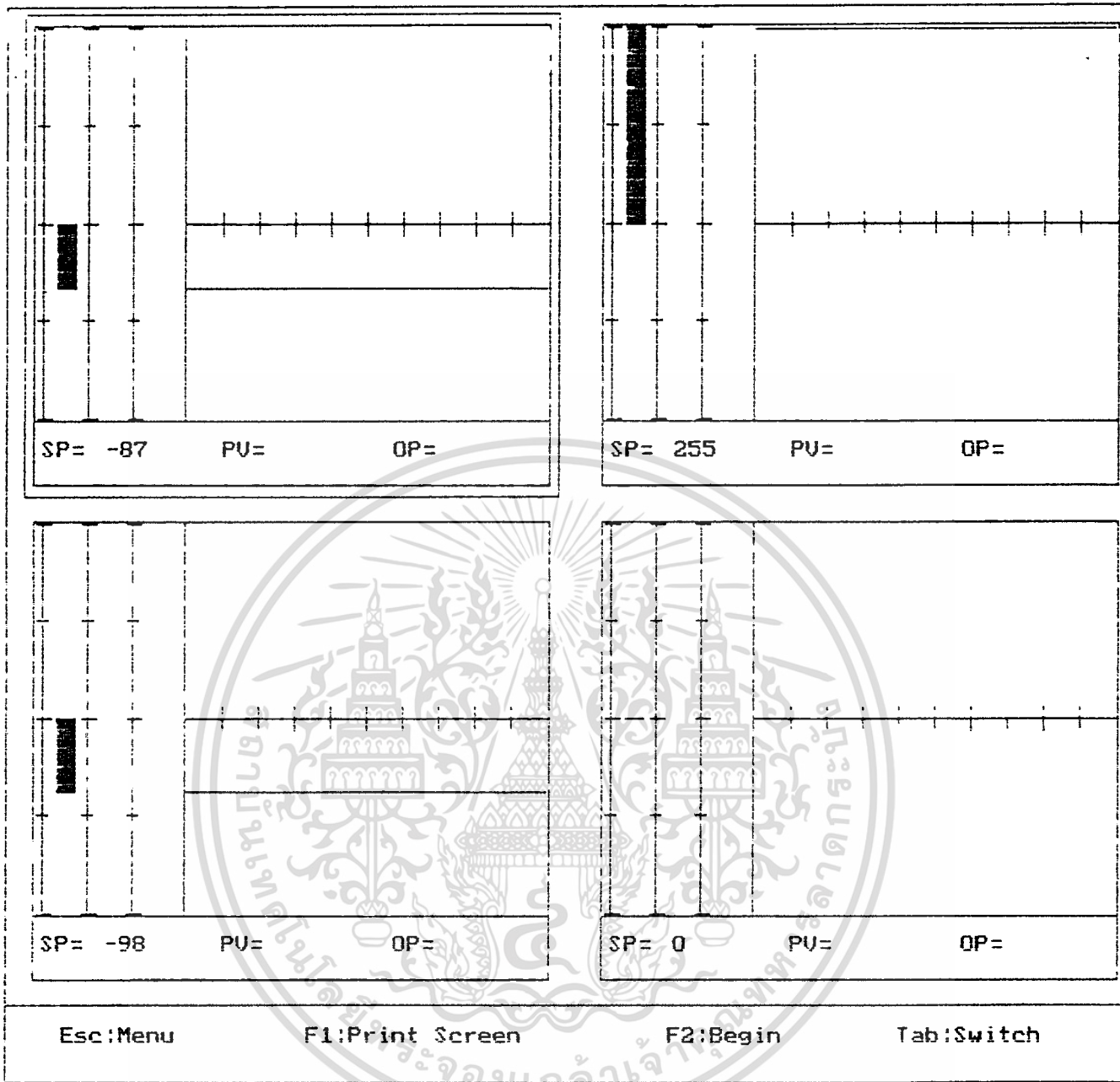
AUD RAT 9600

1 F 2 3 4

รูปที่ 4.7 รายการสำหรับจัดเก็บไฟล์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 ภาพแสดงผลการควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุป และ วิจารณ์

1. สามารถทำการติดต่อสื่อสาร ระหว่าง ไมโครคอนโทรลเลอร์เอ็มซีเอส 8031 กับไมโครคอมพิวเตอร์ ผ่านทางพอร์ตสื่อสารแบบอนุกรม อาร์เอส 232ซี ได้ และสามารถนำมาใช้ประโยชน์ในงานควบคุมได้อย่างมีประสิทธิภาพ ตามลักษณะโปรแกรมที่เขียนขึ้น เพื่อที่จะได้นำมาประยุกต์ใช้กับงานควบคุมลักษณะต่างๆ ได้อย่างเหมาะสม

2. ระบบการควบคุมแบบแยกส่วน มีความเหมาะสมกับระบบที่มีหน่วยงานย่อยเป็นจำนวนมาก เนื่องจากการควบคุมเป็นไปโดยอัตโนมัติ และ รายงาน ผลสภาวะได้ตลอดเวลา เท่าที่ผู้ควบคุมต้องการทราบ โดยเฉพาะอย่างยิ่ง เป็นการควบคุมผ่านทางไมโครคอมพิวเตอร์ สามารถติดต่อกับตัวคอนโทรลเลอร์ได้เป็นจำนวนมาก สัญญาณที่ใช้ควบคุมอาจจะอยู่ในรูปของกระแส หรือ แรงดันก็ได้

3. สำหรับฮาร์ดแวร์ของระบบการควบคุมแบบแยกส่วน สร้างโดยใช้ในไมโครคอนโทรลเลอร์ 8031 เป็นตัวควบคุมการทำงานของระบบ โดยออกแบบให้วงจรมีหน่วยความจำโปรแกรมภายนอกสำรองไว้เป็นจำนวน 8 กิโลไบต์ เพื่อไว้ใช้สำหรับการพัฒนางานต่อไปในอนาคต ทั้งนี้ได้ทำการขยายพอร์ตใช้งานโดยใช้ 8255 เป็นพอร์ตรับส่งค่าต่างๆ ที่ใช้งานควบคุม

โครงการชิ้นนี้ได้ทำการควบคุมลักษณะนี้ มาใช้ทำการควบคุมอัตราเร็วของมอเตอร์กระแสตรง เพราะเนื่องจากในโรงงานอุตสาหกรรมได้ใช้ประโยชน์ของมอเตอร์กระแสตรงในหลายด้าน

4. สำหรับงานในส่วนซอฟต์แวร์ จะต้องมีกำหนดเงื่อนไขในการรับส่งสัญญาณกันระหว่างส่วนไมโครคอมพิวเตอร์ กับตัวคอนโทรลเลอร์ พร้อมทั้งต้องมีการรายงาน

ผลให้แก่ผู้ควบคุมทราบได้ตลอดเวลาที่ต้องการ ข้อมูลทุกอย่างของระบบจะถูกควบคุมให้เป็นไปตามค่าที่กำหนดคร่อมมีการบ้อนค่าในขบวนการกลับมาตรวจสอบตลอดเวลา หรือ เรียกว่าเป็น

การควบคุมแบบมีการบ้อนกลับ การแสดงผลของขบวนการก็จะมีรายการปรากฏบนหน้าจอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไมโครคอมพิวเตอร์ให้เลือก และมีการแสดงกราฟของขบวนการที่กำลังปฏิบัติงานอยู่
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุป และ วิจารณ์

1. สามารถหาการคิดต่อสื่อสาร ระหว่าง โมเด็มคอนโทรลเลอร์เอ็มซีเอส 8031 กับโมเด็มคอมพิวเตอร์ ผ่านทางพอร์ตสื่อสารแบบอนุกรม อาร์เอส 232ซี ได้ และสามารถนำมาใช้ประโยชน์ในงานควบคุมได้อย่างมีประสิทธิภาพ ตามลักษณะโปรแกรมที่เขียนขึ้น เพื่อที่จะได้นำมาประยุกต์ใช้กับงานควบคุมลักษณะต่างๆ ได้อย่างเหมาะสม

2. ระบบการควบคุมแบบแยกส่วน มีความเหมาะสมกับระบบที่มีหน่วยงานย่อยเป็นจำนวนมาก เนื่องจากการควบคุมเป็นไปโดยอัตโนมัติ และ รายงาน ผลสถานะได้ตลอดเวลาเท่าที่ผู้ควบคุมต้องการทราบ โดยเฉพาะอย่างยิ่ง เป็นการควบคุมผ่านทางโมเด็มคอมพิวเตอร์ สามารถติดต่อกับตัวคอนโทรลเลอร์ได้เป็นจำนวนมาก สัญญาณที่ใช้ควบคุมอาจจะอยู่ในรูปของกระแส หรือ แรงดันก็ได้

