ภาควิชาครุศาสตร์วิศวกรรม

คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อ ชุดปฏิบัติการทดลองไมโครคอนโทรลเลอร์ PIC 16F84A

PIC 16F84A Microcontroller Training Board

ชื่อนักศึกษา

1. นางสาวจันจิรา	เงินรวง	รหัสประจำตัว	43035368
2. นางสาวสุธาสินี	สว่างศรี	รหัสประจำตัว	43035393
3. นายสุภสรรณ์	เศิกศิริ	รหัสประจำตัว	43035395

หลักสูตร ครุศาสตร์อุตสาหกรรมบัณฑิต **สาขาวิชา** อิเล็กทรอนิกส์และคอมพิวเตอร์

อาจารย์ที่ปรึกษา อาจารย์สุรพงษ์ สิริพงศ์ดี

อาจารย์ที่ปรึกษาร่วม อาจารย์ปิยะ ศุภวราสุวัฒน์

คณะกรรมการสอบปริญญานิพนธ์		ลายมือชื่อ
1. อาจารย์สุรพงษ์	สิริพงศ์ดี	.....
2. อาจารย์ปิยะ	ศุภวราสุวัฒน์	.....
3. อาจารย์โกศล	ตราชู	.....
4. อาจารย์สุชิน	อาจหาญ	.....
5. อาจารย์ไพบูลย์	พวงวงศ์ตระกูล	.....

วัน/เดือน/ปีที่สอบ วันพฤหัสบดีที่ 1 พฤศจิกายน พ.ศ. 2544 เวลา 15.00 น.

สถานที่สอบ ห้อง ค.311 คณะครุศาสตร์อุตสาหกรรม สจล.

ภาควิชารับรองแล้ว

ลงนาม.....

(ผศ.วิสุทธิ์ อธิพรธรรม)

หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

วันที่ 16 เดือน พ.ย. พ.ศ. 2544

<BT4402022>

ชุดปฏิบัติการทดลองไมโครคอนโทรลเลอร์ PIC 16F84A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ปริญญานิพนธ์

ชุดปฏิบัติการทดลองไมโครคอนโทรลเลอร์ PIC16F84A

PIC16F84A MICROCONTROLLER TRAINING BOARD



นางสาวจันจิรา เงินรวง
นางสาวสุธาสินี สว่างศรี
นายสุภสรรณ์ เศิกศิริ

เลขหมู่.....
เลขทะเบียน 43163
วัน, เดือน, ปี 23 ก.ค. 2545

.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตร์อุตสาหกรรมบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์
ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
b11230624

ปริญญานิพนธ์

เรื่อง ชุดปฏิบัติการทดลองไมโครคอนโทรลเลอร์ PIC16F84A

PIC16F84A Microcontroller Training Board

วัตถุประสงค์

1. เพื่อศึกษาการทำงานของไมโครคอนโทรลเลอร์ ตระกูล PIC16F84A
2. เพื่อออกแบบวงจรและใบงานการทดลอง ที่ใช้ในการทดลองการทำงานของโมดูล
3. เพื่อสร้างชุดปฏิบัติการทดลองไมโครคอนโทรลเลอร์ PIC16F84A โดยแยกเป็นโมดูลและใบงานการทดลอง
4. เพื่อสามารถนำชุดปฏิบัติการทดลองไมโครคอนโทรลเลอร์ PIC16F84A และใบงานการทดลองไปใช้ประกอบการเรียนการสอน

ประโยชน์ที่คาดว่าจะได้รับ

1. มีความรู้และความเข้าใจหลักการทำงานของไมโครคอนโทรลเลอร์ตระกูล PIC16F84A
2. สามารถออกแบบชุดปฏิบัติการทดลองไมโครคอนโทรลเลอร์ PIC16F84A และใบงานการทดลองได้
3. สามารถสร้างชุดปฏิบัติการทดลองไมโครคอนโทรลเลอร์ PIC16F84A และใบงานการทดลองได้
4. สามารถนำชุดปฏิบัติการทดลองไมโครคอนโทรลเลอร์ PIC16F84A และใบงานการทดลองไปใช้ประกอบการเรียนการสอนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	ชุดปฏิบัติการทดลองไมโครคอนโทรลเลอร์ PIC16F84A
นักศึกษา	นางสาวจันจิรา เงินรวง
	นางสาวสุธาสินี สว่างศรี
	นายสุภสรรณ์ เศิกศิริ
อาจารย์ที่ปรึกษา	อาจารย์สุรพงษ์ สิริพงศ์ดี
อาจารย์ที่ปรึกษาร่วม	อาจารย์ปิยะ ศุภวราสุวัฒน์
หลักสูตร	ครุศาสตร์อุตสาหกรรมบัณฑิต
สาขาวิชา	อิเล็กทรอนิกส์และคอมพิวเตอร์
ปีการศึกษา	2544

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ได้นำเสนอการออกแบบและการสร้างชุดปฏิบัติการทดลองไมโครคอนโทรลเลอร์ PIC16F84A ที่สามารถโปรแกรมไฟล์ได้จากพอร์ตขนานของคอมพิวเตอร์ทั้งแบบ 18 ขา 28 ขา และ 40 ขา และยังสามารถต่อใช้งานชุดอุปกรณ์ทดลองต่างๆ ได้โดยแยกออกเป็นโมดูล ส่วนของโมดูลนั้นแบ่งออกเป็น 14 โมดูล คือ โมดูลหลัก PIC-ICSP โมดูลเครื่องมือวัด โมดูลอินพุตเอาต์พุตพื้นฐาน โมดูลแสดงผลแอลอีดีคอตเมตริกซ์ โมดูลแสดงผลตัวเลข 7 ส่วน โมดูลดิพสวิตช์และเมตริกซ์สวิตช์ โมดูลแสดงผลแบบผลึกเหลว โมดูลเชื่อมต่ออุปกรณ์ RTC EEPROM ลำโพง โมดูลแปลงระหว่างสัญญาณแอนะลอกและสัญญาณดิจิตอล โมดูลสื่อสารข้อมูลอนุกรม โมดูลขับดีซีมอเตอร์และสเตปปิ้งมอเตอร์ โมดูลโซลิดสเตทรีเลย์ โมดูลทดลองวงจร และโมดูลภาคจ่ายไฟ นอกจากนี้ได้ออกแบบโปรแกรมทำงานของชุดปฏิบัติการทดลอง จึงช่วยให้สามารถเข้าใจหลักการทำงาน และการใช้งานไมโครคอนโทรลเลอร์ตระกูล PIC16F84A ได้ง่ายขึ้น และยังสามารถนำไปใช้เป็นสื่อการเรียนการสอนในสถานศึกษาต่างๆได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	PIC16F84A Microcontroller Training Board	
Students	Miss Junjira	Ngoenruang
	Miss Sutasinee	Sawangsri
	Mr.Supasan	Serksiri
Advisor	Mr.Surapong	Siripongdee
Co-Advisor	Mr.Piya	Supavarasuwut
Education Level	Bachelor of Science in Industrial Education	
Program in	Electronics and Computer	
Academic	2001	

ABSTRACT

This thesis presents a designing and implementation of PIC16F84A Microcontroller Training Board. The project can be In-Circuit Serial Programming 18, 28 and 40 Pins devised. Into fourteen module; PIC-ICSP, Measurement, Basic Input Output, LED Dot Matrix Display, Seven Segment Display, Dip Switch and Matrix Switch, LCD Character Display, RTC, EEPROM Speaker Interfacing, Analog and Digital Converter, Serial Interfacing, DC Motor and Stepping Motor Driver, Solid State Relay, Circuit Experimental and Power Supply. The project helps the user to easier understand the principle of PIC16F84A Microcontroller Training Board and can be applied to use at institute of education.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ลุล่วงไปได้ด้วยดีอันเนื่องมาจากความร่วมมือของสมาชิกภายในกลุ่มทุกท่าน ขอขอบคุณอาจารย์สุรพงษ์ สิริพงศ์ดี และคณาจารย์ภาควิชาครุศาสตร์วิศวกรรมทุกท่านที่ให้ความอนุเคราะห์เครื่องมือ และอุปกรณ์ รวมทั้งให้คำแนะนำ แนวความคิด ความรู้ต่างๆ แนวทางแก้ไขปัญหาในการจัดทำปริญญานิพนธ์ ขอขอบคุณห้องสมุดคณะครุศาสตร์อุตสาหกรรม ห้องสมุดคณะวิศวกรรมศาสตร์ ที่ช่วยอำนวยความสะดวกและเอื้อเฟื้อสถานที่ในการค้นคว้าข้อมูล สุดท้ายที่ควรระลึกถึงอย่างยิ่ง บิดา และมารดาที่เป็นผู้ให้ความสนับสนุนด้านการศึกษา และเป็นผู้ให้กำลังใจด้วยดีตลอดมา ตั้งแต่อดีตจนถึงปัจจุบัน

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	XI
สารบัญรูป	XIII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์	1
1.2 ขีดความสามารถของโครงงาน	1
1.3 เนื้อหาโดยสังเขป	2
บทที่ 2 ทฤษฎีและหลักการ	4
2.1 ความรู้เบื้องต้นของไมโครคอนโทรลเลอร์ PIC16F8XX	4
2.1.1 คุณสมบัติทางเทคนิคของ PIC16F8XX	4
2.1.2 สถาปัตยกรรมของ PIC16F8XX	5
2.1.3 การจัดขาของ PIC16F8XX	7
2.1.4 การป้อนสัญญาณนาฬิกาให้แก่ PIC16F8XX	8
2.1.5 จังหวะสัญญาณนาฬิกาและไซเคิลการทำงานของ PIC16F8XX	8
2.2 การจัดการหน่วยความจำของ PIC16F8XX	10
2.2.1 หน่วยความจำโปรแกรม	10
2.2.2 หน่วยความจำข้อมูล	11
2.3 รีจิสเตอร์ควบคุมของ PIC16F8XX	12
2.3.1 รีจิสเตอร์ STATUS	12
2.3.2 รีจิสเตอร์ OPTION	13
2.3.3 รีจิสเตอร์ INTCON	15
2.3.4 รีจิสเตอร์โปรแกรมเคาน์เตอร์ PCL และ PCLATH	16
2.3.5 สแต็ก	17
2.3.6 การพุช	17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
2.3.7 รีจิสเตอร์ W	17
2.4 ชุดคำสั่งของ PIC16F86 และการเข้าถึงรีจิสเตอร์	18
2.4.1 กลุ่มคำสั่งจัดการข้อมูลระดับไบต์กับรีจิสเตอร์ไฟล์	18
2.4.2 กลุ่มคำสั่งจัดการข้อมูลระดับบิตกับรีจิสเตอร์ไฟล์	18
2.4.3 กลุ่มคำสั่งจัดการกับค่าคงที่และควบคุมการทำงาน	18
2.5 การเข้าถึงรีจิสเตอร์และข้อมูลของ PIC16F8XX	21
2.5.1 การเข้าถึงข้อมูลแบบทันทีทันใด	21
2.5.2 การเข้าถึงข้อมูลแบบโดยตรง	21
2.5.3 การเข้าถึงข้อมูลแบบโดยอ้อม	21
2.5.4 การเข้าถึงค่าข้อมูลแบบสัมพัทธ์	21
2.6 การใช้งาน Input / Output Port	22
2.6.1 พอร์ต A และรีจิสเตอร์ TRISA	22
2.6.2 พอร์ต B และรีจิสเตอร์ TRISB	22
2.6.3 พอร์ต C และรีจิสเตอร์ TRISC	23
2.7 ไทเมอร์เคาน์เตอร์ภายใน PIC16F8XX	23
2.7.1 การเลือกโหมดไทเมอร์หรือเคาน์เตอร์	23
2.7.2 การอินเตอร์รัพต์ของ TMRO	24
2.7.3 การใช้ TMRO กับสัญญาณนาฬิกาภายนอก	24
2.7.4 ผลทางเอาต์พุตของไทเมอร์เคาน์เตอร์	25
2.7.5 ปรีสเกลเลอร์	25
2.8 การใช้งาน EEPROM ภายใน	25
2.8.1 EECON1 และ EECON2	25
2.8.2 EEDATA	27
2.8.3 EEADR	27
2.9 คุณสมบัติพิเศษของซีพียู PIC16F8XX	27
2.9.1 การทำงานในโหมดสลีปหรือเพาเวอร์ดาวน์	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
2.9.2 การโปรแกรมแบบอนุกรมภายในวงจร	28
2.9.3 วอตซ์ด็อกไทเมอร์	28
2.9.4 คาบเวลาของวอตซ์ด็อกไทเมอร์	29
2.9.5 การรีเซตของ PIC16F8XX	30
2.9.6 เพาเวอร์ออนรีเซต	30
2.9.7 เพาเวอร์อัปไทเมอร์	30
2.9.8 ออสซิลเลเตอร์ สตาร์ทอัปไทเมอร์	31
2.10 การอินเตอร์รัพต์ของ PIC16F8XX	31
2.10.1 รายละเอียดของแหล่งกำเนิดอินเตอร์รัพต์	32
2.10.2 การเก็บข้อมูลเมื่อเกิดการอินเตอร์รัพต์	33
2.10.3 ตำแหน่งเริ่มต้นของการบริการอินเตอร์รัพต์	33
2.10.4 การบริการอินเตอร์รัพต์เมื่อมีแหล่งกำเนิดอินเตอร์รัพต์ภายนอกหลายแหล่ง	33
2.11 ไดโอดเปล่งแสง	34
2.12 คอตเมตริกซ์	36
2.13 LED ตัวเลข 7 ส่วน	37
2.14 สวิตช์เมตริกซ์	39
2.15 จอแสดงผลแบบผลึกเหลว	40
2.15.1 ประเภทของจอแสดงผลแบบผลึกเหลว	40
2.15.2 ส่วนประกอบหลักในจอแสดงผลแบบผลึกเหลว	42
2.15.3 โครงสร้างภายในของตัวควบคุมจอแสดงผลแบบผลึกเหลว	42
2.15.4 จอแสดงผลแบบผลึกเหลว ขนาด 16 ตัวอักษร 1 บรรทัด	43
2.16 I^2C	43
2.17 DS1307 ไอซีสร้างฐานเวลาหรือรีลไทม์คล็อก	44
2.17.1 รายละเอียดขาต่อใช้งานของ DS1307	45
2.17.2 การทำงานของ DS1307	45
2.17.3 โหมดการทำงานของ DS1307	46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
2.18 การแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอล	47
2.18.1 วงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอกแบบกำหนดน้ำหนักตัวต้านทาน	47
2.18.2 วงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอกแบบ R-2R แลดเดอร์	48
2.19 การแปลงสัญญาณดิจิตอลเป็นแอนะลอก	49
2.19.1 แบบเปรียบเทียบแรงดัน	49
2.19.2 แบบวงจรนับแรมป์	50
2.19.3 แบบซักเซสซีฟแอปพร็อกซิเมชั่น	51
2.20 มาตรฐานพอร์ตอนุกรมแบบ RS-232	51
2.20.1 คอนเนกเตอร์สำหรับพอร์ต RS-232 และการเชื่อมต่อ	52
2.21 การสื่อสารข้อมูลแบบอนุกรมด้วย RS485	54
2.21.1 มาตรฐานการสื่อสารข้อมูล	54
2.21.2 รูปแบบของการสื่อสารข้อมูลแบบ RS485	57
2.21.3 มาตรฐานของ RS485	57
2.21.4 การคำนวณหาจำนวนคู่รับ – ส่ง RS485	58
2.21.5 RS485 Protocol	58
2.22 มอเตอร์ไฟฟ้ากระแสตรง	62
2.22.1 ลักษณะโครงสร้างของมอเตอร์แบบขนาน	63
2.22.2 ลักษณะโครงสร้างของมอเตอร์แบบอนุกรม	63
2.22.3 ลักษณะโครงสร้างของมอเตอร์แม่เหล็กแบบถาวร	63
2.22.4 ระบบการควบคุมมอเตอร์ไฟฟ้ากระแสตรงแบบพื้นฐาน	64
2.22.5 การทำงานของแอมปลิไฟล์แบบพัลซ์วิดท์โมดูเลชั่น	65
2.22.6 การควบคุมความเร็วของมอเตอร์ไฟฟ้ากระแสตรง	66
2.23 สเตปเปอร์มอเตอร์	67
2.23.1 ความรู้เบื้องต้น และหลักการทำงาน	67
2.24 โซลิดสเตทรีเลย์	72
2.24.1 ส่วนประกอบและการทำงานของโซลิดสเตทรีเลย์	72
2.24.2 คุณสมบัติทางด้านเอาต์พุตของโซลิดสเตทรีเลย์	72