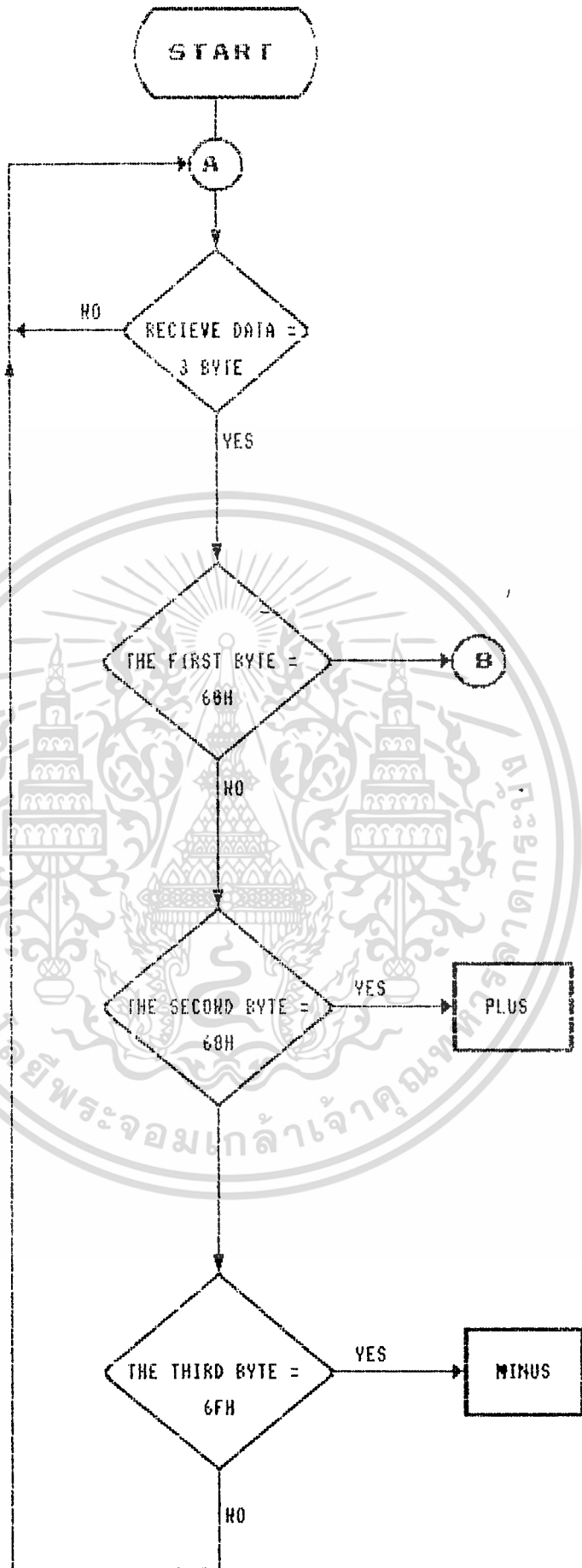
3. สำหรับฮาร์ดแวร์ของระบบการควบคุมแบบแยกส่วน สร้างโดยใช้โมเด็มคอนโทรลเลอร์ 8031 เป็นตัวควบคุมการทำงานของระบบ โดยออกแบบให้วงจรมีหน่วยความจำโปรแกรมภายนอกสำรองไว้เป็นจำนวน 8 กิโลไบต์ เพื่อไว้ใช้สำหรับการพัฒนางานต่อไปในอนาคต ทั้งนี้ได้ทำการขยายพอร์ตทำงานโดยใช้ 8255 เป็นพอร์ตรับส่งค่าต่างๆ ที่ใช้ในงานควบคุม

โครงการงานชิ้นนี้ได้ดำเนินการควบคุมลักษณะนี้ มาใช้ทำการควบคุมอัตราเร็วของมอเตอร์กระแสตรง เพราะเนื่องจากในโรงงานอุตสาหกรรมได้ใช้ประโยชน์ของมอเตอร์กระแสตรงในหลาย ๆ ด้าน

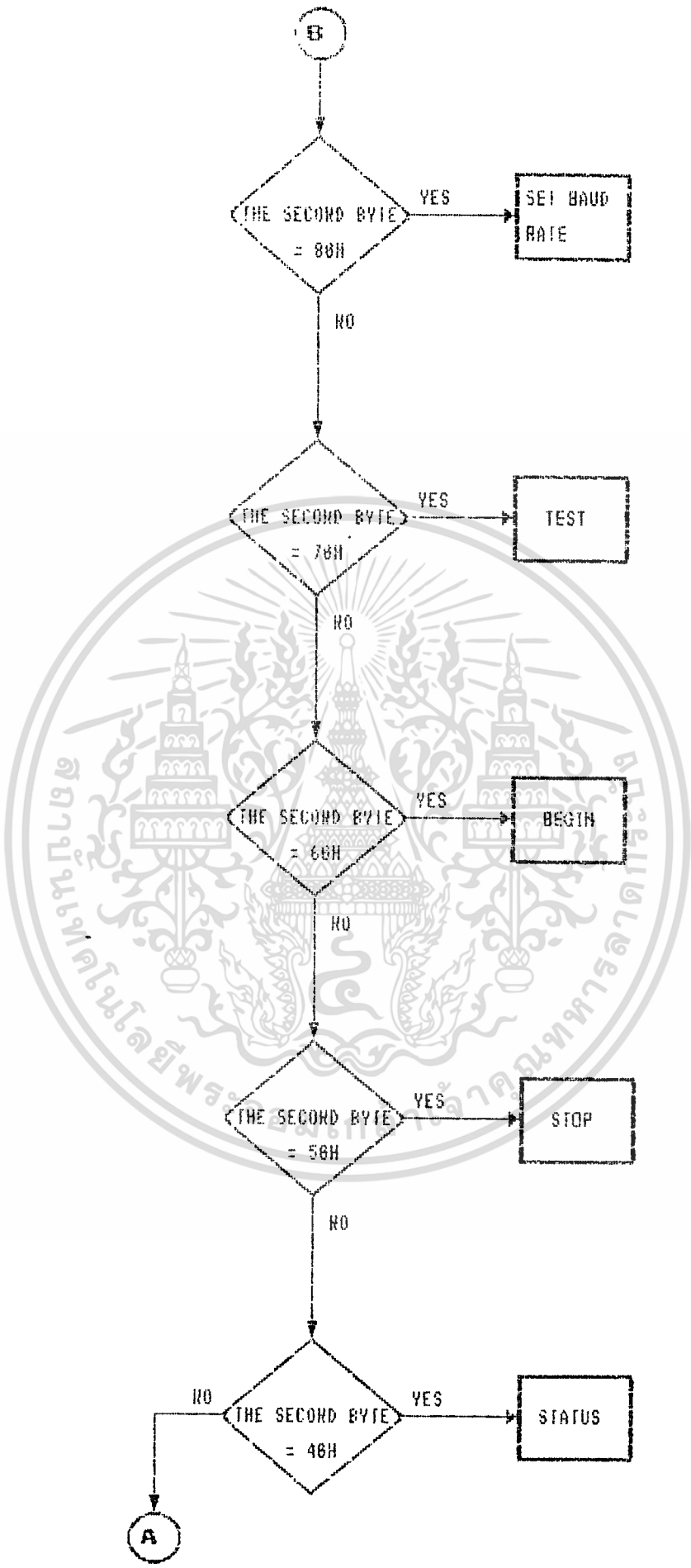
4. สำหรับงานในส่วนซอฟต์แวร์ จะต้องมีกำหนดเงื่อนไขในการรับส่งสัญญาณกันระหว่างส่วนโมเด็มคอมพิวเตอร์ กับตัวคอนโทรลเลอร์ หรือทั้งนี้ต้องมีารรายงานผลให้แก่ผู้ควบคุมทราบได้ตลอดเวลาที่ต้องการ ข้อมูลทุกอย่างของระบบจะถูกควบคุมให้เป็นไปตามค่าที่กำหนดโดยมีการป้อนค่าในขบวนการกลับมาตรวจสอบตลอดเวลา หรือเรียกว่าเป็นการควบคุมแบบมีการป้อนกลับ การแสดงผลของขบวนการก็จะมีรายการปรากฏบนหน้าจอ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามเผยแพร่ต่อสาธารณะโดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CPU"8051.TBL"
HOF"INT8"

: MCS-51 REGISTER ADDRESS :

A: EQU 0E0H
PSW: EQU 0D0H
IP: EQU 0B8H
IE: EQU 0A8H
SBUF: EQU 9FH
SCON: EQU 98H
TH1: EQU 8DH
PCON: EQU 87H
SP: EQU 81H
P1: EQU 90H
TMOD: EQU 89H
TCON: EQU 88H
B: EQU 0F0H
TLO: EQU 8AH
TH0: EQU 8CH
DPH: EQU 83H
DPL: EQU 82H

: MCS-51 BIT ADDRESS :

TI: EQU 99H
RI: EQU 98H
AO: EQU 0E0H
A1: EQU 0E1H
A2: EQU 0E2H
A3: EQU 0E3H
A4: EQU 0E4H
A5: EQU 0E5H
A6: EQU 0E6H
A7: EQU 0E7H
TF0: EQU 8DH
TR0: EQU 8CH
CY: EQU 0D7H
P3.3: EQU 0B3H