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
2.25 การขยายออกทางลำโพง	73
บทที่ 3 การออกแบบ การสร้าง และการทำงาน	74
3.1 กล่าวนำ	74
3.2 การออกแบบทางด้านฮาร์ดแวร์	74
3.2.1 โมดูลหลัก PIC-ICSP	74
3.2.2 โมดูลเครื่องมือวัด	78
3.2.3 โมดูลเบสิกอินพุตเอาต์พุตพื้นฐาน	82
3.2.4 โมดูลแสดงผลแอลอีดีดอตเมตริกซ์	85
3.2.5 โมดูลแสดงผลตัวเลข 7 ส่วน	88
3.2.6 โมดูลดิพสวิตช์และเมตริกซ์สวิตช์	90
3.2.7 โมดูลแสดงผลแบบผลึกเหลว	93
3.2.8 โมดูลเชื่อมต่ออุปกรณ์ RTC / EEPROM / SPEAKER	95
3.2.9 โมดูลแปลงระหว่างสัญญาณแอนะลอกและสัญญาณดิจิตอล	98
3.2.10 โมดูลสื่อสารข้อมูลอนุกรม	102
3.2.11 โมดูลขับดีซีมอเตอร์และสเตปปิ้งมอเตอร์	105
3.2.12 โมดูลโซลิดสเตทรีเลย์	108
3.2.13 โมดูลทดลองวงจร	110
3.2.14 โมดูลภาคจ่ายไฟ	112
3.3 การออกแบบทางด้านซอฟต์แวร์	114
3.3.1 การออกแบบโปรแกรม PIC – ICSP Device	114
3.3.2 การส่งข้อมูลโปรแกรมลงหน่วยความจำโปรแกรมของ PIC16F8XX	114
3.3.3 การเรียกใช้โปรแกรม IC-Prog แบบ Command Line	116
3.3.4 การสร้างโปรแกรม PIC-ICSP Device และรายละเอียดการสร้าง	117
3.3.5 การใช้ฟังก์ชั่นเรียกใช้งานโปรแกรม IC-Prog.exe	117
3.3.6 การควบคุมโมดูล PIC-ICSP	117
3.3.7 การใช้งานโปรแกรม PIC-ICSP Device	117

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
3.4 การออกแบบโปรแกรมทดลองการทำงานของโมดูล	120
3.4.1 ผังการทำงานของโปรแกรมทดลองโมดูลเบสิกอินพุตเอาต์พุตพื้นฐาน	120
3.4.2 ผังการทำงานของโปรแกรมทดลองแอลอีดีดอตเมตริกซ์	120
3.4.3 ผังการทำงานของโปรแกรมทดลองโมดูลแสดงผลตัวเลข 7 ส่วน นับ4 หลัก	121
3.4.4 ผังการทำงานของโปรแกรมทดลองดิพสวิตช์	121
3.4.5 ผังการทำงานของโปรแกรมทดลองเมตริกสวิตช์	122
3.4.6 ผังการทำงานของโปรแกรมทดลองโมดูลแสดงผลแบบผลึกเหลว	123
3.4.7 ผังการทำงานของโปรแกรมทดลองการอ่านข้อมูลวินาทีจาก RTC	124
3.4.8 ผังการทำงานของโปรแกรมทดลองลำโพง ความถี่ 1 kHz	125
3.4.9 ผังการทำงานของโปรแกรมทดลองการแปลงสัญญาณแอนะลอกเป็นดิจิตอล	125
3.4.10 ผังการทำงานของโปรแกรมทดลองการรับข้อมูลดิจิตอลจากดิพสวิตช์	126
3.4.11 ผังการทำงานของโปรแกรมทดลองการรับส่งข้อมูลอนุกรมแบบ RS-232	126
3.4.12 ผังการทำงานของโปรแกรมทดลองการรับส่งข้อมูลอนุกรมแบบ RS-485	127
3.4.13 ผังการทำงานของโปรแกรมทดลองการควบคุมดีซีมอเตอร์	127
3.4.14 ผังการทำงานของโปรแกรมทดลองการควบคุมสเตปปิ้งมอเตอร์แบบเวฟ	128
3.4.15 ผังการทำงานของโปรแกรมทดลองการควบคุมโซลิดสเตทรีเลย์	128
บทที่ 4 การทดลองและผลการทดลอง	129
4.1 กล่าวนำ	129
4.2 การทดลองและผลการทดลอง	129
4.2.1 การทดลองใช้งานโมดูล PIC-ICSP	129
4.2.2 การทดลองใช้งานโมดูลเบสิกอินพุตเอาต์พุตพื้นฐาน	131
4.2.3 การทดลองใช้งานโมดูลเครื่องมือวัด	132
4.2.4 การทดลองใช้งานโมดูลแสดงผลแอลอีดีดอตเมตริกซ์	134
4.2.5 การทดลองใช้งานโมดูลแสดงผลตัวเลข 7 ส่วน	135
4.2.6 การทดลองใช้งานโมดูลดิพสวิตช์และเมตริกซ์สวิตช์	139
4.2.7 การทดลองใช้งานโมดูลแสดงผลแบบผลึกเหลว	142

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
4.2.8 การทดลองใช้งานโมดูลเชื่อมต่ออุปกรณ์ RTC / EEPROM / SPEAKER	146
4.2.9 การทดลองใช้งานโมดูลแปลงสัญญาณแอนะลอกและดิจิตอล	151
4.2.10 การทดลองใช้งานโมดูลสื่อสารข้อมูลอนุกรม	156
4.2.11 การทดลองใช้งานโมดูลขับดีซีมอเตอร์และสเตปปิ้งมอเตอร์	161
4.2.12 การทดลองใช้งานโมดูลโซลิดสเตทรีเลย์	164
บทที่ 5 บทสรุป ปัญหา แนวทางแก้ไขและพัฒนา	175
5.1 บทสรุป	175
5.1.1 โมดูลเชื่อมต่อของชุดปฏิบัติการทดลองไมโครคอนโทรลเลอร์ PIC16F8XX	175
5.1.2 ใบงานการทดลอง	176
5.2 ปัญหาและแนวทางแก้ไข	177
5.2.1 ปัญหาทางด้านฮาร์ดแวร์	177
5.2.2 ปัญหาทางด้านซอฟต์แวร์	178
5.3 ข้อเสนอแนะและแนวทางในการพัฒนา	178
ภาคผนวก ก เครื่องต้นแบบ	179
ภาคผนวก ข คู่มือการใช้โมดูลและโปรแกรม	181
ภาคผนวก ค วงจร และแผ่นวงจรพิมพ์	196
ภาคผนวก ง รายละเอียด และคุณสมบัติของอุปกรณ์	219
ภาคผนวก จ ใบงานการทดลอง	224
ภาคผนวก ฉ เฉลยใบงานการทดลอง	325
บรรณานุกรม	360
ประวัติผู้แต่ง	361

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง	หน้า
ตารางที่ 2.1 รายละเอียดของบิตต่างๆ ในรีจิสเตอร์ STATUS	12
ตารางที่ 2.2 รายละเอียดของบิตต่างๆ ในรีจิสเตอร์ OPTION	13
ตารางที่ 2.3 การกำหนดอัตราส่วนของปรีสเกลเลอร์	15
ตารางที่ 2.4 รายละเอียดของบิตต่างๆ ในรีจิสเตอร์ INTCON	15
ตารางที่ 2.5 สรุปการแบ่งกลุ่มคำสั่งของ PIC16F8XX ตามการกำหนดโดย MPASM	18
ตารางที่ 2.6 รายละเอียดของรีจิสเตอร์ EECON1	26
ตารางที่ 2.7 ข้อมูลดิจิตอลเอาต์พุตและค่าเลขฐานสิบหกเพื่อการแสดงตัวเลขแบบ 7 ส่วน	39
ตารางที่ 2.8 เปรียบเทียบจำนวนของสายเมื่อต่อแบบธรรมดา และแบบเมตริกซ์	40
ตารางที่ 2.9 การจัดขาของคอนเนกเตอร์พอร์ตอนุกรมตามมาตรฐาน RS-232 แบบ DB-9 และDB-25	54
ตารางที่ 2.10 การเปรียบเทียบมาตรฐานการสื่อสารข้อมูลของ EIA	55
ตารางที่ 2.11 ข้อเปรียบเทียบระหว่างการส่งข้อมูลแบบขนานและแบบอนุกรม	56
ตารางที่ 2.12 เฟรมของข้อมูล HEADER PORTION และ DATA PORT	59
ตารางที่ 2.13 รายละเอียดของ HEADER PORTION	59
ตารางที่ 2.14 ชนิดของข้อมูลที่บรรจุอยู่ในส่วนของ DATA PORTION	59
ตารางที่ 2.15 การรับ – ส่งข้อมูลอย่างต่อเนื่องของ File Sepuence Transfer	60
ตารางที่ 2.16 ข้อความที่ผิดพลาดของ Fserr	61
ตารางที่ 2.17 DATA PORTION ของ RS485 Protocol	61
ตารางที่ 2.18 ข้อมูลที่ใช้ในการติดต่อของ DATA FIELD	62
ตารางที่ 2.19 ขั้นตอนการทำงานแบบเวฟ	70
ตารางที่ 2.20 ขั้นตอนการทำงานแบบ 2 เฟส	71
ตารางที่ 2.21 ขั้นตอนการทำงานแบบครึ่งสเต็ป	72
ตารางที่ จ.1 ผลการติดของ LED จากการกดสวิตช์	230
ตารางที่ จ.2 ตัวอย่างความถี่สัญญาณนาฬิกาของวงจรกำเนิดสัญญาณนาฬิกาแบบ RC	233
ตารางที่ จ.3 ความสัมพันธ์ของการทำงานร่วมกันของขา RS, R/ W และ E	251
ตารางที่ จ.4 การควบคุมการเลื่อนเคอร์เซอร์บนจอแสดงผล	253
ตารางที่ จ.5 ข้อมูลของการแสดงผลตัวเลข 0-F ของ LED ตัวเลข 7 ส่วนแบบคาโถดร่วม	262

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง (ต่อ)

ตาราง	หน้า
ตารางที่ จ.6 สรุปการค้นหาตำแหน่งของสวิตช์ที่ถูกกด	271
ตารางที่ จ.7 ผลการติดของ LED ตัวเลข 7 ส่วนจากการกดสวิตช์	273
ตารางที่ จ.8 ลำดับการทำงานของสเตปปิ้งมอเตอร์ได้รับการกระตุ้นแบบเวฟ	292
ตารางที่ จ.9 ลำดับการทำงานของสเตปปิ้งมอเตอร์ได้รับการกระตุ้นแบบสองเวฟ	293
ตารางที่ จ.10 ลำดับการทำงานของสเตปปิ้งมอเตอร์ได้รับการกระตุ้นแบบครึ่งสเตป	293
ตารางที่ จ.11 ผลการกดดิพสวิตช์เมื่อใช้วงจร DAC	312
ตารางที่ จ.12 ผลการกดดิพสวิตช์เมื่อใช้วงจร R-2R Ladder	312
ตารางที่ จ.13 ผลการกดคีย์ตัวอักษรบนคีย์บอร์ด	323
ตารางที่ จ.14 ผลการกดสวิตช์ที่โมดูลเบสิกอินพุตเอาต์พุต	324
ตารางที่ ฉ.1 ผลการติดของ LED จากการกดสวิตช์	327
ตารางที่ ฉ.2 ผลการติดของ LED ตัวเลข 7 ส่วนจากการกดสวิตช์	341
ตารางที่ ฉ.3 ผลการกดดิพสวิตช์เมื่อใช้วงจร DAC	354
ตารางที่ ฉ.4 ผลการกดดิพสวิตช์เมื่อใช้วงจร R-2R Ladder	355
ตารางที่ ฉ.5 ผลการกดคีย์ตัวอักษรบนคีย์บอร์ด	358
ตารางที่ ฉ.6 ผลการกดสวิตช์ที่โมดูลเบสิกอินพุตเอาต์พุต	359

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูป	หน้า
รูปที่ 2.1 สถาปัตยกรรมของไมโครคอนโทรลเลอร์ PIC16F8XX	6
รูปที่ 2.2 การจัดขาต่อใช้งานของ PIC16F8XX	7
รูปที่ 2.3 แผนผังแสดงจังหวะการทำงานของ PIC16F8XX	9
รูปที่ 2.4 การทำงานแบบไปป์ไลน์ที่ใช้ใน PIC16F8XX	9
รูปที่ 2.5 การจัดสรรหน่วยความจำโปรแกรมของ PIC16F8XX	11
รูปที่ 2.6 การถ่ายทอดข้อมูลภายในโปรแกรมเคาน์เตอร์	17
รูปที่ 2.7 แผนผังการทำงานของไทม์เมอร์เคาน์เตอร์ภายใน PIC16F8XX	24
รูปที่ 2.8 แผนผังการทำงานของวอตซ์ด็อกไทเมอร์	29
รูปที่ 2.9 การต่อวงจรรีเซตจากภายนอก	30
รูปที่ 2.10 แผนผังลอจิกนัยสำคัญของตัวแปรที่ใช้ควบคุมการอินเตอร์รัพต์	32
รูปที่ 2.11 โครงสร้างและสัญลักษณ์ของไดโอดเปล่งแสง	34
รูปที่ 2.12 การตรวจสอบไดโอดเปล่งแสง	35
รูปที่ 2.13 การต่อไดโอดเปล่งแสงแบบแมตริกซ์	36
รูปที่ 2.14 การสร้างตัวอักษร	37
รูปที่ 2.15 การจัดวางไดโอดเปล่งแสงเพื่อแสดงผลแบบ 7 ส่วน	37
รูปที่ 2.16 การจัดขาของ LED ตัวเลข 7 ส่วน ทั้งแบบตัวถังเดี่ยวและตัวถังคู่	38
รูปที่ 2.17 ไดโอดเปล่งแสงแบบ 7 ส่วน และการแสดงตัวเลขของฐานสิบและฐานสิบหก	38
รูปที่ 2.18 แผนผังการทำงานของจอแสดงผลแบบผลึกเหลวชนิดอักขระ	41
รูปที่ 2.19 รูปร่างและการจัดขาจอแสดงผลแบบผลึกเหลว แบบอักขระ	43
รูปที่ 2.20 การจัดขาของไอซี DS1307	45
รูปที่ 2.21 โครงสร้างภายในของ DS1307	46
รูปที่ 2.22 วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะลอกแบบกำหนดน้ำหนักตัวต้านทาน	48
รูปที่ 2.23 วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะลอกแบบ R-2R แลดเดอร์	48
รูปที่ 2.24 การแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะลอก	49
รูปที่ 2.25 วงจรแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอล	49
รูปที่ 2.26 แบบวงจรนับแรมป์	50
รูปที่ 2.27 แบบซักเซสซีฟแอปพร็อกซิเมชั่น	41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ 2.28 คอนเนกเตอร์อนุกรม 9 ขา หรือแบบ DB-9 (มองจากด้านหลังคอมพิวเตอร์)	53
รูปที่ 2.29 คอนเนกเตอร์อนุกรม 25 ขา หรือแบบ DB-25 (มองจากด้านหลังคอมพิวเตอร์)	53
รูปที่ 2.30 โครงสร้างของการสื่อสารข้อมูลแบบอนุกรมด้วย RS485	55
รูปที่ 2.31 ลักษณะโครงสร้างของมอเตอร์แบบขนาน	63
รูปที่ 2.32 ลักษณะโครงสร้างของมอเตอร์แบบอนุกรม	63
รูปที่ 2.33 ลักษณะโครงสร้างของมอเตอร์แม่เหล็กแบบถาวร	64
รูปที่ 2.34 ระบบการควบคุมมอเตอร์ไฟฟ้ากระแสตรงแบบพื้นฐาน	64
รูปที่ 2.35 วงจรพื้นฐานของการขยายพัลซ์วิดท์โมดูเลชั่น	65
รูปที่ 2.36 การควบคุมแรงดันไฟตรงของอาร์เมเจอร์	66
รูปที่ 2.37 การควบคุมความเข้มของสนามแม่เหล็ก	67
รูปที่ 2.38 โครงสร้างของสเตปเปอร์มอเตอร์	68
รูปที่ 2.39 การควบคุมระบบสเตปเปอร์มอเตอร์	68
รูปที่ 2.40 การพับขดลวดบนสเตเตอร์ของสเต็ปมอเตอร์	69
รูปที่ 2.41 วงจรจ่ายไฟให้กับสเตปเปอร์มอเตอร์	70
รูปที่ 2.42 ระดับผลที่ได้จากลำโพง	73
รูปที่ 3.1 วงจรภาค In-Circuit Serial Programming ของ PIC-ICSP Module	75
รูปที่ 3.2 วงจรภาคกำเนิดสัญญาณนาฬิกา	76
รูปที่ 3.3 แผนผังอุปกรณ์ของโมดูลหลัก PIC-ICSP	77
รูปที่ 3.4 โมดูลหลัก PIC-ICSP	77
รูปที่ 3.5 วงจรหลักและวงจรภาคแสดงผลของโมดูลเครื่องมือวัด	78
รูปที่ 3.6 วงจรของภาคโวลต์มิเตอร์	79
รูปที่ 3.7 วงจรภาคเครื่องนับความถี่	80
รูปที่ 3.8 วงจรภาครับส่งข้อมูลอนุกรม RS-485	80
รูปที่ 3.9 วงจรในส่วนอื่นของโมดูลเครื่องมือวัด	81
รูปที่ 3.10 แผนผังอุปกรณ์ของโมดูลเครื่องมือวัด	81
รูปที่ 3.11 โมดูลเครื่องมือวัด	82
รูปที่ 3.12 วงจรของส่วนเบสิกสวิตช์พื้นฐาน	83

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ 3.13 วงจรของส่วน LED แสดงผลพื้นฐาน	84
รูปที่ 3.14 แผนผังอุปกรณ์ของโมดูลเบสิกอินพุตเอาต์พุตพื้นฐาน	84
รูปที่ 3.15 โมดูลเบสิกอินพุตเอาต์พุตพื้นฐาน	85
รูปที่ 3.16 วงจรของโมดูลแสดงผลแอลอีดีดอตเมตริกซ์	86
รูปที่ 3.17 แผนผังอุปกรณ์ของโมดูลแสดงผลแอลอีดีดอตเมตริกซ์	87
รูปที่ 3.18 โมดูลแสดงผลแอลอีดีดอตเมตริกซ์	87
รูปที่ 3.19 วงจรของโมดูลแสดงผลตัวเลข 7 ส่วน	88
รูปที่ 3.20 แผนผังอุปกรณ์ของโมดูลแสดงผลตัวเลขเจ็ดส่วน	89
รูปที่ 3.21 โมดูลแสดงผลตัวเลขเจ็ดส่วน	90
รูปที่ 3.22 วงจรของดิพสวิตช์ขนาด 8 บิต	91
รูปที่ 3.23 วงจรเมตริกซ์สวิตช์ขนาด 4x4	91
รูปที่ 3.24 แผนผังอุปกรณ์โมดูลดิพสวิตช์และเมตริกซ์สวิตช์	92
รูปที่ 3.25 โมดูลดิพสวิตช์และเมตริกซ์สวิตช์	92
รูปที่ 3.26 วงจรของโมดูลแสดงผลแบบผลึกเหลว	93
รูปที่ 3.27 แผนผังอุปกรณ์ของโมดูลแสดงผลแบบผลึกเหลว	94
รูปที่ 3.28 โมดูลแสดงผลแบบผลึกเหลว	95
รูปที่ 3.29 วงจรติดต่อกับไอซีฐานเวลา DS1307	96
รูปที่ 3.30 วงจรติดต่อกับ EEROM ตระกูล 24XXX	96
รูปที่ 3.31 วงจรเชื่อมต่อลำโพง	97
รูปที่ 3.32 แผนผังอุปกรณ์ของโมดูลเชื่อมต่ออุปกรณ์ RTC/EEPROM/SPEAKER	97
รูปที่ 3.33 โมดูลเชื่อมต่ออุปกรณ์ RTC/EEPROM/SPEAKER	98
รูปที่ 3.34 วงจรสร้างสัญญาณแอนะลอก	99
รูปที่ 3.35 วงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอกแบบใช้ไอซี DAC0808	100
รูปที่ 3.36 วงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอกแบบ R-2R Ladder	101
รูปที่ 3.37 แผนผังอุปกรณ์ของโมดูลแปลงสัญญาณแอนะลอกและดิจิตอล	101
รูปที่ 3.38 โมดูลแปลงสัญญาณแอนะลอกและดิจิตอล	102
รูปที่ 3.39 วงจรสื่อสารข้อมูลอนุกรมมาตรฐาน RS-232	103

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ 3.40 วงจรสื่อสารข้อมูลอนุกรมมาตรฐาน RS-485 และ RS-422	103
รูปที่ 3.41 แผนผังอุปกรณ์ของโมดูลสื่อสารข้อมูลอนุกรม	104
รูปที่ 3.42 โมดูลสื่อสารข้อมูลอนุกรม	104
รูปที่ 3.43 วงจรขับดีซีมอเตอร์	105
รูปที่ 3.44 วงจรขับสเตปปิ้งมอเตอร์	106
รูปที่ 3.45 จั๊มเปอร์ที่ใช้เลือกแรงดันในการขับมอเตอร์	106
รูปที่ 3.46 แผนผังอุปกรณ์ของโมดูลขับดีซีมอเตอร์และสเตปปิ้งมอเตอร์	107
รูปที่ 3.47 โมดูลขับดีซีมอเตอร์และสเตปปิ้งมอเตอร์	107
รูปที่ 3.48 วงจรโซลิดสเตทรีเลย์ขนาด 8 ช่อง	108
รูปที่ 3.49 แผนผังอุปกรณ์ของโมดูลโซลิดสเตทรีเลย์	109
รูปที่ 3.50 โมดูลโซลิดสเตทรีเลย์	110
รูปที่ 3.51 วงจรในโมดูลทดลองวงจร	111
รูปที่ 3.52 แผนผังอุปกรณ์ของโมดูลทดลองวงจร	111
รูปที่ 3.53 โมดูลทดลองวงจร	112
รูปที่ 3.54 วงจรภาคจ่ายไฟ	113
รูปที่ 3.55 แผนผังอุปกรณ์ของโมดูลภาคจ่ายไฟ	113
รูปที่ 3.56 โมดูลภาคจ่ายไฟ	114
รูปที่ 3.57 ขั้นตอนการโหลดข้อมูลโปรแกรมลงบนหน่วยความจำโปรแกรม	115
รูปที่ 3.58 ฟอร์มของโปรแกรม PIC-ICSP Device	117
รูปที่ 3.59 หน้าต่างแสดงขณะทำการโปรแกรม	118
รูปที่ 3.60 เมื่อโปรแกรมเสร็จสมบูรณ์	118
รูปที่ 3.61 เมื่อเกิดความผิดพลาดขณะทำการโปรแกรม	119
รูปที่ 3.62 ผังการทำงานของโปรแกรมทดลองโมดูลเบสิกอินพุตเอาต์พุตพื้นฐาน	120
รูปที่ 3.63 ผังการทำงานของโปรแกรมทดลองแอลอีดีดอตเมตริกซ์	120
รูปที่ 3.64 ผังการทำงานของโปรแกรมทดลองโมดูลแสดงผลตัวเลข 7 ส่วน นับเลข 4 หลัก	121
รูปที่ 3.65 ผังการทำงานของโปรแกรมทดลองดิพสวิตช์	121
รูปที่ 3.66 ผังการทำงานของโปรแกรมทดลองเมตริกสวิตช์	122

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ 3.67 ผังการทำงานของโปรแกรมทดลองโมดูลแสดงผลแบบผลึกเหลว	123
รูปที่ 3.68 ผังการทำงานของโปรแกรมทดลองการอ่านข้อมูลวินาทีจาก RTC	124
รูปที่ 3.69 ผังการทำงานของโปรแกรมทดลองลำโพง ความถี่ 1 kHz	125
รูปที่ 3.70 ผังการทำงานของโปรแกรมทดลองการแปลงสัญญาณแอนะลอกเป็นดิจิตอล	125
รูปที่ 3.71 ผังการทำงานของโปรแกรมทดลองการรับข้อมูลดิจิตอลจากดิพสวิตช์	126
รูปที่ 3.72 ผังการทำงานของโปรแกรมทดลองการรับส่งข้อมูลอนุกรมแบบ RS-232	126
รูปที่ 3.73 ผังการทำงานของโปรแกรมทดลองการรับส่งข้อมูลอนุกรมแบบ RS-485	127
รูปที่ 3.74 ผังการทำงานของโปรแกรมทดลองการควบคุมดีซีมอเตอร์	127
รูปที่ 3.75 ผังการทำงานของโปรแกรมทดลองการควบคุมสเตปปิ้งมอเตอร์แบบเวฟ	128
รูปที่ 3.76 ผังการทำงานของโปรแกรมทดลองการควบคุมโซลิดสเตทรีเลย์	128
รูปที่ 4.1 การเชื่อมต่อโมดูล PIC-ICSP เข้ากับคอมพิวเตอร์	129
รูปที่ 4.2 โปรแกรมที่ใช้ทดลองดาวน์โหลดข้อมูลของโมดูล PIC-ICSP	130
รูปที่ 4.3 การเชื่อมต่อใช้งานโมดูลเบสิกอินพุตเอาต์พุต	131
รูปที่ 4.4 โปรแกรมทดลองโมดูลเบสิกอินพุตเอาต์พุต	132
รูปที่ 4.5 การเชื่อมต่อใช้งานโมดูลแอลอีดีดอตเมตริกซ์	134
รูปที่ 4.6 โปรแกรมทดลองโมดูลแอลอีดีดอตเมตริกซ์	135
รูปที่ 4.7 การเชื่อมต่อใช้งานโมดูลแสดงผลตัวเลข 7 ส่วน	136
รูปที่ 4.8 โปรแกรมทดลองโมดูลแสดงผลตัวเลข 7 ส่วน	139
รูปที่ 4.9 การเชื่อมต่อใช้งานดิพสวิตช์	139
รูปที่ 4.10 โปรแกรมทดลองดิพสวิตช์	140
รูปที่ 4.11 การเชื่อมต่อใช้งานเมตริกซ์สวิตช์	141
รูปที่ 4.12 โปรแกรมทดลองเมตริกสวิตช์	142
รูปที่ 4.13 การเชื่อมต่อใช้งานโมดูลแสดงผลแบบผลึกเหลวแบบ 8 บิต	143
รูปที่ 4.14 โปรแกรมทดลองโมดูลแสดงผลแบบผลึกเหลว	145
รูปที่ 4.15 การเชื่อมต่อใช้งานอุปกรณ์ทั้ง 3 ชนิดของ โมดูลเชื่อมต่ออุปกรณ์	146
รูปที่ 4.16 โปรแกรมทดลองอ่านข้อมูลวินาทีจาก RTC	150
รูปที่ 4.17 โปรแกรมทดลองลำโพง ความถี่ 1 kHz	151