ORG 0000H
LJMP INIT

INIT: ORG 0030H
MOV PSW,#00H
MOV SP,#50H
MOV SCON,#50H
MOV PCON,#00H
MOV IE,#92H
MOV IP,#10H
MOV TH1,#0E8H
MOV TMOD,#21H
MOV TCON,#40H
MOV 33H,#68H
MOV DPTR,#8000H
MOV A,#00H

;ENABLE SERIAL & TIMERO INTERRUPT
;SERIAL INTERRUPT PRIORITY
;BAUD RATE = 1200
;USE TIMERO IN MODE1

;68H = 01101000B
;ADDR.8000H = PORT A

MOVX @DPTR,A

MOV R6,#0FFH

;WAIT FOR 8255

D8255: MOV R5,#0FFH

AGAIN: DJNZ R5,AGAIN

DJNZ R6,D8255

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของสำนักงานคณะกรรมการการอุดมศึกษา ห้ามเผยแพร่โดยไม่ได้รับอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่าในรูปแบบใดก็ตาม การนำเอกสารนี้ไปเผยแพร่โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV A,#00H
CJNE A,#32H,INIT ;CHANGE #02 - #32 [2]

HAL: MOV P1,#55H

NEW: MOV R0,#02H ;R0 = COUNTER
MOV R1,#30H ;FILL DATA AT ADDR.30H
MOV IE,#00H ;DISABLE INTERRUPT

START: JBC RI,PROLL
LJMP START
PROLL: MOV A,SBUF
MOV @R1,A
INC R1

WAIT: JBC RI,GET
SJMP WAIT
GET: MOV A,SBUF
MOV @R1,A
INC R1
DJNZ R0,WAIT

MOV DPTR,#8003H ;ADDR.8003H IS CONTROL PORT
MOV A,#90H ;CONTROL WORD [ONLY PORT A
; IS I\N\ PORT]
MOVX @DPTR,A

MOV A,30H
CJNE A,#60H,TOPLUS ;#60H = (01100000)B

SETBAUD: MOV A,31H
CJNE A,#80H,TEST
MOV A,32H
JB A0,BAUD1 ;REG.A = #01H
JB A1,BAUD2 ;REG.A = #02H
JB A2,BAUD4 ;REG.A = #04H
JB A3,BAUD9 ;REG.A = #08H
LJMP BAUD1

BAUD9: MOV TH1,#0FDH
MOV P1,#01H
LJMP NEW

BAUD4: MOV TH1,#0FAH
MOV A,#02H
MOV P1,A
LJMP NEW

BAUD2: MOV TH1,#0F4H
MOV A,#04H
MOV P1,A
LJMP NEW

BAUD1: MOV TH1,#0E8H
MOV A,#08H
MOV P1,A
LJMP NEW

TEST: MOV A,31H
CJNE A,#70H,BEGIN
MOV A,32H
CJNE A,#01H,TONEW ;#01H = SLAVE NO.
SETB P3.3
CLR P3.3

```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ทำไปประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV   DFTR,#8000H
MOVX  A,@DFTR           ;READ PROCESS VALUE FROM A/D
MOV   34H,A             ;STORE P.V.

ANS:  MOV   A,33H       ;AT ADDR.33H HAS (01101000)B
                                ;OR (01101111)B
      MOV   SBUF,A
TRAN1: JBC  TI,NEXT1    ;CHECK BIT TI
      SJMP TRAN1
NEXT1: MOV   A,32H       ;AT ADDR. #32H = SLAVE NO.
      MOV   SBUF,A
TRAN2: JBC  TI,NEXT8    ;CHECK BIT TI
      SJMP TRAN2
NEXT8: MOV   A,34H       ;AT ADDR. 34H = P.V.
      MOV   SBUF,A

LASB: JBC  TI,TONEW
      SJMP LASB

TOPLUS: LJMP PLUS

BEGIN: MOV   A,31H
      CJNE A,#60H,STOP
      MOV   A,35H
      JNB  A0,TONEW
      MOV   R6,#10H
      AGN: MOV   R5,#0FFH
      AGN1: MOV   R7,#0FFH
      AGN2: DJNZ R7,AGN2
            DJNZ R5,AGN1
            DJNZ R6,AGN

      SETB P3.3
      CLR  P3.3
      MOV   DFTR,#8000H
      MOVX  A,@DFTR           ;READ PROCESS VALUE FROM A/D
      MOV   34H,A             ;STORE P.V.

DO:   MOV   A,33H
      MOV   SBUF,A
      MOV   R5,#7FH           ;DELAY
REPT3: MOV   R6,#0FFH
DELY2: DJNZ R6,DELY2
      DJNZ R5,REPT3
TRAN3: JBC  TI,NEXT3    ;CHECK BIT TI
      SJMP TRAN3
NEXT3: MOV   A,32H
      MOV   SBUF,A
TRAN4: JBC  TI,NEXT9    ;CHECK BIT TI
      SJMP TRAN4
NEXT9: MOV   A,34H
      MOV   SBUF,A
LASB0: JBC  TI,TONEW
      SJMP LASB0

STOP: MOV   A,31H
      CJNE A,#50H,STATUS
TONEW: LJMP NEW

STATUS: MOV   A,31H
        CJNE A,#40H,TONEW
        MOV   35H,32H           ;STORE STATUS VALUE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งนี้ ขอสงวนสิทธิ์ในเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PLUS: MOV A,30H
      CJNE A,#68H,MINUS          ;#68H = (01101000)B
      MOV 33H,A
      MOV A,31H
      CJNE A,#01,WORK
      MOV DPTR,#8001H           ;ADDR.8001H = PORT B
      MOV A,00H
      MOVX @DPTR,A
      MOV A,32H
      MOV P1,A
WORK:  MOV A,35H
      JNB A0,TONEW
      SETB P3.3
      CLR P3.3
      MOV DPTR,#8000H
      MOVX A,@DPTR              ;READ PROCESS VALUE FROM A/D
      MOV 34H,A                 ;STORE P.V.

```

```

      MOV A,33H
      MOV SBUF,A
      MOV R5,#7FH              ;DELAY
REPT7: MOV R6,#OFFH
DELY7: DJNZ R6,DELY7
      DJNZ R5,REPT7

```

```

TRAN7: JBC TI,NEXT7            ;CHECK BIT TI
      SJMP TRAN7
NEXT7:  MOV A,31H
      MOV SBUF,A

```

```

TRAN10: JBC TI,NEXT10         ;CHECK BIT TI
      SJMP TRAN10
NEXT10: MOV A,34H
      MOV SBUF,A
LASB1:  JBC TI,TONEW
      SJMP LASB1

```

```

MINUS: MOV A,30H
      CJNE A,#6FH,TONEW       ;#6FH = (01101111)B
      MOV 33H,A
      MOV A,31H
      CJNE A,#01,WORK1
      MOV P1,#00H
      MOV DPTR,#8001H         ;ADDR.8001H = PORT B
      MOV A,32H
      MOVX @DPTR,A

```

```

WORK1: MOV A,35H
      JNB A0,TONEW
      SETB P3.3
      CLR P3.3
      MOV DPTR,#8000H
      MOVX A,@DPTR           ;READ PROCESS VALUE FROM A/D
      MOV 34H,A              ;STORE P.V.

```

```

      MOV A,33H
      MOV SBUF,A
      MOV R5,#7FH              ;DELAY
REPT11: MOV R6,#OFFH
DELY11: DJNZ R6,DELY11
      DJNZ R5,REPT11

```

เอกสาร DELY11: การทำงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามเผยแพร่ต่อแหล่งอื่น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

TRAN11: JBC TI,OUT1          ;CHECK BIT TI
      SJMP TRAN11

```

NEW2: LJMP NEW

OUT1: MOV A,31H
TRAN8: JBC TI,OUTB
 SJMP TRAN8
OUTB: MOV A,34H
 MOV SBUF,A
LASB2: JBC TI,NEW2
 SJMP LASB2

RETI

; SUBROUTINE FOR TIMERO INTERRUPT ;

ORG 000BH
LJMP CHECKTO

ORG 0100H
CHECKTO: JBC TFO,TIME
 SJMP CHECKTO

TIME: CLR TRO
 MOV PSW,#00H
 PUSH 01 ;PUSH R1 OF BANK0
 PUSH A
 PUSH TH1
 PUSH SBUF

 MOV A,#00H
 ORL A,RO
 JNZ TONEW
 POP SBUF
 POP TH1
 POP TH0
 POP A
 POP 01
 RETI

; SUBROUTINE FOR SERIAL INTERRUPT ;

ORG 0023H
LJMP NEW

END

```

#include<graphics.h>
#include<bios.h>
#include<stdlib.h>
#include<stdio.h>
#include<dos.h>

#define UP 0x4800
#define DOWN 0x5000
#define LEFT 0x4b00
#define RIGHT 0x4d00
#define ENTER 0x1c0d
#define ESC 0x11b
#define Tab 0xf09
#define F1 0x3b00
#define F2 0x3c00
#define F3 0x3d00
#define F4 0x3e00
#define F5 0x3f00
#define F6 0x4000
#define F7 0x4100
#define F8 0x4200
#define F9 0x4300
#define F10 0x4400
#define START 0x02

char *menu_msg[] =
{"Baud Rate", " Test ", " System ", " File "};
char *menu0_msg[] =
{" 9600 ", " 4800 ", " 2400 ", " 1200 "};
char *menu1_msg[] =
{" No. 1 ", " No. 2 ", " No. 3 ", " No. 4 "};
char *menu2_msg[] =
{"Setpoint ", "Parameter", "Run "};
char *menu3_msg[] =
{"Load ", "Save ", "OS shell ", "Quit "};
char *B_msg[12];
int LbaUd=1200; /*SET begin BAUD RAT*/
int NEW_PV[4];
int OLD_Delta_PV[4];
int OLD_PV[4];
int OLD_OP[4];
int SIGN[4];
int SET[4];
int PID[12];
unsigned char STATE_TEST=0x00;
int Sub_Time;
int GanXSP[4];
int GanYSP[4];
int GanXPV[4];
int GanYPV[4];
int OLDTime;

void Fram(int, int, int);
void SetCom1(int);
void ShowMonitor(int);
void cursor_on();
void cursor_off();
void high(int, int, int, int);
void low(int, int, int, int);
void box(int, int, int, int);
int TextMenu(void);
void InCom1(void);

```

```

void Bselect_menu(int,int,int);
int Aselect_menu(int,int);
void TeSt_Com1(int);
void Block_Active(int,int);
void BAR_Setpoint(int,float,int);
void DRAW_OUT(int);
void RUN_PROCESS(void);
void SET_POINT(int,int);
int SET_SETPOINT(int,int,int);
void SET_PID(int,int);
void SET_SETPID(int,int,int);
void hardcopy(int);
void CopyDot(int);
void copybyte(int);
void Draw_Bar_GRA(int,int);
void Load(void);
void Save(void);
void CHANGE_STATE(int,int);
void OUT_STATE(void);
int BEGIN(void);
void END(void);
int PID_ALGORITHM(int);

```

```

/*****

```

```

main()

```

```

{
int n;

```

```

STATE_TEST=0xff;

```

```

SET[0] = 0; /* set point*/

```

```

SET[1] = 0; /* set point*/

```

```

SET[2] = 0; /* set point*/

```

```

SET[3] = 0; /* set point*/

```

```

NEW_PV[0] = 0;

```

```

NEW_PV[1] = 0;

```

```

NEW_PV[2] = 0;

```

```

NEW_PV[3] = 0;

```

```

OLD_Delta_PV[0] = 0;

```

```

OLD_Delta_PV[1] = 0;

```

```

OLD_Delta_PV[2] = 0;

```

```

OLD_Delta_PV[3] = 0;

```

```

OLD_PV[0] = 0;

```

```

OLD_PV[1] = 0;

```

```

OLD_PV[2] = 0;

```

```

OLD_PV[3] = 0;

```

```

Sub_Time = 1;

```

```

OLD_OP[0] = 0;

```

```

OLD_OP[1] = 0;

```

```

OLD_OP[2] = 0;

```

```

OLD_OP[3] = 0;

```

```

PID[0]=0;PID[1]=0;PID[2]=0;PID[3]=0;PID[4]=0;PID[5]=0;

```

```

PID[6]=0;PID[7]=0;PID[8]=0;PID[9]=0;PID[10]=0;PID[11]=0;

```

```

n=TextMenu();

```

```

Bselect_menu(10,4,n);

```

```

return (0);

```

```

} /*end main */

```

```

/*****

```

```

void RUN_PROCESS(void)

```

```

{

```

```

int driver,mode;

```

```

int      vivi;
char     S_TA_TUS;
struct   time timer;
int      NEWTime;
int      DetTaTime;
int      Pe,mama;
int      i,x,y;
int      IN;
char     CONDITION;
int      Buffer_Set;

```

```

mode     = 0;
driver   = DETECT;

```

```

initgraph(&driver,&mode,"");
line(5,5,5,479-5);
line(5,479-5,639-5,479-5);
line(639-5,5,639-5,479-5);
line(639-5,5,5,5);
line(5,440,639-5,440);

```

```

OUT_STATE();
ShowMonitor(0);
ShowMonitor(1);
ShowMonitor(2);
ShowMonitor(3);
Draw_Bar_GRA(0,1);
Draw_Bar_GRA(1,1);
Draw_Bar_GRA(2,1);
Draw_Bar_GRA(3,1);
outtextxy(20,450,
"   Esc:Menu           F1:Print Screen           F2:Begin           Tab:

```

```

S_TA_TUS=0;
Block_Active(S_TA_TUS,1);
if(SIGN[S_TA_TUS]==1) /*SING=1(+)*
    Buffer_Set=SET[S_TA_TUS];
else Buffer_Set=-SET[S_TA_TUS];
inportb(0x3f8);
vivi=BEGIN();
if(vivi){
    bar(195,195,445,255);
    bar3d(205,205,435,245,0,0);
    outtextxy(220,223,"SYSTEM FAIL PRESS ANY KEY");
    getch();
    closegraph();
    return;}

```

```

gettime(&timer);
OLDTime=(timer.ti_sec)*10+(timer.ti_hund/10);
for(CONDITION=1;CONDITION;)
{
if((inportb(0x3fd)&1)==1)InCom1();

```

```

gettime(&timer);
/* gotoxy(2,25);
printf("sec = %d ",timer.ti_sec); */
NEWTime=(timer.ti_sec)*10+(timer.ti_hund/10);
if(NEWTime>=OLDTime)DetTaTime=NEWTime-OLDTime;
else DetTaTime=NEWTime+600-OLDTime;
/* gotoxy(2,18);
printf("DetTaTime = %d ",DetTaTime);*/
OLDTime=NEWTime;
for(i=0;i<4;i++)

```

```

{
moveto(GanXSP[i],GanYSP[i]);
if(SIGN[i]==1){Pe=SET[i];mama=NEW_PV[i];}
  else {Pe=-SET[i];mama=-NEW_PV[i];}
switch(i){
  case 0: x=20 ;y=13 ;break;
  case 1: x=335;y=13 ;break;
  case 2: x=20 ;y=229;break;
  case 3: x=335;y=229;break;}
GanXSP[i]+=DetTaTime;
GanYSP[i]=y+86-(Pe*85/256);

setcolor(YELLOW);
if(GanXSP[i]<x+285)
{lineto(GanXSP[i],GanYSP[i]);
 setcolor(RED);
 moveto(GanXPV[i],GanYPV[i]);
 GanYPV[i]=y+86-(mama*85/256);
 GanXPV[i]=GanXSP[i];
 lineto(GanXSP[i],GanYPV[i]);}
/* gotoxy(2,20);
 printf("Odd = %d ",NEWTime); */
}
IN=bioskey(1);
switch(IN){

case UP: bioskey(0);
Buffer_Set++;
if(Buffer_Set>255)Buffer_Set=255;
BAR_Setpoint(S_TA_TUS,Buffer_Set,1);
SET[S_TA_TUS]=abs(Buffer_Set);
if(Buffer_Set>=0)SIGN[S_TA_TUS]=1;
delay(900);
break;

case DOWN: bioskey(0);
Buffer_Set--;
if(Buffer_Set<-255)Buffer_Set=-255;
BAR_Setpoint(S_TA_TUS,Buffer_Set,1);
SET[S_TA_TUS]=abs(Buffer_Set);
if(Buffer_Set<0)SIGN[S_TA_TUS]=0;
delay(900);
break;

case Tab : bioskey(0);
Block_Active(S_TA_TUS,0);
S_TA_TUS++;
if(S_TA_TUS>3)S_TA_TUS=0;
Block_Active(S_TA_TUS,1);
if(SIGN[S_TA_TUS]==1)
Buffer_Set=SET[S_TA_TUS];
else Buffer_Set=-SET[S_TA_TUS];
break;

case F1 : bioskey(0);
Draw_Bar_GRA(0,0);
Draw_Bar_GRA(1,0);
Draw_Bar_GRA(2,0);
Draw_Bar_GRA(3,0);เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
END();
hardcopy(4); และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
Draw_Bar_GRA(0,1);
Draw_Bar_GRA(1,1);

```