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ 4.18 การเชื่อมต่อวงจรปรับแรงดันเพื่อทดลองแปลงสัญญาณแอนะลอกเป็นดิจิตอล	152
รูปที่ 4.19 โปรแกรมทดลองแปลงสัญญาณแอนะลอกเป็นดิจิตอล	153
รูปที่ 4.20 การเชื่อมต่อวงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอกแบบ R-2R Ladder	154
รูปที่ 4.21 โปรแกรมรับข้อมูลดิจิตอลจากดิพสวิตช์	154
รูปที่ 4.22 การเชื่อมต่อวงจร DAC0808 เข้ากับโมดูลหลักและดิพสวิตช์	155
รูปที่ 4.23 การเชื่อมต่อส่งสัญญาณข้อมูลอนุกรมแบบ RS-232 กับคอมพิวเตอร์	156
รูปที่ 4.24 โปรแกรมทดลองรับส่งข้อมูลกับคอมพิวเตอร์อนุกรมแบบ RS-232	157
รูปที่ 4.25 การเชื่อมต่อโมดูลทดลองส่งสัญญาณข้อมูลแบบ RS-485	159
รูปที่ 4.26 โปรแกรมทดลองรับส่งข้อมูลอนุกรมแบบ RS-485	160
รูปที่ 4.27 การเชื่อมต่อโมดูลขับดีซีมอเตอร์	161
รูปที่ 4.28 โปรแกรมควบคุมดีซีมอเตอร์	162
รูปที่ 4.29 การเชื่อมต่อโมดูลขับสเตปปิ้งมอเตอร์	163
รูปที่ 4.30 โปรแกรมควบคุมสเตปปิ้งมอเตอร์แบบเวฟ	164
รูปที่ 4.31 การเชื่อมต่อโมดูลโซลิดสเตทรีเลย์เข้ากับโมดูลหลัก	165
รูปที่ 4.32 โปรแกรมควบคุมโซลิดสเตทรีเลย์	165
รูปที่ 4.33 ผลการทดลองโมดูลเบสิกอินพุตเอาต์พุต	166
รูปที่ 4.34 ผลการทดลองโมดูลเครื่องมือวัด (เครื่องมือวัดแรงดัน)	167
รูปที่ 4.35 ผลการทดลองโมดูลแอลอีดีดอตเมตริกซ์	167
รูปที่ 4.36 ผลการทดลองโมดูลแสดงผลตัวเลข 7 ส่วน	168
รูปที่ 4.37 ผลการทดลองดิพสวิตช์	168
รูปที่ 4.38 ผลการทดลองเมตริกสวิตช์	169
รูปที่ 4.39 ผลการทดลองโมดูลแสดงผลแบบผลึกเหลว	169
รูปที่ 4.40 ผลการทดลองอ่านข้อมูลวินาทีจาก RTC	170
รูปที่ 4.41 ผลการทดลองแปลงสัญญาณแอนะลอกเป็นดิจิตอล	170
รูปที่ 4.42 ผลการทดลองวงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอกแบบ R-2R Ladder	171
รูปที่ 4.43 ผลการทดลองวงจร DAC0808 เข้ากับโมดูลหลักและดิพสวิตช์	171
รูปที่ 4.44 ผลการทดลองรับส่งข้อมูลกับคอมพิวเตอร์อนุกรมแบบ RS-232	172

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ 4.45 ผลการทดลองส่งสัญญาณข้อมูลแบบ RS-485	172
รูปที่ 4.46 ผลการทดลองควบคุมดีซีมอเตอร์	173
รูปที่ 4.47 ผลการทดลองควบคุมสเตปปิ้งมอเตอร์แบบเวฟ	173
รูปที่ 4.48 ผลการทดลองควบคุมโซลิดสเตทรีเลย์	174
รูปที่ ก.1 เครื่องต้นแบบโมดูลทั้งหมด	180
รูปที่ ก.2 เครื่องต้นแบบโมดูลที่บรรจุในกระเป๋า	180
รูปที่ ข.1 จุดเชื่อมต่อใช้งานของโมดูลหลัก PIC-ICSP	182
รูปที่ ข.2 จุดเชื่อมต่อใช้งานของโมดูลเครื่องมือวัด	183
รูปที่ ข.3 จุดเชื่อมต่อใช้งานของโมดูลเบสิกอินพุตเอาต์พุตพื้นฐาน	184
รูปที่ ข.4 จุดเชื่อมต่อใช้งานของโมดูลแสดงผลแอลอีดีดอตเมตริกซ์	184
รูปที่ ข.5 จุดเชื่อมต่อใช้งานของโมดูลแสดงผลตัวเลข 7 ส่วน	185
รูปที่ ข.6 จุดเชื่อมต่อใช้งานของโมดูลดิพสวิตช์และเมตริกซ์สวิตช์	186
รูปที่ ข.7 จุดเชื่อมต่อใช้งานของโมดูลแสดงผลแบบผลึกเหลว	186
รูปที่ ข.8 จุดเชื่อมต่อใช้งานของโมดูลเชื่อมต่ออุปกรณ์ RTC / EEPROM / SPEAKER	187
รูปที่ ข.9 จุดเชื่อมต่อใช้งานของโมดูลแปลงสัญญาณแอนะลอกและดิจิตอล	187
รูปที่ ข.10 จุดเชื่อมต่อใช้งานของโมดูลสื่อสารข้อมูลอนุกรม	189
รูปที่ ข.11 จุดเชื่อมต่อใช้งานของโมดูลขับดีซีมอเตอร์และสเตปปิ้งมอเตอร์	190
รูปที่ ข.12 จุดเชื่อมต่อใช้งานของโมดูลโซลิดสเตทรีเลย์	191
รูปที่ ข.13 จุดเชื่อมต่อใช้งานของโมดูลทดลองวงจร	192
รูปที่ ข.14 จุดเชื่อมต่อใช้งานของโมดูลภาคจ่ายไฟ	192
รูปที่ ข.15 หน้าต่างแสดงขณะทำการโปรแกรม	193
รูปที่ ข.16 เมื่อโปรแกรมเสร็จสมบูรณ์	194
รูปที่ ข.17 เมื่อเกิดความผิดพลาดขณะทำการโปรแกรม	194
รูปที่ ค.1 วงจรสมบูรณ์ของโมดูลหลัก PIC-ICSP	197
รูปที่ ค.2 วงจรสมบูรณ์ของโมดูลเครื่องมือวัด	198
รูปที่ ค.3 วงจรสมบูรณ์ของโมดูลเบสิกอินพุตเอาต์พุตพื้นฐาน	199
รูปที่ ค.4 วงจรสมบูรณ์ของโมดูลแสดงผลแอลอีดีดอตเมตริกซ์	200

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ ค.5 วงจรสมบูรณ์ของโมดูลแสดงผลตัวเลข 7 ส่วน	201
รูปที่ ค.6 วงจรสมบูรณ์ของโมดูลดิพสวิตช์และเมตริกซ์สวิตช์	202
รูปที่ ค.7 วงจรสมบูรณ์ของโมดูลแสดงผลแบบผลึกเหลว	203
รูปที่ ค.8 วงจรสมบูรณ์ของโมดูลเชื่อมต่ออุปกรณ์ RTC / EEPROM / SPEAKER	204
รูปที่ ค.9 วงจรสมบูรณ์ของโมดูลแปลงสัญญาณแอนะลอกและดิจิตอล	205
รูปที่ ค.10 วงจรสมบูรณ์ของโมดูลสื่อสารข้อมูลอนุกรม	206
รูปที่ ค.11 วงจรสมบูรณ์ของโมดูลขับดีซีมอเตอร์และสเตปปิ้งมอเตอร์	207
รูปที่ ค.12 วงจรสมบูรณ์ของโมดูลโซลิดสเตทรีเลย์	208
รูปที่ ค.13 วงจรสมบูรณ์ของโมดูลทดลองวงจร	209
รูปที่ ค.14 วงจรสมบูรณ์ของโมดูลภาคจ่ายไฟ	210
รูปที่ ค.15 แผ่นวงจรพิมพ์ด้านล่างของโมดูลหลัก PIC-ICSP	211
รูปที่ ค.16 แผ่นวงจรพิมพ์ด้านบนของโมดูลหลัก PIC-ICSP	211
รูปที่ ค.17 แผ่นวงจรพิมพ์ด้านล่างของโมดูลเครื่องมือวัด	212
รูปที่ ค.18 แผ่นวงจรพิมพ์ด้านล่างของโมดูลเบสิกอินพุตเอาต์พุตพื้นฐาน	212
รูปที่ ค.19 แผ่นวงจรพิมพ์ด้านล่างของโมดูลแสดงผลแอลอีดีดอตเมตริกซ์	213
รูปที่ ค.20 แผ่นวงจรพิมพ์ด้านล่างของโมดูลแสดงผลตัวเลข 7 ส่วน	213
รูปที่ ค.21 แผ่นวงจรพิมพ์ด้านล่างของโมดูลดิพสวิตช์และเมตริกซ์สวิตช์	214
รูปที่ ค.22 แผ่นวงจรพิมพ์ด้านล่างของโมดูลแสดงผลแบบผลึกเหลว	214
รูปที่ ค.23 แผ่นวงจรพิมพ์ด้านล่างของโมดูลเชื่อมต่ออุปกรณ์ RTC / EEPROM / SPEAKER	215
รูปที่ ค.24 แผ่นวงจรพิมพ์ด้านล่างของโมดูลแปลงสัญญาณแอนะลอกและดิจิตอล	215
รูปที่ ค.25 แผ่นวงจรพิมพ์ด้านล่างของโมดูลสื่อสารข้อมูลอนุกรม	216
รูปที่ ค.26 แผ่นวงจรพิมพ์ด้านล่างของโมดูลขับดีซีมอเตอร์และสเตปปิ้งมอเตอร์	216
รูปที่ ค.27 แผ่นวงจรพิมพ์ด้านล่างของโมดูลโซลิดสเตทรีเลย์	217
รูปที่ ค.28 แผ่นวงจรพิมพ์ด้านล่างของโมดูลทดลองวงจร	218
รูปที่ ค.29 แผ่นวงจรพิมพ์ด้านล่างของโมดูลภาคจ่ายไฟ	218
รูปที่ จ.1 ตัวอย่างโปรแกรมย่อยเป็นการอินิเชียลพอร์ต A	226
รูปที่ จ.2 การเชื่อมต่อระหว่างโมดูลหลัก PIC-ICSP และโมดูลเบสิกอินพุตเอาต์พุต	226

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ จ.3 โปรแกรมการใช้งานพอร์ตของ PIC16F876	227
รูปที่ จ.4 โปรแกรมการควบคุมเอาต์พุตจากอินพุตพอร์ตของ PIC16F876	228
รูปที่ จ.5 การต่อวงจรกำเนิดสัญญาณนาฬิกาแบบ RC	232
รูปที่ จ.6 การต่อวงจรกำเนิดสัญญาณนาฬิกาแบบ XT	233
รูปที่ จ.7 การเชื่อมต่อระหว่างโมดูลหลัก PIC-ICSP และโมดูลเบสิกอินพุตเอาต์พุต	234
รูปที่ จ.8 โปรแกรมใช้ในการทดลองการกำเนิดสัญญาณนาฬิกา	234
รูปที่ จ.9 ความหมายของบิตต่างๆ ในรีจิสเตอร์ INTCON	237
รูปที่ จ.10 การเชื่อมต่อระหว่างโมดูล PIC-ICSP และโมดูลเบสิกอินพุตเอาต์พุต	238
รูปที่ จ.11 โปรแกรมการอินเตอร์รัพต์เนื่องจากแปลงลอจิกที่บิต 4 ถึง 7 ของพอร์ต B	239
รูปที่ จ.12 การอินเตอร์รัพต์จากการป้อนสัญญาณภายนอกเข้าที่ขา RB0/INT	241
รูปที่ จ.13 ความหมายของบิตแต่ละบิตในรีจิสเตอร์ OPTION	244
รูปที่ จ.14 แผนผังการทำงานของไทเมอร์เคาน์เตอร์ใน PIC 16F876	244
รูปที่ จ.15 การเชื่อมต่อระหว่างโมดูลหลัก PIC-ICSP และโมดูลเบสิกอินพุตเอาต์พุต	246
รูปที่ จ.16 การใช้ TMR0 ไม่มีอินเตอร์รัพต์ ความถี่ 1 เมกะเฮิรตซ์	246
รูปที่ จ.17 การใช้ TMR0 มีอินเตอร์รัพต์ ความถี่ 0.5 เฮิรตซ์	247
รูปที่ จ.18 การใช้ไทเมอร์เคาน์เตอร์กับสัญญาณนาฬิกาภายนอก	248
รูปที่ จ.19 รูปร่างและการจัดขาของจอแสดงผลแบบผลึกเหลวแบบอักขระ	251
รูปที่ จ.20 คำสั่งเลือกโหมดการป้อนกลับ	252
รูปที่ จ.21 คำสั่งควบคุมการแสดงผล	252
รูปที่ จ.22 การควบคุมการเลื่อนเคอร์เซอร์และข้อมูลตัวอักษร	253
รูปที่ จ.23 การกำหนดฟังก์ชั่นการทำงาน	253
รูปที่ จ.24 การอ่าน Busy และแอดเดรส	254
รูปที่ จ.25 ตัวอย่างโปรแกรมหน่วงเวลา 5 มิลลิวินาที	255
รูปที่ จ.26 ตัวอย่างโปรแกรมการส่งสัญญาณพัลซ์	255
รูปที่ จ.27 ตัวอย่างโปรแกรมหน่วงเวลา 125 ไมโครวินาที	255
รูปที่ จ.28 การเชื่อมต่อโมดูลหลัก PIC-ICSP กับโมดูล LCD แบบ 8 บิต	256
รูปที่ จ.29 โปรแกรมการแสดงผลของ LED แบบ 8 บิต	258

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ จ.30 รูปร่างและการกำหนดชื่อเซกเมนต์ต่างๆของ LED ตัวเลข 7 ส่วน	261
รูปที่ จ.31 วงจรภายในของ LED ตัวเลข 7 ส่วนทั้งแบบคาโถดร่วมและอาโนดร่วม	261
รูปที่ จ.32 การเชื่อมต่อระหว่างโมดูลหลัก PIC-ICSP และโมดูลแสดงผล LED ตัวเลข 7 ส่วน	263
รูปที่ จ.33 โปรแกรมแสดงผล LED ตัวเลข 7ส่วนหลักเดียว	264
รูปที่ จ.34 โปรแกรมแสดงผล LED ตัวเลข 7 ส่วนหลายหลัก	265
รูปที่ จ.35 วงจรของสวิตช์แบบต่อเข้ากับไฟเลี้ยงและกราวด์	269
รูปที่ จ.36 การต่อสวิตช์แบบเมตริกซ์	269
รูปที่ จ.37 การต่อสวิตช์เมตริกส์ขนาด 4x4	270
รูปที่ จ.38 การเชื่อมต่อโมดูล PIC-ICSP กับโมดูลสวิตช์เมตริกส์แสดงผลที่ LED ตัว 7 ส่วน	271
รูปที่ จ.39 โปรแกรมการอ่านค่าคีย์แพดโดยใช้หลักการสแกนคีย์	272
รูปที่ จ.40 การจัดขาและวงจรภายในของไอซี ULN280	276
รูปที่ จ.41 การเชื่อมโมดูลหลัก PIC-ICSP, โมดูลเบสิกอินพุตเอาต์พุตและโมดูลขับมอเตอร์	276
รูปที่ จ.42 การขับมอเตอร์ไฟฟ้ากระแสตรง (DC Motor)	277
รูปที่ จ.43 ดิวตี้ไซเคิ้ลของสัญญาณ	280
รูปที่ จ.44 ความสัมพันธ์ระหว่างค่าของดิวตี้ไซเคิ้ล	280
รูปที่ จ.45 สัญญาณ PWM เอาต์พุต	281
รูปที่ จ.46 การเชื่อมต่อระหว่างโมดูล PIC-ICSP, โมดูขับดีซีมอเตอร์และโมดูลเบสิกอินพุต	283
รูปที่ จ.47 โปรแกรมควบคุมความเร็วดีซีมอเตอร์โดยใช้สัญญาณพัลซ์วิธมอดูเลชั่น	284
รูปที่ จ.48 การจัดขาและวงจรภายในของไอซี ULN2803	288
รูปที่ จ.49 วงจรขับโหลดโดยใช้ทรานซิสเตอร์	289
รูปที่ จ.50 วงจรขับโหลดโดยใช้ทรานซิสเตอร์ต่อแบบคาสเคด	289
รูปที่ จ.51 วงจรขับโหลดโดยใช้ทรานซิสเตอร์แบบดาร์ลิงตัน	290
รูปที่ จ.52 วงจรการขับโหลดด้วยออปโต้คัปเปอร์	291
รูปที่ จ.53 การพันขดลวดของสเตปปิ้งมอเตอร์	291
รูปที่ จ.54 การเชื่อมต่อระหว่างโมดูลหลัก PIC-ICSP เข้ากับโมดูลการขับมอเตอร์	294
รูปที่ จ.55 การขับสเตปปิ้งมอเตอร์	295
รูปที่ จ.56 โปรแกรมขับสเตปปิ้งมอเตอร์แบบ 2 เฟส	296

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ จ.57 แผนผังการทำงานอย่างง่ายของวงจรแปลงสัญญาณแอนะลอกเป็นดิจิตอล	299
รูปที่ จ.58 การสุ่มสัญญาณแอนะลอกเพื่อกำหนดข้อมูลดิจิตอล	299
รูปที่ จ.59 วงจรแปลงสัญญาณแอนะลอกเป็นดิจิตอลแบบเปรียบเทียบแรงดัน	300
รูปที่ จ.60 แผนผังแสดงการทำงานของวงจร ADC แบบวงจรนับแรมป์	301
รูปที่ จ.61 แผนผังแสดงการทำงานของวงจร ADC แบบซักเซสซีฟแอปพร็อกซิเมชั่น	302
รูปที่ จ.62 การเชื่อมต่อโมดูลแปลงสัญญาณแอนะลอกเป็นดิจิตอลแสดงผลออกทาง LED	302
รูปที่ จ.63 การเชื่อมต่อโมดูลแปลงสัญญาณ แสดงผลเป็นตัวเลขที่ LED 7 ส่วน	303
รูปที่ จ.64 โปรแกรมแปลงสัญญาณแอนะลอกเป็นดิจิตอล แสดงผลทาง LED ขนาด 8 บิต	303
รูปที่ จ.65 โปรแกรมแปลงสัญญาณแอนะลอกเป็นดิจิตอล 10 บิต แสดงผลเป็นเลข 0 ถึง 1023	305
รูปที่ จ.66 วงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอกแบบกำหนดน้ำหนักตัวต้านทาน	308
รูปที่ จ.67 พื้นฐานของวงจร R-2R Ladder	309
รูปที่ จ.68 การจัดขาของไอซี DAC0808	310
รูปที่ จ.69 วงจรเพื่อการใช้งานเบื้องต้นของไอซี DAC0808	310
รูปที่ จ.70 การเชื่อมต่อโมดูลหลัก PIC-ICSP เข้ากับโมดูลดิพสวิตช์และโมดูลแปลงสัญญาณ	311
รูปที่ จ.71 โปรแกรมการทดลองแปลงสัญญาณดิจิตอลเป็นแอนะลอก	311
รูปที่ จ.72 การเชื่อมต่อโมดูลหลัก PIC-ICSP และโมดูลโซลิดสเตทรีเลย์	316
รูปที่ จ.73 โปรแกรมใช้ในการทดลองโซลิดสเตทรีเลย์	317
รูปที่ จ.74 ระดับสัญญาณลอจิก RS-232C	319
รูปที่ จ.75 การกำหนดจุดต่อของ RS-232C	319
รูปที่ จ.76 สัญญาณของขาต่างๆ ที่ใช้งาน	320
รูปที่ จ.77 การเชื่อมต่อโมดูลสื่อสารข้อมูลกับคอมพิวเตอร์	321
รูปที่ จ.78 โปรแกรมรับส่งข้อมูลระหว่างไมโครคอนโทรลเลอร์ PIC16F876กับคอมพิวเตอร์	322
รูปที่ ฉ.1 เฉลยโปรแกรมกำหนดพอร์ต A เป็นเอาต์พุต	327
รูปที่ ฉ.2 โปรแกรมแสดงรูปแบบไฟวิ่ง 4 แบบ	329
รูปที่ ฉ.3 เฉลยโปรแกรมแสดงการอินเตอร์รัพต์เนื่องจากระดับลอจิกเปลี่ยนแปลง	333
รูปที่ ฉ.4 เฉลยโปรแกรมนับเลข 0000 ถึง 9999 แสดงบน Seven Segment	340
รูปที่ ฉ.5 เฉลยโปรแกรมสแกนคีย์ สวิตช์เมตริกซ์ 4x4	343

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ ฉ.6 เฉลยโปรแกรมควบคุมความเร็วดีซีมอเตอร์	345
รูปที่ ฉ.7 เฉลยการเขียนโปรแกรมแปลงสัญญาณ A to D แสดงผลที่ Seven Segment	353
รูปที่ ฉ.8 เฉลยโปรแกรมใช้โซลิดสเตทรีเลย์ทำไฟวิ่ง	357

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์

ไมโครคอนโทรลเลอร์เป็นส่วนหนึ่งของระบบอิเล็กทรอนิกส์ ที่มีการใช้ในการควบคุมทางอุตสาหกรรมอย่างกว้างขวาง ยิ่งในปัจจุบันได้มีการพัฒนาไมโครคอนโทรลเลอร์ (Microcontroller) ให้ทำงานในลักษณะ RISC (Reduced Instruction Set Computer) ขึ้น มีคำสั่งการใช้งานเพียง 33-35 คำสั่ง โดยในหนึ่งคำสั่งใช้สัญญาณนาฬิกาเพียงลูกเดียว ยกเว้นคำสั่งที่ใช้ในการกระโดด (Jump) นั่น คือ ไมโครคอนโทรลเลอร์ตระกูล PIC ซึ่งได้รับความนิยมในการนำไปควบคุมสินค้าและผลิตภัณฑ์ต่างๆ อย่างมากมาย ส่งผลให้เกิดความต้องการเรียนรู้ และใช้งานไมโครคอนโทรลเลอร์ตระกูลดังกล่าว

จึงได้สร้างชุดปฏิบัติการทดลองไมโครคอนโทรลเลอร์ PIC16F8XX ขึ้น เพื่อเรียนรู้การทำงานของไมโครคอนโทรลเลอร์ตระกูลนี้ และใช้ประกอบการเรียนการสอนให้มีประสิทธิภาพยิ่งกว่าเดิม นอกจากนี้ยังช่วยเพิ่มทักษะให้กับผู้เรียนซึ่งการทดลองจะทำให้ผู้เรียนเข้าใจได้ง่ายยิ่งขึ้น

1.2 ขีดความสามารถของโครงงาน

โครงงานมีขีดความสามารถดังนี้

1. สามารถต่อใช้งานชุดอุปกรณ์ทดลองต่างๆ โดยแยกเป็นโมดูล
2. สามารถโปรแกรมลงบน PIC16F8XX ได้ในขณะต่อใช้งาน
3. สามารถควบคุมการประมวลผล และการรีเซต ของ PIC16F8XX ได้จากคอมพิวเตอร์
4. สามารถใช้ได้กับไมโครคอนโทรลเลอร์ PIC16F8XX ได้ทั้งแบบ 18 ขา 28 ขา และ40 ขา
5. สามารถต่อโมดูลทดลองต่างๆ เข้ากับพอร์ตใดๆ ของ PIC16F8XX ตามที่ต้องการได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 เนื้อหาโดยสังเขป

เนื้อหาภายในปริญญานิพนธ์ฉบับนี้แบ่งออกเป็นบทต่างๆ เพื่อความสะดวกต่อการศึกษาและทำความเข้าใจ ในแต่ละบทประกอบด้วยเนื้อหาดังต่อไปนี้

บทที่ 2 ทฤษฎีและหลักการ ประกอบด้วยเนื้อหาดังต่อไปนี้ คือ ความรู้เบื้องต้นของไมโครคอนโทรลเลอร์ PIC16F8X และ PIC16F8XX การจัดการหน่วยความจำของ PIC16F8XX รีจีสเตอร์ควบคุมของ PIC16F8XX ชุดคำสั่งของ PIC16F8XX และการเข้าถึงรีจีสเตอร์และข้อมูลของ PIC16F8XX การใช้งาน Input / Output Port ไทเมอร์เคาน์เตอร์ภายใน PIC16F8XX การใช้งาน EEPROM ภายใน คุณสมบัติพิเศษของซีพียู PIC16F8XX การอินเตอร์รัพต์ของ PIC16F8XX ไดโอดเปล่งแสง ดอตเมตริกซ์ LED ตัวเลข 7 ส่วน สวิตช์เมตริกซ์ จอแสดงผลแบบผลึกเหลว I^2C ไอซีสร้างฐานเวลาหรือเรียลไทม์คล็อก (DS1307) การแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอล และการแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะลอก มาตรฐานพอร์ตอนุกรมแบบ RS-232 และRS485 มอเตอร์ไฟฟ้ากระแสตรง สเตปเปอร์มอเตอร์ โซลิดสเตตรีเลย์

บทที่ 3 การออกแบบ การสร้าง และการทำงาน กล่าวถึงเนื้อหาเกี่ยวกับ การออกแบบทางด้านซอฟต์แวร์ ซึ่งแบ่งออกเป็น การออกแบบโปรแกรม PIC – ICSP Device การส่งข้อมูลโปรแกรมลงหน่วยความจำโปรแกรมของ PIC16F8XX การเรียกใช้โปรแกรม IC-Prog แบบ Command Line การสร้างโปรแกรม PIC-ICSP Device และรายละเอียดการสร้าง การใช้ฟังก์ชั่นเรียกใช้งานโปรแกรม IC-Prog.EXE การควบคุมโมดูล PIC-ICSP การใช้งานโปรแกรม PIC-ICSP Device และ การออกแบบทางด้านฮาร์ดแวร์ ซึ่งแบ่งออกเป็นการออกแบบโมดูลหลัก PIC-ICSP โมดูลเครื่องมือวัด โมดูลเบสิกอินพุตเอาต์พุตพื้นฐาน โมดูลแสดงผลแอลอีดีดอตเมตริกซ์ โมดูลแสดงผลตัวเลข 7 ส่วน โมดูลดิพสวิตช์และเมตริกซ์สวิตช์ โมดูลแสดงผลแบบผลึกเหลว โมดูลเชื่อมต่ออุปกรณ์ RTC / EEPROM / SPEAKER โมดูลแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอลและแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะลอก โมดูลสื่อสารข้อมูลอนุกรม โมดูลขับดีซีมอเตอร์และสเตปปิ้งมอเตอร์ โมดูลโซลิดสเตทรีเลย์ โมดูลทดลองวงจร และโมดูลภาคจ่ายไฟ

บทที่ 4 การทดลองและผลการทดลอง ประกอบด้วย การทดลองและผลการทดลองของโมดูลหลัก PIC-ICSP โมดูลเครื่องมือวัด โมดูลอินพุตเอาต์พุตพื้นฐาน โมดูลแสดงผลแอลอีดีดอตเมตริกซ์ โมดูลแสดงผลตัวเลข 7 ส่วน โมดูลดิพสวิตช์และเมตริกซ์สวิตช์ โมดูลจอแสดงผลแบบผลึกเหลว โมดูลเชื่อมต่ออุปกรณ์ RTC / EEPROM / SPEAKER โมดูลแปลงระหว่างสัญญาณแอนะลอกและสัญญาณดิจิตอล โมดูลสื่อสารข้อมูลอนุกรม โมดูลขับดีซีมอเตอร์และสเตปปิ้งมอเตอร์ โมดูลโซลิดสเตทรีเลย์ โมดูลทดลองวงจร และโมดูลภาคจ่ายไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5 บทสรุป ปัญหา แนวทางแก้ไขและพัฒนา ขั้นการสรุปผล ในการจัดทำโครงงาน ปัญหาที่เกิดขึ้น และได้เสนอแนะแนวทางการพัฒนาให้มีประสิทธิภาพมากยิ่งขึ้น

ภาคผนวก ก เครื่องต้นแบบ

ภาคผนวก ข คู่มือการใช้โมดูลและโปรแกรม

ภาคผนวก ค วงจร และแผ่นวงจรพิมพ์

ภาคผนวก ง รายละเอียดและคุณสมบัติของอุปกรณ์

ภาคผนวก จ ใบงานการทดลอง

ภาคผนวก ฉ เฉลยใบงานการทดลอง

บรรณานุกรม

ประวัติผู้แต่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 ความรู้เบื้องต้นของไมโครคอนโทรลเลอร์ PIC16F8XX

PIC16F8XX เป็นไมโครคอนโทรลเลอร์ในตระกูล PIC (Peripheral Interface Controller) ของบริษัทไมโครชิปเทคโนโลยี (Microchip Technology) ไมโครคอนโทรลเลอร์ตระกูล PIC มีด้วยกันหลายเบอร์ แต่ละเบอร์ก็มีความสามารถแตกต่างกันไป ภายใน PIC16F8XX มีหน่วยความจำโปรแกรม (Program Memory) แบบแฟลช (Flash) ซึ่งเป็นหน่วยความจำที่สามารถเขียนและลบได้ด้วยสัญญาณไฟฟ้า