```

        Draw_Bar_GRA(2,1);
        Draw_Bar_GRA(3,1);
        ShowMonitor(0);
        ShowMonitor(1);
        ShowMonitor(2);
        ShowMonitor(3);
        inportb(0x3f8);
        vivi=BEGIN();
        if(vivi){bar(195,195,445,255);
                bar3d(205,205,435,245,0,0);
        outtextxy(220,223,"SYSTEM FAIL PRESS ANY KEY");
                getch();
                closegraph();
                return;}
        gettime(&timer);
        OLDTime=(timer.ti_sec)*10+(timer.ti_hund/10);
        break;

    case F2      : bioskey(0);
        ShowMonitor(0);
        ShowMonitor(1);
        ShowMonitor(2);
        ShowMonitor(3);
        inportb(0x3f8);
        vivi=BEGIN();
        if(vivi){bar(195,195,445,255);
                bar3d(205,205,435,245,0,0);
        outtextxy(220,223,"SYSTEM FAIL PRESS ANY KEY");
                getch();
                closegraph();
                return;}
        gettime(&timer);
        OLDTime=(timer.ti_sec)*10+(timer.ti_hund/10);
        break;

    case ENTER: bioskey(0);
        vivi=BEGIN();
        if(vivi){bar(195,195,445,255);
                bar3d(205,205,435,245,0,0);
        outtextxy(220,223,"SYSTEM FAIL PRESS ANY KEY");
                getch();
                closegraph();
                return;}

        break;

    case ESC: bioskey(0);
        closegraph();
        CONDITION=0;
        END();

    case 0: break;

    default : bioskey(0);

```

```

} /*end switch*/
} /*end for */
}

```

```

/*****
void DRAW_OUT(int STATUS)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

โดยไม่ได้รับอนุญาต ยกเว้นหากมีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int x,y;
int Pe;
char *TEXT_OP = "test";

```

```

switch(STATUS){
case 0: x=20 ;y=13 ;break;
case 1: x=335;y=13 ;break;
case 2: x=20 ;y=229;break;
case 3: x=335;y=229;break;}

if(SIGN[STATUS]==1)Pe=OLD_OP[STATUS];
else Pe=-OLD_OP[STATUS];
setfillstyle(1,CYAN);
bar(x+65,1+y,x+75,171+y);
bar(230+x,175+y,270+x,195+y); /*del OP*/
setfillstyle(1,RED);
bar(65+x,y+86-(Pe*85/256),75+x,y+86);
itoa(Pe,TEXT_OP,10);
outtextxy(235+x,181+y,TEXT_OP);
}

```

```

/*****
void BAR_Setpoint(int STATUS,float SP,int color)
{

```

```

int x,y;
char *TEXT_SP = "test";
if(color==0)
setfillstyle(1,BLACK);
else setfillstyle(1,CYAN);

switch(STATUS){
case 0: x=20 ;y=13 ;break;
case 1: x=335;y=13 ;break;
case 2: x=20 ;y=229;break;
case 3: x=335;y=229;break;}
bar(13+x,171+y,23+x,1+y);
bar(x+35,y+175,x+75,y+195); /*Del set point*/
itoa(SP,TEXT_SP,10);
setcolor(YELLOW);
outtextxy(x+40,y+181,TEXT_SP);
setfillstyle(1,RED);
bar(13+x,y+86-(SP*85/256),23+x,y+86);
}

```

```

/*****

```

```

int PID_ALGORITHM(int n)
{
int Num=4*n;
int DubBle_DelTa_PV;
int DeViATion;
int New_DelTa_PV;
int DelTa_OP;

```

```

New_DelTa_PV = NEW_PV[n] - OLD_PV[n];
DeViATion = SET[n] - NEW_PV[n];
DubBle_DelTa_PV = New_DelTa_PV - OLD_DelTa_PV[n];
OLD_PV[n] = NEW_PV[n];
OLD_DelTa_PV[n] = New_DelTa_PV;

```

```

DelTa_OP=(100/PID[Num])*( New_DelTa_PV+
((Sub_Time*DeViATion)/PID[Num+1])+
((PID[Num+2]*DubBle_DelTa_PV)/Sub_Time)
);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น การนำออกหรือเผยแพร่โดยไม่ได้รับอนุญาตจะถือว่าผิดกฎหมาย

```

OLD_OP[n]+=DelTa_OP;
if(OLD_OP[n]>255) OLD_OP[n]=255;
if(OLD_OP[n]<0) OLD_OP[n]=0;

```

```
return(OLD_OP[n]);
}
```

```
void Block_Active(int STATUS,int ON_OFF)
{
int x,y;
switch(STATUS){
case 0: x=20 ;y=13 ;break;
case 1: x=335;y=13 ;break;
case 2: x=20 ;y=229;break;
case 3: x=335;y=229;break;}
switch(ON_OFF){
case 0: setcolor(BLACK) ; break;
case 1: setcolor(YELLOW); break; }
line(x-5 ,y-5 ,x+291,y-5 );
line(x+291,y-5 ,x+291,y+205);
line(x+291,y+205,x-5 ,y+205);
line(x-5 ,y+205,x-5 ,y-5 );
}

void ShowMonitor(int STATUS)
{
float CPF;
int Pe,mama;
int X,Y;
switch(STATUS){
case 0: X=20 ;Y=13 ;break;
case 1: X=335;Y=13 ;break;
case 2: X=20 ;Y=229;break;
case 3: X=335;Y=229;break;}
setcolor(YELLOW);
setfillstyle(1,BLACK); /* set bk GRAP */
bar3d(84+X,Y,286+X,172+Y,0,0);
line(X,172+Y,286+X,172+Y);
line(85+X,86+Y,285+X,86+Y);
setcolor(RED);
for(CPF=1;CPF<10;CPF++)
line(85+X+(300-100)*CPF/10,91+Y,85+X+(300-100)*CPF/10,81+Y);
if(SIGN[STATUS]==1) {Pe=SET[STATUS];mama=NEW_PV[STATUS];}
else {Pe=-SET[STATUS];mama=NEW_PV[STATUS];}
GanXSP[STATUS]=X+85;
GanXPV[STATUS]=X+85;
GanYSP[STATUS]=Y+86-(Pe*85/256);
GanYPV[STATUS]=Y+86-(mama*85/256);
}

void Draw_Bar_GRA(int STATUS,int color)
{
int X,Y;
int Pe;

if(color==0)
setfillstyle(1,BLACK);
else setfillstyle(1,CYAN);

switch(STATUS){
case 0: X=20 ;Y=13 ;break;
case 1: X=335;Y=13 ;break;
case 2: X=20 ;Y=229;break;
case 3: X=335;Y=229;break;}

```

```
bar(X+1,Y+1,X+83,Y+171);
bar(X+1,Y+173,X+284,Y+199);
```

```
setcolor(YELLOW);
line(X,Y,X,200+Y);
line(X,Y,286+X,Y);
line(X,200+Y,286+X,200+Y);
line(286+X,Y,286+X,200+Y);
```

```
setcolor(WHITE);
Fram(5+X,1+Y,170); /* same */
outtextxy(5+X,181+Y,"SP=");
Fram(30+X,1+Y,170);
outtextxy(105+X,181+Y,"PV=");
Fram(55+X,1+Y,170);
outtextxy(200+X,181+Y,"OP=");
if(SIGN[STATUS]==1) {Pe=SET[STATUS];}
else {Pe=-SET[STATUS];}
```

```
BAR_Setpoint(STATUS,Pe,color);
}
```

```
/******
```

```
void Fram(int x,int y,int h)
{line(x,y,x,y+h);
 line(x-3,y,x+5,y);
 line(x-3,y+h,x+5,y+h);
 line(x-3,y+h/2,x+5,y+h/2);
 line(x-3,y+h/4,x+3,y+h/4);
 line(x-3,y+3*h/4,x+3,y+3*h/4);
}
```

```
/******
```

```
void SetCom1(int BaUd)
{ int BauDLo;
 int BauDHi;
 int BauDOut;
 switch(BaUd){
 case 9600: BauDLo=0x0c; BauDHi=0x00; BauDOut=0x08; break;
 case 4800: BauDLo=0x18; BauDHi=0x00; BauDOut=0x04; break;
 case 2400: BauDLo=0x30; BauDHi=0x00; BauDOut=0x02; break;
 case 1200: BauDLo=0x60; BauDHi=0x00; BauDOut=0x01; break; }
LBaUd=BaUd;
```

```
outportb(0x3f8,0x60); /*out set slave*/
delay(8);
outportb(0x3f8,0x80); /*constant set baud rat*/
delay(8);
outportb(0x3f8,BauDOut);
delay(8);
```

```
outportb(0x3fb,0x80); /*set com1*/
outportb(0x3f8,BauDLo); /*set baud lo*/
outportb(0x3f9,BauDHi); /*set baud hi*/
outportb(0x3fb,0x03); /*8 bit,1 stop,no parity*/
```

```
inportb(0x3f8);
gotoxy(12,25);
textbackground(3);
textcolor(15);
```

```
cprintf("ที่ส่ง %d",BaUd);}
```

เอกสารนี้สงวนลิขสิทธิ์ไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

```
/******
void TeSt_Com1(int SLAVE)
```

```

{
int i;
int REDATA[3];

inportb(0x3f8);
  REDATA[0]=0xffff;
outportb(0x3f8,0x60);
  delay(8);
  REDATA[1]=0xffff;
outportb(0x3f8,0x70);
  delay(8);
  REDATA[2]=0xffff;
outportb(0x3f8,SLAVE);

for(i=0;i<10000;i++)
  if((inportb(0x3fd)&1)==1)
    {REDATA[0]=inportb(0x3f8);
      i=10005;}
for(i=0;i<2000;i++)
  if((inportb(0x3fd)&1)==1)
    {REDATA[1]=inportb(0x3f8);
      i=2005;}
for(i=0;i<2000;i++)
  if((inportb(0x3fd)&1)==1)
    {REDATA[2]=inportb(0x3f8);
      i=2005;}

cprintf("Rx | %x | %x | %x |",REDATA[0],REDATA[1],REDATA[2]);
if(((REDATA[0]==0x68)||((REDATA[0]==0x6f)&&(REDATA[1]==SLAVE)
&&(REDATA[2]!=0xffff)))
  {NEW_PV[SLAVE-1]=REDATA[2];
  CHANGE_STATE(SLAVE,1);}
else {CHANGE_STATE(SLAVE,0);}
}
/*****/
void CHANGE_STATE(int SLAVE,int ENA)
{

textbackground(3);
textcolor(15);

if(ENA==1)
switch(SLAVE){
  case 1: gotoxy(52,25);cprintf("T");STATE_TEST|=0x01;br
  case 2: gotoxy(56,25);cprintf("T");STATE_TEST|=0x02;br
  case 3: gotoxy(60,25);cprintf("T");STATE_TEST|=0x04;br
  case 4: gotoxy(64,25);cprintf("T");STATE_TEST|=0x08;br
else
switch(SLAVE){
  case 1: gotoxy(52,25);cprintf("F");STATE_TEST&=0xfe;br
  case 2: gotoxy(56,25);cprintf("F");STATE_TEST&=0xfd;br
  case 3: gotoxy(60,25);cprintf("F");STATE_TEST&=0xfb;br
  case 4: gotoxy(64,25);cprintf("F");STATE_TEST&=0xf7;br
}
/*****/
void OUT_STATE(void)
{ outportb(0x3f8,0x60);
  delay(8);
  outportb(0x3f8,0x40);
  delay(8);
  outportb(0x3f8,STATE_TEST);
}
/*****/

```

```

int BEGIN(void)
{
unsigned char OSS;

if(STATE_TEST==0x00||STATE_TEST==0xf0)
return(1);
else if((STATE_TEST&0x01)==0x01)
{ OSS=0x01;}
else if((STATE_TEST&0x02)==0x02)
{ OSS=0x02;}
else if((STATE_TEST&0x04)==0x04)
{ OSS=0x03;}
else OSS=0x04;

outportb(0x3f8,0x60);
delay(8);
outportb(0x3f8,0x60);
delay(8);
outportb(0x3f8,OSS);
return(0);
}
/*****/
void END(void)
{ outportb(0x3f8,0x60);
delay(8);
outportb(0x3f8,0x50);
delay(8);
outportb(0x3f8,0x00);
}
/*****/
void InCom1(void)
{
int x,y;
char *TEXT_IP="text";
int i;
int, DatA[3]; /*inport com1*/
int OutPut;
int Pe;
int CON;
DatA[0]=inportb(0x3f8);
DatA[1]=0xffff;
DatA[2]=0xffff;
for(i=0;i<20000;i++)
if((inportb(0x3fd)&1)==1)
{DatA[1]=inportb(0x3f8);
i=20010;};
for(i=0;i<2000;i++)
if((inportb(0x3fd)&1)==1)
{DatA[2]=inportb(0x3f8);
i=2010;};
for(CON=0;CON>0;)
{gotoxy(10,10);
printf("Re = %x! %x! %x",DatA[0],DatA[1],DatA[2]);
if(((DatA[0]==0x68)|| (DatA[0]==0x6f))&&(0x01<=DatA[1]<=0x04))
{
switch(DatA[1]){
case 1: x=20 ;y=13 ;NEW_PV[0]=DatA[2];break;
case 2: x=335;y=13 ;NEW_PV[1]=DatA[2];break;
case 3: x=20 ;y=229;NEW_PV[2]=DatA[2];break;
case 4: x=335;y=229;NEW_PV[3]=DatA[2];break;}
setfillstyle(1,CYAN);
bar(38+x,171+y,48+x,1+y);
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง การนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย

```

bar(135+x,175+y,x+180,y+195);
setfillstyle(1,RED);
if(SIGN[Data[1]-1]==1){Pe=Data[2];Data[0]=0x68;}
    else {Pe=-Data[2];Data[0]=0x6f;}
bar(38+x,y+86-(Pe*85/256),48+x,y+86);
itoa(Pe,TEXT_IP,10);
setcolor(YELLOW);
outtextxy(x+140,y+181,TEXT_IP);
OutPut=PID_ALGORITHM(Data[1]-1);
DRAW_OUT(Data[1]-1);
outportb(0x3f8,Data[0]);
delay(8);
outportb(0x3f8,Data[1]);
delay(8);
outportb(0x3f8,OutPut);
CON=1;
}
else{ inportb(0x3f8);
    Data[0]=0xfff;
    outportb(0x3f8,0x60);
    delay(8);
    Data[1]=0xfff;
    outportb(0x3f8,0x30);
    delay(8);
    Data[2]=0xfff;
    outportb(0x3f8,0xf0);

    for(i=0;i<10000;i++)
        if((inportb(0x3fd)&1)==1)
            {Data[0]=inportb(0x3f8);
                i=10005;}
    for(i=0;i<2000;i++)
        if((inportb(0x3fd)&1)==1)
            {Data[1]=inportb(0x3f8);
                i=2005;}
    for(i=0;i<2000;i++)
        if((inportb(0x3fd)&1)==1)
            {Data[2]=inportb(0x3f8);
                i=2005;}}
}
}
}

```

/******

```
int TextMenu(void)
```

```
{
int i,x=10,y=1; /* x,y cursor position 10 3*/
int n,m;
int BaUd=1200;
```

```
textbackground(LIGHTBLUE); /*color BK menu*/
clrscr();
cursor_off();
textbackground(LIGHTGRAY); /*color Header menu*/
gotoxy(0,y);
cprintf("
");
```

```
textcolor(YELLOW);
```

```
for(i=0;i<4;i++)
```

```
{ gotoxy (x+(i*17),y);
```

```
cprintf(menu_msg[i]);}
```

```
textbackground(LIGHTBLUE); /*color BK menu*/
```

```
textcolor(15); /* darw BOX */
```

```
box(1,2,79,24); /* ----- */
```

```
outportb(0x3fb,0x80); /*set com1*/
```

```

outportb(0x3f8,0x60);          /*set baud lo*/
outportb(0x3f9,0x00);          /*set baud hi*/
outportb(0x3fb,0x03);          /*8 bit,1 stop,no parity*/
inportb(0x3f8);
gotoxy(3,25);
cprintf("BAUD RAT");
gotoxy(50,25);
cprintf("1 2 3 4");
gotoxy(12,25);
textbackground(3);
textcolor(15);
cprintf(" %d ",BaUd);
gotoxy(51,25);
cprintf(" ");
gotoxy(55,25);
cprintf(" ");
gotoxy(59,25);
cprintf(" ");
gotoxy(63,25);
cprintf(" ");
n=Aselect_menu(x,y);
return (n);
}
/*****
void SET_PID(int x,int y)
{
char *SETPID_msg[]={ " No.1:P ", " :I ", " :D ",
" No.2:P ", " :I ", " :D ",
" No.3:P ", " :I ", " :D ",
" No.4:P ", " :I ", " :D "};

char buf[80*4*2];
int i;
int k=12;
int m=0;
int in;
char CONDITION;

for(i=0;i<k;i++)
B_msg[i]=SETPID_msg[i];
textcolor(0);          /*draw BOX */
textbackground(7);
box(x-1,y-1,x+13,y+k); /*-----*/
textcolor(YELLOW);    /*set color msg*/
for(i=0;i<k;i++)
{ gotoxy (x,y+i);cprintf(B_msg[i]);cprintf("%3d ",PID[i]);}
high(m,x,y,1);cprintf("%3d ",PID[0]);
for(CONDITION=1;CONDITION;)
{ in=bioskey(0);
switch(in)
{case UP:m--;
if(m<0)
{m=k-1;
high(m,x,y,1);cprintf("%3d ",PID[m]);
m=0;
high(m,x,y,0);cprintf("%3d ",PID[m]);
m=k-1;}
else{high(m,x,y,1);cprintf("%3d ",PID[m]);
m++;
high(m,x,y,0);cprintf("%3d ",PID[m]);}
case DOWN:m++;
break;}
}
}

```

เอกสารนี้เป็นเอกสารที่ไม่ได้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if(m>k-1)
        {m=0;
        high(m,x,y,1);cprintf("%3d ",PID[m]);
        m=k-1;
        high(m,x,y,0);cprintf("%3d ",PID[m]);
        m=0; }
        else{high(m,x,y,1);cprintf("%3d ",PID[m]);
        m--;
        high(m,x,y,0);cprintf("%3d ",PID[m]);
        m++; }
        break;

```

```

case ENTER:gettext(15,15,51,17,buf);
        SET_SETPID(x,y,m);
        puttext(15,15,51,17,buf);
        break;