ไมโครคอนโทรลเลอร์ PIC16F8XX เป็นไมโครคอนโทรลเลอร์สมัยใหม่จัดอยู่ในกลุ่มของไมโครโปรเซสเซอร์แบบ RISC (Reduced Instruction Set Computer) คือ ไมโครคอนโทรลเลอร์ตระกูลนี้จะมีชุดคำสั่งน้อยเพียง 33-35 คำสั่งพื้นฐานเท่านั้นและทุกคำสั่งสามารถทำงานให้เสร็จได้ด้วยการใช้สัญญาณนาฬิกาเพียงลูกเดียว ยกเว้นคำสั่งที่ใช้ในการกระโดด (Jump) ทั้งยังทำงานในลักษณะไปป์ไลน์ (Pipe Line) เหมือนกับไมโครโปรเซสเซอร์สมัยใหม่ ความเร็วในการทำงานจึงสูงมากเมื่อเทียบกับไมโครคอนโทรลเลอร์เบอร์อื่นๆ ที่ความถี่ของสัญญาณนาฬิกาเท่ากัน

2.1.1 คุณสมบัติทางเทคนิคของ PIC16F8XX

สามารถแบ่งออกเป็น 3 ส่วนคือ หน่วยประมวลผลกลาง (Central Processing Unit : CPU) ส่วนของเพอริเฟอรัล (Peripheral) และคุณสมบัติพิเศษอื่นๆ

1) คุณสมบัติทางเทคนิคของหน่วยประมวลผลกลางภายใน PIC16F8XX

หน่วยประมวลผลกลางเป็นแบบ RISC (Reduced Instruction Set Computer) คำสั่งมีขนาด 14 บิต (Bit)

1.1) คำสั่งใช้เวลาในการประมวลผลเพียง 1 ไซเคิล (Cycle) ของสัญญาณนาฬิกา ยกเว้นชุดคำสั่งที่ใช้ในการกระโดดจะใช้เวลา 2 ไซเคิลของสัญญาณนาฬิกา

1.2) ประมวลผลข้อมูลขนาด 8 บิต

1.3) มีสแต็ก (Stack) 8 ระดับ

1.4) โหมดการอ้างแอดเดรส (Address) 3 โหมดคือ แบบโดยตรง (Direct) แบบโดยอ้อม (Indirect) และแบบสัมพัทธ์ (Relative)

1.5) มีหน่วยความจำข้อมูล (Data Memory) เป็นแบบอีอีพรอม (EEPROM) สามารถลบและเขียนได้ด้วยสัญญาณไฟฟ้าและเก็บข้อมูลได้นาน 40 ปี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6) ขนาดของหน่วยความจำโปรแกรมซึ่งเป็นแบบแฟลชมีขนาด 8 กิโลเวิร์ด (1 เวิร์ดของ PIC16F876 มีขนาด 14 บิต) หน่วยความจำอีอีพรอมภายใน 256 ไบต์ (Byte) หน่วยความจำแรม (RAM) 368 ไบต์ (Byte) ซึ่งใช้เป็นรีจีสเตอร์ (Register)

2) คุณสมบัติทางเทคนิคของเพอริเฟอรัลใน PIC16F876

2.1) มีขาอินพุต / เอาต์พุต (Input / Output) 22 ขา (สำหรับ PIC16F876) สามารถกำหนดให้เป็นขาอินพุตหรือขาเอาต์พุตได้อย่างอิสระ

2.2) กระแสซิงก์ (Synchronous) สูงสุด 25 มิลลิแอมป์

2.3) กระแสซอร์ส (Source) สูงสุด 25 มิลลิแอมป์

2.4) มีเพาเวอร์ออนรีเซตในตัว (POR : Power On Reset)

2.5) มีเพาเวอร์อัปไทม์เมอร์ในตัว (PWRT : Power Up Timer)

2.6) มีออสซิลเลเตอร์สตาร์ตอัปไทม์เมอร์ (OST : Oscillator Start Up Timer)

2.7) มีวอตซ์ด็อกไทม์เมอร์ (WDT : Watch Dog Timer) พร้อมกับวงจรออสซิลเลเตอร์ RC ภายใน เพื่อช่วยให้การทำงานของไมโครคอนโทรลเลอร์มีความแน่นอนยิ่งขึ้น

2.8) ป้องกันการคัดลอกข้อมูลในหน่วยความจำโปรแกรม

2.9) มีโหมดประหยัดพลังงานหรือโหมดสลีป (Sleep Mode)

2.10) สามารถเลือกวงจรออสซิลเลเตอร์ที่ใช้กำหนดการทำงานได้

2.11) การเขียนข้อมูลสู่หน่วยความจำภายในไมโครคอนโทรลเลอร์ เป็นแบบอนุกรมผ่านขาใช้งานเพียง 2 ขา

2.12) ได้รับการพัฒนาภายใต้เทคโนโลยีซีมอสแฟลช (CMOS Flash Technology) และอีอีพรอมความเร็วสูง พลังงานต่ำ

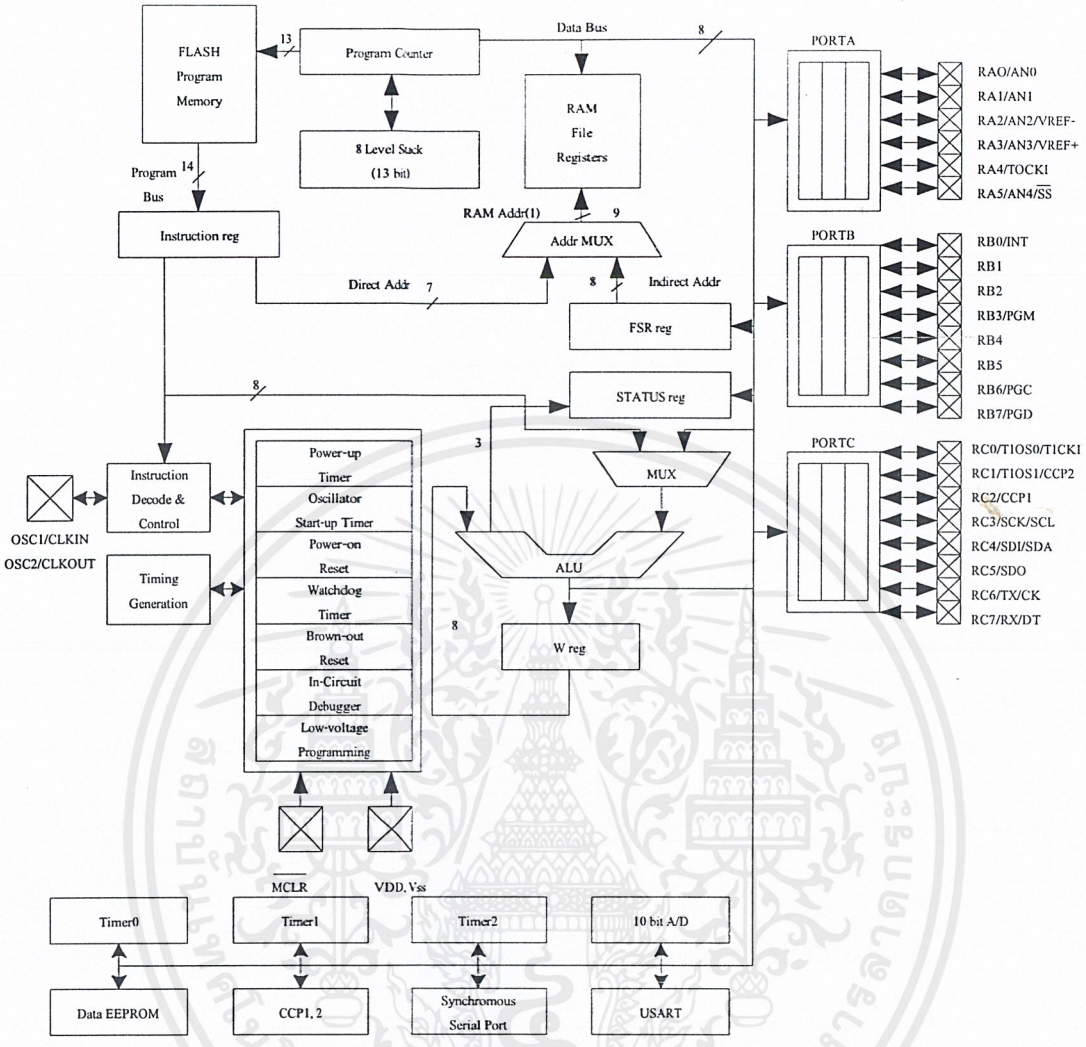
2.13) ย่านไฟเลี้ยง 2.0 - 6.0 โวลต์ (V)

2.14) ปริมาณการใช้กระแสไฟฟ้าน้อยกว่า 2 มิลลิแอมป์ ที่ไฟเลี้ยง +5 โวลต์ สัญญาณนาฬิกาความถี่ 4 เมกะเฮิรตซ์ 15 ไมโครแอมป์ (uA) ที่ไฟเลี้ยง +3 โวลต์ สัญญาณนาฬิกาที่ความถี่ 32 กิโลเฮิรตซ์ (KHz) น้อยกว่า 1 ไมโครแอมป์ ขณะสแตนด์บาย (Stand By)

2.1.2 สถาปัตยกรรมของ PIC16F8XX

ไมโครคอนโทรลเลอร์ PIC16F8XX ได้รับการบรรจุหน่วยประมวลผล หน่วยความจำ หน่วยคำนวณทางคณิตศาสตร์และหน่วยอินพุตเอาต์พุตไว้พร้อมทั้งยังมีไทม์เมอร์และวอตซ์ด็อกครบถ้วนสมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 สถาปัตยกรรมของไมโครคอนโทรลเลอร์ PIC16F876

1) PIC16F876 มีการจัดสรรหน่วยความจำดังนี้

1.1) หน่วยความจำโปรแกรมมีโครงสร้างเป็นหน่วยความจำแบบแฟลช ขนาด 8 กิโลเวิร์ด

1.2) หน่วยความจำข้อมูลเป็นหน่วยความจำแบบอีอีพรอมขนาด 256 ไบต์

1.3) หน่วยความจำแรมได้รับการกำหนดให้ทำงานเป็นรีจีสเตอร์กำหนดแฟ้มข้อมูล หรือรีจีสเตอร์ไฟล์ (Register File) ขนาด 368 ไบต์

การเข้าถึงหน่วยความจำของหน่วยประมวลผลกลาง (CPU) ภายในไมโครคอนโทรลเลอร์นี้สามารถทำได้ในลักษณะโดยตรง โดยอ้อม และแบบสัมพัทธ์ โดยมีรีจีสเตอร์ FSR (File Select Register) ทำหน้าที่ในการควบคุมการเข้าถึงหน่วยความจำ เมื่อไมโครคอนโทรลเลอร์ทำงานตามคำสั่งที่กำหนดให้ข้อมูลของชุดคำสั่งจะถูกนำไปเก็บไว้ในรีจีสเตอร์คำสั่ง (Instruction Register)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นจะถูกส่งต่อไปยังวงจรถอดรหัส เพื่อควบคุมไทม์เมอร์ทั้งหมดภายในไมโครคอนโทรลเลอร์ และยังส่งไปควบคุมหน่วยคำนวณทางคณิตศาสตร์โดยผ่านวงจรมัลติเพล็กซ์ (Multiplex) ด้วย

ใน PIC16F876 มีไทม์เมอร์/เคาน์เตอร์ขนาด 8 บิต 1 ตัว, ขนาด 16 บิต 2 ตัว ปรีสเกลเลอร์ (Prescaler) ที่สามารถโปรแกรมได้ สามารถรับสัญญาณอินพุตทั้งจากภายนอกและภายใน

หน่วยคำนวณทางคณิตศาสตร์ (Arithmetic Logic Unit : ALU) มีขนาด 8 บิต ในส่วนของพอร์ตอินพุตเอาต์พุตใน PIC16F876 มีด้วยกัน 3 พอร์ต (Port) คือ พอร์ต A พอร์ต B และพอร์ต C โดยที่พอร์ต A มี 6 บิตคือ RA0-RA5 ส่วนพอร์ต B มี 8 บิตคือ RB0-RB7 และพอร์ต C มี 8 บิต คือ RC0-RC7

2.1.3 การจัดขาของ PIC16F876

ไมโครคอนโทรลเลอร์ PIC16F876 สามารถจัดขาต่อใช้งานได้เป็น 4 กลุ่ม คือ

1) กลุ่มขาสัญญาณนาฬิกา

มี 2 ขา คือ OSC1 / CLKIN (ขา 9) และ OSC2 / CLKOUT (ขา 10)

2) กลุ่มขาควบคุม

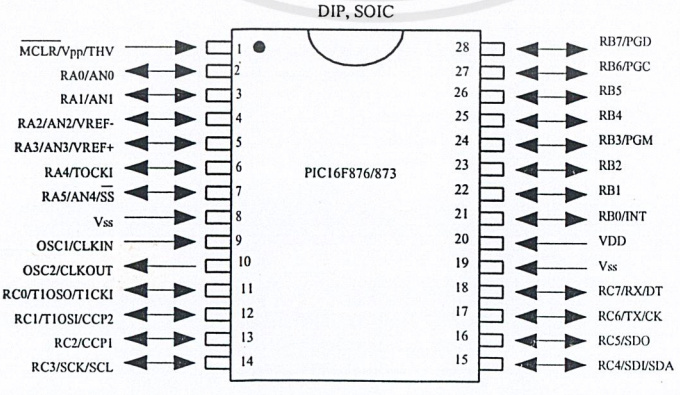
มี 1 ขา คือ MCLR (ขา 1)

3) กลุ่มขาพอร์ตอินพุตเอาต์พุต

มี 22 ขา แบ่งเป็นขาพอร์ต A 6 ขา ได้แก่ RA0-RA5 (ขา 2 ถึงขา 7) ขาพอร์ต B ได้แก่ขา RB0-RB7 (ขา 21 ถึงขา 28) และขาพอร์ต C ได้แก่ขา RC0-RC7 (ขา 11 ถึงขา 18)

4) กลุ่มขาไฟเลี้ยง

มี 3 ขา คือ ขา V_{SS} (ขา 8 และขา 19) หรือขาต่อกราวด์ และขา V_{DD} (ขา 20) หรือขาต่อไฟเลี้ยง ปกติใช้ +5 โวลต์



รูปที่ 2.2 การจัดขาต่อใช้งานของ PIC16F876

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.4 การป้อนสัญญาณนาฬิกาให้แก่ PIC16F876

การป้อนสัญญาณนาฬิกาให้แก่ PIC16F876 ทำได้ 3 วิธีใหญ่ ๆ โดยใช้อุปกรณ์ 3 รูปแบบ คือ ตัวต้านทานร่วมกับตัวเก็บประจุ (RC Clock Circuit) เซรามิกเรโซเนเตอร์ (Ceramic Rasonater) และคริสตอล (Crystal)