```

```

default :CONDITION=0;
        for(i=0;i<3;i++)
        B_msg[i]=menu2_msg[i];
        break;

```

```

}}}

```

/******

```

void SET_SETPID(int x,int y,int m)

```

```

{int new;
char inbuf[5];

```

```

cursor_on();
textcolor(BLACK);
textbackground(WHITE);
box(15,15,51,17);
gotoxy(16,16);
cprintf(" ");
gotoxy(16,16);
cprintf(" ENTER NUMBER OF PID [0-255] : ");
gets(inbuf);
sscanf(inbuf,"%d",&new);
if(new>255)new=255;
if(new<0)new=0;
PID[m]=new;
high(m,x,y,1);
cprintf("%3d ",PID[m]);
cursor_off();
}

```

/******

```

void SET_POINT(int x,int y)

```

```

{
char *SETPOINT_msg[] ={" No.1: "," No.2: "," No.3: "," No.4: "};
char buf[80*4*2];
int i;
int k=4;
int m=0;
int in;
int Pe[4];
char CONDITION;

```

```

for(i=0;i<k;i++)
B_msg[i]=SETPOINT_msg[i];

```

```

textcolor(0);/*draw BOX*/
textbackground(7);
box(x-1,y-1,x+14,y+k);
textcolor(YELLOW);
/*set color msg*/

```

เอกสารนี้เป็นส่วนหนึ่งของระบบงานเพื่อการศึกษาเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า
 ได้โดยไม่ได้รับอนุญาตจากทางผู้พัฒนา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(i=0;i<k;i++)
{ gotoxy (x,y+i);
cprintf(B_msg[i]);
if(SIGN[i]==1)Pe[i]=SET[i];
  else Pe[i]=-SET[i];
cprintf("%4d ",Pe[i]);}
high(m,x,y,1);cprintf("%4d ",Pe[0]);
for(CONDITION=1;CONDITION;)
{ in=bioskey(0);
switch(in)
{case  UP:m--;
  if(m<0)
  {m=k-1;
  high(m,x,y,1);cprintf("%4d ",Pe[m]);
  m=0;
  high(m,x,y,0);cprintf("%4d ",Pe[m]);
  m=k-1;}
  else{high(m,x,y,1);cprintf("%4d ",Pe[m]);
  m++;
  high(m,x,y,0);cprintf("%4d ",Pe[m]);
  m--;}
  break;
case  DOWN:m++;
  if(m>k-1)
  {m=0;
  high(m,x,y,1);cprintf("%4d ",Pe[m]);
  m=k-1;
  high(m,x,y,0);cprintf("%4d ",Pe[m]);
  m=0; }
  else{high(m,x,y,1);cprintf("%4d ",Pe[m]);
  m--;
  high(m,x,y,0);cprintf("%4d ",Pe[m]);
  m++;}
  break;
case ENTER:gettext(1,15,80,17,buf);
  Pe[m]=SET_SETPOINT(x,y,m);
  puttext(1,15,80,17,buf);
  break;
default  :CONDITION=0;
  for(i=0;i<3;i++)
  B_msg[i]=menu2_msg[i];
  break;
}}}
/*****
int SET_SETPOINT(int x,int y,int m)
{int new;
char inbuf[5];

cursor_on();
textcolor(BLACK);
textbackground(WHITE);
box(19,15,63,17);
gotoxy(20,16);
cprintf("
gotoxy(20,16);
cprintf("  ENTER NUMBER OF SET-POINT[0-255] : ");
gets(inbuf);
sscanf(inbuf,"%d",&new);
if(new>255)new=255;
if(new<-255)new=-255;

```

```

if(new>=0)SIGN[m]=1;
else SIGN[m]=0;
SET[m]=abs(new);
high(m,x,y,1);
cprintf("%4d ",new);
cursor_off();
return(new);
}

```

```

/*****

```

```

int Aselect_menu(int gx,int gy)

```

```

{

```

```

int n=0,in,i;

```

```

low(n,gx,gy,1);

```

```

for(;;)

```

```

{ in=bioskey(0);

```

```

switch(in)

```

```

{case LEFT:n--;

```

```

if(n<0)

```

```

{ n=3;

```

```

low(n,gx,gy,1);

```

```

n=0;

```

```

low(n,gx,gy,0);

```

```

n=3;}

```

```

else

```

```

{ low(n,gx,gy,1);

```

```

n++;

```

```

low(n,gx,gy,0);

```

```

n--;}

```

```

break;

```

```

case RIGHT:n++;

```

```

if(n>3)

```

```

{ n=0;

```

```

low(n,gx,gy,1);

```

```

n=3;

```

```

low(n,gx,gy,0);

```

```

n=0; }

```

```

else

```

```

{ low(n,gx,gy,1);

```

```

n--;

```

```

low(n,gx,gy,0);

```

```

n++; }

```

```

break;

```

```

case ENTER:return n;

```

```

} /*switch*/

```

```

} /*for(;;)*/

```

```

}

```

```

/*emd f(x)*/

```

```

/*****

```

```

void Bselect_menu(int x,int y,int n)

```

```

{

```

```

int m=0,in,i,k,Xi;

```

```

char buf[80*25*2];

```

```

char buf1[80*23*2];

```

```

gettext(1,2,80,23,buf1);

```

```

loop1: Xi=x+(17*n);

```

```

m=0;

```

```

textbackground(LIGHTGRAY);

```

```

textcolor(YELLOW);

```

```

switch(n)

```

```

{case 0:k=4;

```

```

for(i=0;i<k;i++)

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้าม B_msg[i]=menu0_msg[i]; จำของเอกสารทุกครั้งที่มีการนำไปใช้

```

break;

```

```

case 1:k=4;
    for(i=0;i<k;i++)
        B_msg[i]=menu1_msg[i];
    break;
case 2:k=3;
    for(i=0;i<k;i++)
        B_msg[i]=menu2_msg[i];
    break;
case 3:k=4;
    for(i=0;i<k;i++)
        B_msg[i]=menu3_msg[i];
    break;}
textcolor(WHITE);
box(Xi-1,y-1,Xi+9,y+k);
textcolor(YELLOW);
for(i=0;i<k;i++)
    { gotoxy (Xi,y+i);
      cprintf(B_msg[i]);}
high(m,Xi,y,1);
for(;;)
    { in=bioskey(0);
      switch(in)
        {case UP:m--;
          if(m<0)
            {m=k-1;
             high(m,Xi,y,1);
             m=0;
             high(m,Xi,y,0);
             m=k-1;}
          else{high(m,Xi,y,1);
              m++;
              high(m,Xi,y,0);
              m--;}
          break;
        case DOWN:m++;
          if(m>k-1)
            {m=0;
             high(m,Xi,y,1);
             m=k-1;
             high(m,Xi,y,0);
             m=0; }
          else{high(m,Xi,y,1);
              m--;
              high(m,Xi,y,0);
              m++; }
          break;
        case LEFT:
          puttext(1,2,80,23,buf1);
          low(n,x,y-3,0);
          n--;
          if(n<0)n=3;
          low(n,x,y-3,1);
          goto loop1;
        case RIGHT:
          puttext(1,2,80,23,buf1);
          low(n,x,y-3,0);
          n++;
          if(n>3)n=0;
          low(n,x,y-3,1);
          goto loop1;
        case ENTER:
          if(n==0)
            switch(m)
              {case 0:SetCom1(9600);break;
               case 1:SetCom1(4800);break;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา
 งานนี้ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 งานนี้ ไม่อนุญาตให้นำไปใช้

```

        case 2:SetCom1(2400);break;
        case 3:SetCom1(1200);break;}
    else if(n==1)
    switch(m)
    {case 0:gotoxy(13,9);TeSt_Com1(1);break;
    case 1:gotoxy(13,9);TeSt_Com1(2);break;
    case 2:gotoxy(13,9);TeSt_Com1(3);break;
    case 3:gotoxy(13,9);TeSt_Com1(4);break;}
    else if(n==2)
    switch(m)
    {case 0: gettext(1,1,80,25,buf);
        SET_POINT(45,6);
        puttext(1,1,80,25,buf);
        break;
        case 1: gettext(1,1,80,25,buf);
        SET_PID(45,7);
        puttext(1,1,80,25,buf);
        cursor_off();
        break;
        case 2: gettext(1,1,80,25,buf);
        RUN_PROCESS();
        puttext(1,1,80,25,buf);
        cursor_off();
        break;}
    else if(n==3)
    switch(m)
    {case 2:gettext(1,1,80,25,buf);
    textbackground(BLACK);
    clrscr();
    printf("\nType EXIT to return to DCS1 . . .");
    system("command");
    puttext(1,1,80,25,buf);
    break;
    case 0:gettext(1,1,80,25,buf);
    Load();
    puttext(1,1,80,25,buf);
    break;
    case 1:gettext(1,1,80,25,buf);
    Save();
    puttext(1,1,80,25,buf);
    break;
    case 3:textbackground(BLACK);
    clrscr();exit(0);}
    } /*switch*/
} /*for(;;)*/
} /*end f(x)*/
/*****/
void Save(void)
{
FILE *fp;
char inname[30];
int bal[25];

cursor_on();
textcolor(BLACK);
textbackground(WHITE);
box(19,15,61,17);
gotoxy(20,16);
cprintf("ที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้");
gotoxy(20,16);
cprintf("สน ENTER SAVE FILE NAME");
gets(inname);

```

เอกสารนี้เป็นทรัพย์สินของมหาวิทยาลัยพระจอมเกล้าธนบุรี การนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย

```

cursor_off();

if((fp=fopen(inname,"wb"))==NULL)
{
    sound(1400);
    textbackground(RED);
    box(19,15,61,17);
    gotoxy(20,16);
    cprintf("
    gotoxy(20,16);
    printf("Cannot open file name: %s\n",inname);
    delay(250);
    nosound();
    getch();
    return;}

```

```

bal[0]=0x5314;    bal[1]=SET[0];    bal[2]=SET[1];
bal[3]=SET[2];    bal[4]=SET[3];    bal[5]=PID[0];
bal[6]=PID[1];    bal[7]=PID[2];    bal[8]=PID[3];
bal[9]=PID[4];    bal[10]=PID[5];    bal[11]=PID[6];
bal[12]=PID[7];    bal[13]=PID[8];    bal[14]=PID[9];
bal[15]=PID[10];    bal[16]=PID[11];    bal[17]=SIGN[0];
bal[18]=SIGN[1];    bal[19]=SIGN[2];    bal[20]=SIGN[3];

```

```

fwrite(bal,sizeof(int),25,fp);
fclose(fp);
}

```

/******
void Load(void)
{
FILE *fp;
char inname[30];
int bal[25];

```

cursor_on();
textcolor(BLACK);
textbackground(WHITE);
box(19,15,61,17);
gotoxy(20,16);
cprintf("
gotoxy(20,16);
cprintf(" ENTER LOAD FILE NAME : ");
gets(inname);
cursor_off();
if((fp=fopen(inname,"rb"))==NULL)
{
    sound(1400);
    textbackground(RED);
    box(19,15,61,17);
    gotoxy(20,16);
    cprintf("
    gotoxy(20,16);
    cprintf("CANNOT OPEN FILE NAME: %s\n",inname);
    delay(250);
    nosound();
    getch();
    return;}
if(fread(bal,sizeof(int),25,fp)!=25)
if(feof(fp))
cprintf("end-of-file");
else

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ cprintf("end-of-file"); และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
else

```

        cprintf("file read error");
    }
fclose(fp);
if(bal[0]!=0x5314)
{SET[0]=bal[1];
SET[2]=bal[3];
PID[0]=bal[5];    PID[1]=bal[6];
PID[3]=bal[8];    PID[4]=bal[9];
PID[6]=bal[11];   PID[7]=bal[12];
PID[9]=bal[14];   PID[10]=bal[15];
SIGN[0]=bal[17];  SIGN[1]=bal[18];
SIGN[3]=bal[20];}
SET[1]=bal[2];
SET[3]=bal[4];
PID[2]=bal[7];
PID[5]=bal[10];
PID[8]=bal[13];
PID[11]=bal[16];
SIGN[2]=bal[19];
else
{
    sound(700);
    textbackground(YELLOW);
    box(19,15,61,17);
    gotoxy(20,16);
    cprintf("
");
    gotoxy(20,16);
    cprintf("FORMAT ERROR IN FILE: %s\n",inname);
    delay(250);
    nosound();
    getch();}
}
/*****/
void low(int n,int x,int y,int h1)
{textmode(3);    /* c80*/
if(h1==1)
{ textcolor(128);    /*blink*/
textbackground(RED);
gotoxy(x+(n*17),y);
cprintf(menu_msg[n]);}
else { gotoxy(x+(n*17),y);
textbackground(LIGHTGRAY);
textcolor(YELLOW);
cprintf(menu_msg[n]);}
}
/*****/
void high(int m,int x,int y,int h1)
{textmode(3);    /* c80*/
if(h1==1)
{ textcolor(128);    /*blink*/
textbackground(GREEN);    /*green*/
gotoxy(x,y+m);
cprintf(B_msg[m]);}
else { gotoxy(x,y+m);
textbackground(LIGHTGRAY);
textcolor(YELLOW);
cprintf(B_msg[m]);}
}
/*****/
void cursor_off()
{union REGS reg;
reg.h. ch=0x20; /* turn on bit 5*/
reg.h. cl=0;
reg.h. ah=1;
int86(0x10,&reg,&reg);
}
/*****/
void cursor_on()
{union REGS reg;
reg.h. ch=0x00; /* turn on bit 5*/

```

```

reg.h. cl=13;
reg.h. ah=1;
int86(0x10,&reg,&reg);
}
/*****
void box(int xul,int yul,int xlr,int ylr)
{
    int i;
    char hz=196;
    char vt=179;
    char ul=218;
    char lr=217;
    char ll=192;
    char ur=191;
    gotoxy(xul,yul);
    for(i=xul;i<=xlr;i++)
        putchar(hz);
    gotoxy(xul,ylr);
    for(i=xul;i<=xlr;i++)
        putchar(hz);
    for(i=yul;i<=ylr;i++)
        {gotoxy(xul,i);
        putchar(vt);
        gotoxy(xlr,i);
        putchar(vt);}
    gotoxy(xul,yul); putchar(ul);
    gotoxy(xlr,yul); putchar(ur);
    gotoxy(xlr,ylr); putchar(lr);
    gotoxy(xul,ylr); putchar(ll);
}
/*****
void copybyte(int byte)
{
    biosprint(0,byte,0);
}
/*****
void CopyDot(int scan)
{
    int Bits[8] = {128,64,32,16,8,4,2,1};
    int Bit,PrintByte,maxx,xpix,n1,n2,exit;
    maxx=getmaxx()+1;
    do
    {
        PrintByte=0;maxx=maxx--;exit=27;
        for(Bit=0;Bit<=7;Bit++)
        {
            if(getpixel(maxx,scan+Bit)!=0)
                {PrintByte=PrintByte+Bits[Bit];}
        }
        exit=PrintByte;
        if(maxx==0){exit=27;}
    }
    while (exit==0);
    if(PrintByte!=0)
    { maxx=maxx++;
      n2=maxx/256; n1=maxx-n2*256;
      copybyte(27); copybyte(51); copybyte(24);
      copybyte(27); copybyte(42); copybyte(6);
      copybyte(n1); copybyte(n2);
      for(xpix=0;xpix<=maxx;xpix++)
      { PrintByte=0;
        for(Bit=0;Bit<=7;Bit++)

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ไม่ว่ากรณีใดๆทั้งสิ้น ห้ามอวดอ้างถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

{

```

if (getpixel(xpix,scan+Bit)!=0)
  {PrintByte=PrintByte+Bits[Bit];}
  }
  copybyte(PrintByte);
}
}
}
/*****
void hardcopy(int left)
{
int scanline;
  copybyte(27);  copybyte(108);  copybyte(left);
  scanline=0;
  do
  {
    CopyDot(scanline);  scanline=scanline+8;
    copybyte(10);
  }
  while(scanline<(getmaxy()+1));
  copybyte(27);  copybyte(2);
}
*****/

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงการ และ ปรวิณานิพนธ์นี้เื่ออาจจะที่จะสำเร็จลงได้ ตามวัตถุประสงค์ ถ้าหาก
ขาดความร่วมมือ ความแนะนำ และกาสั่งใจ ในนามของผู้จัดทำจึงต้องขอขอบคุณแก่ผู้ที่ให้
ความร่วมมือทุกฝ่าย อันได้แก่ อาจารย์ เพื่อนๆ ที่ได้ช่วยหาความคิดเห็น และที่สำคัญที่
สุดขอแสดงความขอบพระคุณ ต่ออาจารย์ ชนิษฐา แซ่ตั้ง และ คุณ ณรงค์ เร เชียงแสน
ที่ได้เสียสละ เวลาและให้ความช่วยเหลือในทุกด้าน และหวังว่าความดีที่ได้จากปรวิณ
านิพนธ์ จะได้ขอมอบกลับไปยัง พ่อแม่ ที่ได้ให้ปัจจัยที่จำเป็นทุกอย่าง ต่อการศึกษา
ตลอดจนคุณครู อาจารย์ ทุกท่านที่มอบความรู้ให้ ตั้งแต่ชั้นพื้นฐาน จนถึงระดับมหาวิทยาลัย
จึงขอแสดงความขอบคุณมา ณ. ที่นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. A.Buxbaum, A.Straughen, K.schierau , "Design of Control Systems for DC Drives", Prentice-Hall International Inc., 1990
2. "Data Acquisition Linear Device", National Semiconductor
3. Kenneth J. Ayala, "The 8051 Microcontroller", West Publishing Company, 1991
4. "MCS-51 Microcontroller Data Book", ETT Company, Thailand
5. "Technical Reference Peronal Computer XT", IBM Company, 1983
6. "The MicroXL MFCN Field Control Unit Manual", First Edition, 1989

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้