1) **การป้อนสัญญาณนาฬิกาโดยใช้ตัวต้านทานและตัวเก็บประจุ (RC Clock Circuit)**

การป้อนสัญญาณนาฬิกานี้เหมาะสำหรับงานที่ไม่เข้มงวดเรื่องความแม่นยำและเสถียรภาพของความถี่ของสัญญาณนาฬิกา เป็นวิธีการที่ง่ายที่สุด และใช้งบประมาณต่ำที่สุด

2) การป้อนสัญญาณนาฬิกาโดยใช้คริสตอล

วงจรกำเนิดสัญญาณนาฬิกาแบบนี้มีความเที่ยงตรงสูงมาก ส่งผลให้การคำนวณเกี่ยวกับการหน่วงเวลาในการเขียนโปรแกรมกระทำได้อย่างแม่นยำมากขึ้น โดยสัญญาณนาฬิกาที่ป้อนให้แก่ ไมโครคอนโทรลเลอร์ PIC16F8XX จะถูกหารลง 4 เท่า

3) การป้อนสัญญาณนาฬิกาโดยใช้เซรามิกเรโซเนเตอร์

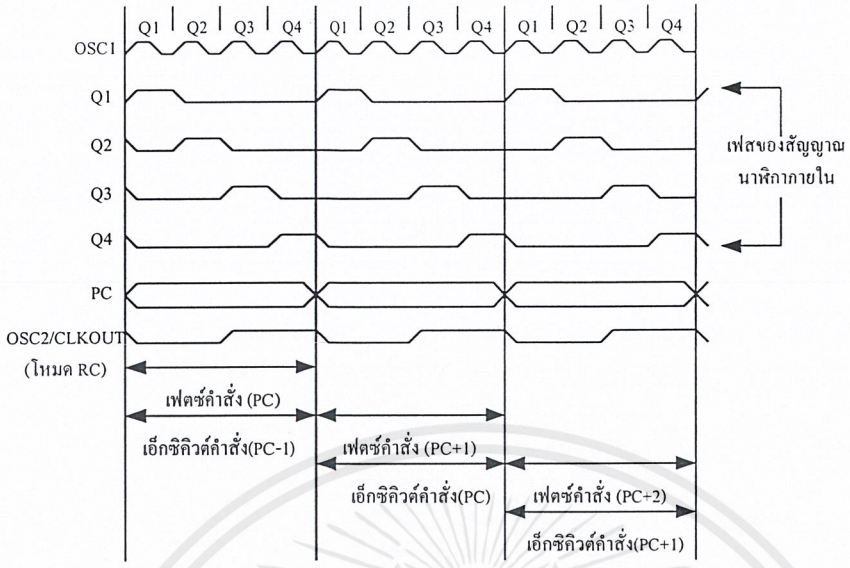
การป้อนสัญญาณนาฬิกาโดยใช้เซรามิกเรโซเนเตอร์นี้ใช้เมื่อต้องการให้รูปร่างของสัญญาณนาฬิกามีคุณภาพและความเที่ยงตรงสูงๆ

2.1.5 จังหวะสัญญาณนาฬิกาและไซเคิลการทำงานของ PIC16F8XX

สัญญาณนาฬิกาอินพุตของ OSC1 (Oscilator1) จะถูกหารด้วย 4 แล้วแบ่งเป็น 4 ช่วง กำหนดเป็น ช่วงสัญญาณนาฬิกา Q1 ช่วงสัญญาณนาฬิกา Q2 ช่วงสัญญาณนาฬิกา Q3 และช่วงสัญญาณนาฬิกา Q4 โปรแกรมเคาน์เตอร์ (Program Counter) ภายในซีพียูจะเพิ่มค่าขึ้นทุกๆ ครั้งที่ช่วงสัญญาณนาฬิกา Q1 เมื่อโปรแกรมเคาน์เตอร์เพิ่มค่าขึ้นคำสั่งจะถูกเฟตช์จากหน่วยความจำโปรแกรม จากนั้นซีพียูทำการแลตช์ (Latch) ข้อมูลคำสั่งนั้นไว้ในรีจีสเตอร์คำสั่ง ที่ช่วงสัญญาณนาฬิกา Q4 คำสั่งจะถูกถอดรหัสและเอ็กซิคิวต์ จนเสร็จสิ้นภายในช่วงสัญญาณนาฬิกา Q1-Q4 หรือภายใน 1 ไซเคิลของสัญญาณนาฬิกา

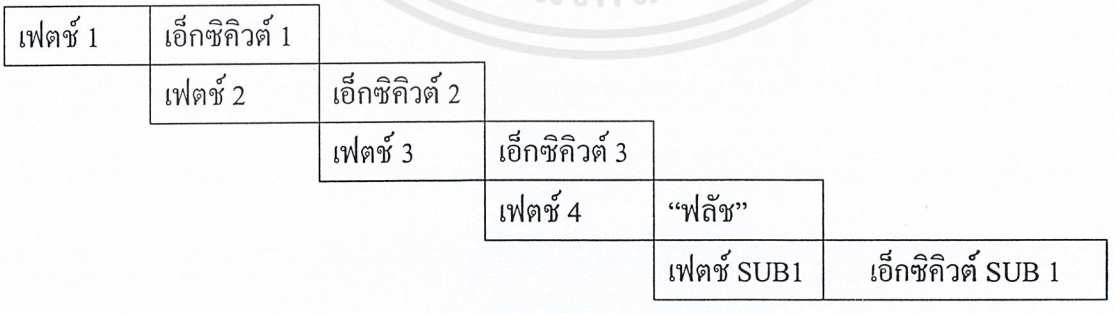
แผนผังการทำงานแสดงความสัมพันธ์ระหว่างจังหวะของสัญญาณนาฬิกากับการประมวลผลคำสั่งของไมโครคอนโทรลเลอร์ PIC16F8XX พบว่า ไมโครคอนโทรลเลอร์ PIC16F8XX จะทำงานโดยใช้สัญญาณนาฬิกาเพียง 1 ลูกต่อหนึ่งคำสั่ง แต่สัญญาณนาฬิกาเพียง 1 ลูกจะถูกแบ่งออกเป็น 4 ช่วงเพื่อกำหนดจังหวะการทำงานของซีพียูให้มีความชัดเจนมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 แผนผังแสดงจังหวะการทำงานของ PIC16F8XX

ไซเคิลการทำงาน (Instruction Cycle) ของ PIC16F8XX แบ่งเป็น 2 ไซเคิล คือ การเฟตช์และการเอ็กซิคิวต์ ประกอบด้วย 4 คิวไซเคิล (Q-cycle : Quadrature Cycle) คือ Q1 Q2 Q3 และ Q4 การเฟตช์และเอ็กซิคิวต์คำสั่งของไมโครคอนโทรลเลอร์ จะมีลักษณะเป็นแบบไปป์ไลน์ (PipeLine : การทำงานที่เหลื่อมล้ำกันในแต่ละขั้นตอนการทำงานของคอมพิวเตอร์ เพื่อที่จะทำให้คอมพิวเตอร์ทำงานได้เร็วขึ้น) กล่าวคือ เมื่อซีพียูกระทำคำสั่งหนึ่ง โดยที่คำสั่งนั้นมี 2 ขั้นตอน เมื่อทำงานไปแล้ว 1 ขั้นตอน คือขั้นตอนการเฟตช์คำสั่ง ซีพียูจะเริ่มเฟตช์คำสั่งต่อไปทันที พร้อมๆ กับการ เอ็กซิคิวต์คำสั่งก่อนหน้านี้



รูปที่ 2.4 การทำงานแบบไปป์ไลน์ที่ใช้ใน PIC16F8XX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างโปรแกรมเพื่อแสดงการเฟตช์คำสั่งต่อไปทันที พร้อมๆกับการเอ็กซิคิวต์คำสั่ง

```
movlw 0x55    ; นำค่า 0x55 มาเก็บไว้ที่รีจีสเตอร์ W
movwf PORTB ; นำข้อมูลไปเก็บไว้ที่รีจีสเตอร์ PORT B
call    SUB1  ; เรียกโปรแกรมย่อยที่มีชื่อว่า SUB1
bsf     PORTA,3       ; เป็นการเซตข้อมูลใน PORT A บิตที่ 3
```

ช่วงเวลา "ฟลัช" (Flush) หมายถึง ช่วงเวลาที่ซีพียูทำงานคำสั่งก่อนหน้ายังไม่เสร็จสิ้น เนื่องจากต้องทำการค้นหาตำแหน่งแอดเดรส (Address) ของโปรแกรมย่อยที่ต้องกระโดดไปทำงาน ช่วงเวลาแบบนี้จะเกิดขึ้น เมื่อซีพียูกระทำคำสั่งเกี่ยวกับการกระโดด ในขณะเดียวกันซีพียูก็จะทำการเฟตช์คำสั่งต่อไปพร้อมๆ กัน ทำให้ซีพียูสามารถทำงานได้เสมือนว่าใน 1 คำสั่งการทำงานใช้เวลา 1 ไซเคิลของสัญญานาฬิกา เทคนิคลักษณะนี้เรียกว่า "ไปป์ไลน์" (Pipeline) ซึ่งเป็นที่นิยมมากในไมโครโปรเซสเซอร์ (Microprocessor) สมัยใหม่

ไซเคิลการเฟตช์คำสั่งเริ่มต้น ในทุกๆ คิวไซเคิลที่ 1 พร้อมกับคำสั่งข้อมูลที่ได้รับการเฟตช์จะถูกนำเข้าไปเก็บไว้ในรีจีสเตอร์คำสั่งในคิวไซเคิลที่ 1 (Q1) จากนั้นคำสั่งจะได้รับการถอดรหัสและนำไปดำเนินการจนเสร็จสิ้นภายในคิวไซเคิลที่ 2-4 (Q2-Q4) โดยกำหนดให้ในคิวไซเคิลที่ 2 หน่วยความจำข้อมูลที่ใช้เก็บค่าโอเปอร์แรนด์ (Operand) จะถูกอ่าน และถูกเขียนข้อมูลใหม่เข้าไปแทนที่ในคิวไซเคิลที่ 4

2.2 การจัดการหน่วยความจำของ PIC16F8XX

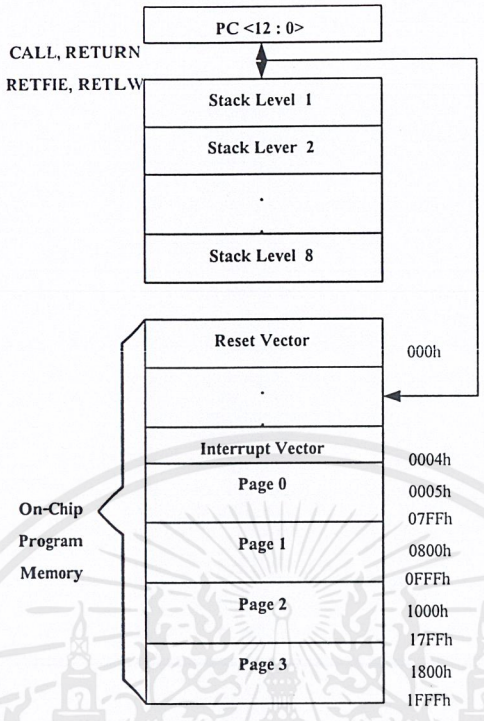
การจัดการหน่วยความจำของ PIC16F8XX แบ่งออกเป็น 3 ส่วน คือ หน่วยความจำโปรแกรม หน่วยความจำข้อมูลและหน่วยความจำส่วนของ EEPROM

2.2.1 หน่วยความจำโปรแกรม

หน่วยความจำโปรแกรมของ PIC16F8XX เป็นหน่วยความจำแบบแฟลช ซึ่งสามารถที่จะโปรแกรมลงบนหน่วยความจำนี้ได้ทั้งในโหมดโปรแกรม และในขณะทำงานปกติ

PIC16F8XX มี โปรแกรมเคาน์เตอร์ (Program Counter : PC) ขนาด 13 บิต ที่สามารถอ้างตำแหน่งหน่วยความจำได้ 8K X 14 ตำแหน่ง โดยมีตำแหน่งรีเซตเวคเตอร์ (Reset Vector) อยู่ที่ตำแหน่ง 0000h และตำแหน่งอินเตอร์รัพต์เวคเตอร์ (Interrupt Vector) อยู่ที่ตำแหน่ง 0004h และมีความลึกของสแต็ก (Stack) 8 ระดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 การจัดสรรหน่วยความจำโปรแกรมของ PIC16F8XX

2.2.2 หน่วยความจำข้อมูล

พื้นที่ของหน่วยความจำข้อมูลได้รับการจัดสรรเป็น 2 ส่วน คือ พื้นที่ของรีจีสเตอร์ฟังก์ชั่นพิเศษ (Special Function Register : SFR) และพื้นที่ของรีจีสเตอร์ใช้งานทั่วไป (General Function Register : GPR) การจัดสรรหน่วยความจำข้อมูลทั้ง SFR และ GPR จะถูกแบ่งออกเป็น 4 แบงค์ (Bank) ซึ่งเป็นที่อยู่ของรีจีสเตอร์ใช้งานทั่วไป และรีจีสเตอร์ฟังก์ชั่นพิเศษ เราสามารถเปลี่ยนแบงค์ได้โดยการเลือกที่บิต RP0 และ RP1 แต่ละแบงค์มีถึง 128 ไบต์ และส่วนต้นๆ ของแต่ละแบงค์ จะเป็นที่อยู่ของรีจีสเตอร์ฟังก์ชั่นพิเศษ และส่วนถัดไปจะเป็นที่อยู่ของ รีจีสเตอร์ใช้งานทั่วไป ซึ่งถูกใช้เหมือนกับแรมทั่วไป

การเข้าถึงหน่วยความจำข้อมูลสามารถทำได้ทั้งแบบโดยตรง (Direct) หรือแบบโดยอ้อม (Indirect) ผ่านทางรีจีสเตอร์เลือกแฟ้มข้อมูล (File Select Register : FSR)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 รีจีสเตอร์ควบคุมของ PIC16F8XX

ใน PIC16F8XX มีรีจีสเตอร์ควบคุมที่มีบทบาทสำคัญอยู่ 6 ตัว คือ STATUS OPTION INTCON PCL PCLATH และ W

2.3.1 รีจีสเตอร์ STATUS

เป็นรีจีสเตอร์ที่ใช้แสดงสถานะทางคณิตศาสตร์ของหน่วยประมวลผลทางคณิตศาสตร์ สถานะการทำงานของ PIC16F8XX และใช้เป็นตัวกำหนดการเลือกแบงค์ของหน่วยความจำข้อมูล การเข้าถึงรีจีสเตอร์ STATUS เพื่ออ่านและเขียนข้อมูลสามารถกระทำได้ด้วยวิธีการเดียวกับการอ่านและเขียนรีจีสเตอร์ตัวอื่นๆ

ตารางที่ 2.1 รายละเอียดของบิตต่างๆ ในรีจีสเตอร์ STATUS

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0	R : อ่านค่าได้ W : เขียนค่าได้ U : ไม่ใช้งาน "0" -n ค่าที่เกิดหลังเพาเวอร์ออนรีเซต
R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-X	R/W-X	R/W-X	
IRP	RP1	RP0	TO	PD	Z	DC	C	

บิต 7 : IRP บิตเลือกแบงค์ของรีจีสเตอร์ แต่ใน PIC16F8XX ต้องกำหนดให้เป็น "0"

บิต 6-5 : RP1 และ RP0 บิตเลือกแบงค์ของรีจีสเตอร์ เข้าถึงโดยการอ้างแอดเดรสโดยตรง (Direct Addressing) การเลือกแบงค์ของรีจีสเตอร์สามารถทำได้โดยการกำหนดบิต RP0 และ RP1

00 = แบงค์ 0 (แอดเดรส 0000H-007FH)

01 = แบงค์ 1 (แอดเดรส 0080H-00FFH)

10 = แบงค์ 2 (แอดเดรส 0100H-017FH)

11 = แบงค์ 3 (แอดเดรส 0180H-01FFH)

บิต 4 : TO (Time Out Bit) บิตขอบเขตเวลา บิตนี้สามารถอ่านค่าได้อย่างเดียวเท่านั้น

"1" เมื่อจ่ายไฟเลี้ยงให้ PIC16F8XX หรือมีการกระทำคำสั่ง Clrwdt หรือ Sleep

"0" เมื่อวอตซ์ด็อกไทม์เมอร์ภายในทำงานครบเวลาที่กำหนดหรือเกิดไทม์เอาต์

บิต 3 : PD (Power Down Bit) บิตเพาเวอร์ดาวน์ ทำงานในโหมดสลีป (Sleep)

"1" เมื่อมีการจ่ายไฟเลี้ยงให้แก่ PIC16F8XX หรือเมื่อมีการกระทำคำสั่ง Clrwdt

"0" เมื่อมีการกระทำคำสั่ง Sleep

บิต 2 : Z (Zero Bit) บิตศูนย์เป็นบิตแสดงผลการกระทำทางคณิตศาสตร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

"1" เมื่อมีการกระทำคำสั่งทางคณิตศาสตร์และลอจิกแล้ว เกิดค่าศูนย์ (0) ขึ้น

"0" เมื่อมีการกระทำคำสั่งทางคณิตศาสตร์และลอจิกแล้วไม่เกิดค่าศูนย์ (0) ขึ้น

บิต 1 : DC (Digit Carry / Borrow Bit) บิตทดหรือบิตระหว่างหลัก เป็นบิตแสดงผลทางคณิตศาสตร์

ในกรณีที่กระทำคำสั่ง Addwf และ Addlw บิต DC จะเกิดผลดังนี้

"1" เมื่อมีเกิดการทดจากบิตล่างที่ 4 ไปยังกลุ่มบิตบน (จากบิตที่ 4 ไปยังบิตที่ 5)

"0" เมื่อไม่เกิดการทดจากบิตล่างที่ 4 ไปยังกลุ่มบิตบน (จากบิตที่ 4 ไปยังบิตที่ 5)

ในกรณีที่กระทำคำสั่ง Subwf และ Sublw บิต DC จะเกิดผลดังนี้

"1" เมื่อไม่เกิดการยืมจากบิตล่างที่ 4 โดยกลุ่มบิตบน (บิตที่ 5 ยืมค่าจากบิตที่ 4)

"0" เมื่อเกิดการยืมจากบิตล่างที่ 4 โดยกลุ่มบิตบน (บิตที่ 5 ยืมค่าจากบิตที่ 4)

บิต 0 : C (Carry Bit / Borrow Bit) บิตทดหรือยืม ทำงานคล้ายกับบิต DC แต่จะเกิดการเปลี่ยนแปลงค่าก็ต่อเมื่อเกิดการทดหรือยืมค่าจากบิตนัยสำคัญสูงสุด (MSB : Most Significant Bit)

ในกรณีที่กระทำคำสั่ง Addwf และ Addlw บิต C จะเกิดผลดังนี้

"1" เมื่อมีการกระทำคำสั่งทางคณิตศาสตร์และลอจิกแล้ว เกิดการทด

"0" เมื่อมีการกระทำคำสั่งทางคณิตศาสตร์และลอจิกแล้ว ไม่เกิดการทด

ในกรณีที่กระทำคำสั่ง Subwf และ Sublw บิต C จะเกิดผลดังนี้

"1" เมื่อมีการกระทำคำสั่งทางคณิตศาสตร์และลอจิกแล้ว ไม่เกิดการยืม

"0" เมื่อมีการกระทำคำสั่งทางคณิตศาสตร์และลอจิกแล้ว เกิดการยืม

2.3.2 รีจีสเตอร์ OPTION

เป็นรีจีสเตอร์ที่ควบคุมการอินเตอร์รัพต์จากสัญญาณภายนอก

ตารางที่ 2.2 รายละเอียดของบิตต่างๆ ในรีจีสเตอร์ OPTION

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0	R :อ่านค่าได้ W : เขียนค่าได้ U : ไม่ใช้งาน "0" -nค่าที่เกิดหลังเพาเวอร์ออนรีเซต
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	
RBPU	INTEG	TOCS	TOSE	PSA	PS2	PS1	PS0	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต 7 : RBPU (Port B Pull Up Enable Bit) ใช้ในการอีนาเบิลการพูลอัปที่ขาพอร์ต B

"1" ดิสเอเบิลการพูลอัปที่ขาพอร์ต B

"0" อีนาเบิลการพูลอัปที่ขาพอร์ต B

บิต 6 : INTEG (Interrupt Edge Select Bit) บิตเลือกขอบขาของสัญญาณอินเตอร์รัพต์จากภายนอกใช้เลือกขอบขาของสัญญาณที่ทำให้เกิดการอินเตอร์รัพต์ที่ขา RB0 / INT

"1" เลือกขอบขาขึ้นของสัญญาณ

"0" เลือกขอบขาลงของสัญญาณ

บิต 5 : TOCS (TMRO Clock Source Select Bit) บิตเลือกแหล่งจ่ายสัญญาณนาฬิกาให้แก่ไทม์เมอร์เคาน์เตอร์

"1" เลือกจากการเปลี่ยนแปลงระดับสัญญาณที่ขา RA4 / TOCKI

"0" เลือกจากสัญญาณนาฬิกาที่ใช้กำหนดไซเคิลการทำงานภายใน PIC16F8XX

บิต 4 : TOSE (TMRO Source Edge Select Bit) ใช้เลือกขอบขาของสัญญาณนาฬิกาที่จ่ายเข้ามายังขา RA4 / TOCKI ในกรณีที่เลือกแหล่งจ่ายสัญญาณนาฬิกาให้ไทม์เมอร์เคาน์เตอร์ผ่านทางขา RA4 / TOCKI โดยการกำหนดบิต TOCS ให้เป็น "1"

"1" TMRO เพิ่มค่าเมื่อเกิดการเปลี่ยนแปลงระดับสัญญาณจากสูงมาต่ำ

"0" TMRO เพิ่มค่าเมื่อเกิดการเปลี่ยนแปลงระดับสัญญาณ จากต่ำมาสูง

บิต 3 : PSA (Prescaler Assignment Bit) บิตกำหนดการทำงานของปรีสเกลเลอร์

"1" ปรีสเกลเลอร์ทำงานร่วมกับวอตซ์ด็อกไทม์เมอร์ เรียกว่า โพสต์สเกลเลอร์

"0" กำหนดให้ปรีสเกลเลอร์ทำงานร่วมกับ TMRO

บิต 2-0 :PS2-PS0 (Prescaler Rate Select bit) บิตเลือกอัตราส่วนของปรีสเกลเลอร์ เป็นบิตที่ใช้ในการกำหนดอัตราส่วนในการทำงานของปรีสเกลเลอร์ เมื่อทำงานกับ TMRO และวอตซ์ด็อกไทม์เมอร์ อัตราส่วนนี้จะมีค่าไม่เท่ากัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 การกำหนดอัตราส่วนของปรีสเกลเลอร์

PS2	PS1	PS0	อัตราส่วนเมื่อ ทำงานกับ WDT	อัตราส่วนเมื่อ ทำงานกับ TMRO
0	0	0	1 : 1	1 : 2
0	0	1	1 : 2	1 : 4
0	1	0	1 : 4	1 : 8
0	1	1	1 : 8	1 : 16
1	0	0	1 : 16	1 : 32
1	0	1	1 : 32	1 : 64
1	1	0	1 : 64	1 : 128
1	1	1	1 : 128	1 : 256

2.3.3 รีจีสเตอร์ INTCON

เป็นรีจีสเตอร์ที่เก็บค่าบิตของการอีนาเบิลสัญญาณอินเตอร์รัพต์ มีแอดเดรสอยู่ที่ 0x0B มีบทบาทสำคัญมากในเรื่องการอินเตอร์รัพต์

ตารางที่ 2.4 รายละเอียดของบิตต่างๆ ในรีจีสเตอร์ INTCON

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0	R :อ่านค่าได้ W : เขียนค่าได้ U : ไม่ใช้งาน "0" -n ค่าที่เกิดหลัง เพาเวอร์ออนรีเซต
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x	
GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	

บิต 7 : GIE (Global Interrupt Enable Bit) บิตอีนาเบิลการอินเตอร์รัพต์

"1" อีนาเบิลการอินเตอร์รัพต์ทั้งหมด

"0" ดิสเอเบิลการอินเตอร์รัพต์ทั้งหมด

บิต 6 : EEIE (EEPROM Write Complete Interrupt Enable Bit) บิตอีนาเบิลอินเตอร์รัพต์เมื่อเกิดการเขียนข้อมูลในหน่วยความจำข้อมูลอีอีพรอมเสร็จสมบูรณ์

"1" อีนาเบิลการอินเตอร์รัพต์

"0" ดิสเอเบิลการอินเตอร์รัพต์

บิต 5 : TOIE (TMRO Overflow Interrupt Enable Bit)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

"1" อีนาเบิลการอินเตอร์รัพต์

"0" ดิสเอเบิลการอินเตอร์รัพต์

บิต 4 : INTE (RBO / INT Interrupt Enable Bit)

"1" อีนาเบิลการอินเตอร์รัพต์

"0" ดิสเอเบิลการอินเตอร์รัพต์

บิต 3 : RBIE (Port B Change Interrupt Enable Bit) ระดับสัญญาณที่ขาพอร์ต B บิตที่ 4-7

"1" อีนาเบิลการอินเตอร์รัพต์

"0" ดิสเอเบิลการอินเตอร์รัพต์

บิต 2 : TOIF (TMRO Overflow Interrupt Flag)

"1" แจ้งว่าเกิดการอินเตอร์รัพต์อันเนื่องมาจาก TMRO เกิดการโอเวอร์โฟลว์

"0" ไม่เกิดการอินเตอร์รัพต์อันเนื่องมาจาก TMRO เกิดการโอเวอร์โฟลว์

บิต 1 : INTF (RBO / INT Interrupt Flag)

"1" แจ้งว่าเกิดการอินเตอร์รัพต์อันเนื่องมาจากสัญญาณภายนอกที่ขา RB0 / INT

"0" ไม่เกิดการอินเตอร์รัพต์อันเนื่องมาจากสัญญาณภายนอกที่ขา RB0 / INT

บิต 0 : RBIF (Port B Change Interrupt Flag) ระดับสัญญาณที่ขาพอร์ต B บิตที่ 4-7

"1" เกิดการอินเตอร์รัพต์จากการเปลี่ยนระดับสัญญาณที่ขาพอร์ต B บิตที่ 4-7

"0" ไม่เกิดการอินเตอร์รัพต์จากการเปลี่ยนระดับสัญญาณที่ขาพอร์ต B บิตที่ 4-7

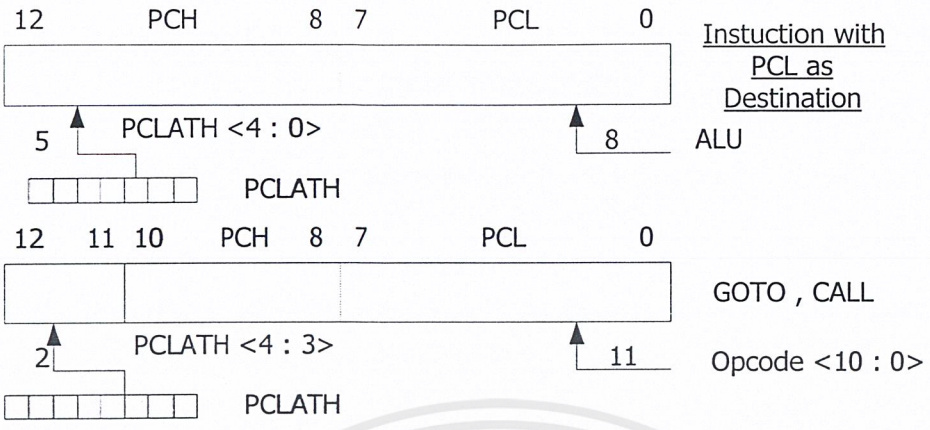
2.3.4 รีจีสเตอร์โปรแกรมเคาน์เตอร์ PCL และ PCLATH

โปรแกรมเคาน์เตอร์ (Program Counter : PC) เป็นรีจีสเตอร์ที่มีหน้าที่ชี้ตำแหน่งแอดเดรสต่อไปของหน่วยความจำโปรแกรมที่ซีพียูจะต้องไปทำงาน

โปรแกรมเคาน์เตอร์หรือ PC ใน PIC16F8XX มีขนาด 13 บิต แบ่งเป็น 2 ส่วน คือ รีจีสเตอร์โปรแกรมเคาน์เตอร์ไบต์ต่ำ หรือ PCL (Program Counter Low Byte) ซึ่งในส่วนนี้มีขนาด 8 บิต สามารถอ่านและเขียนค่าได้โดยตรง ในขณะที่อีกส่วนหนึ่งมีขนาด 5 บิต ไม่สามารถอ่านหรือเขียนข้อมูลได้โดยตรง ต้องอาศัยการเขียนและอ่านค่าผ่านรีจีสเตอร์ PCLATH โดยรีจีสเตอร์ PCLATH จะทำการเก็บค่าของ 5 บิตบนโปรแกรมเคาน์เตอร์ไว้และถ่ายทอดลงสู่ 5 บิตบนของโปรแกรมเคาน์เตอร์ไบต์สูง (PCH) ก็ต่อเมื่อโปรแกรมเคาน์เตอร์มีการโหลดค่าใหม่เข้ามา ซึ่งจะเกิดขึ้นเมื่อกระทำคำสั่ง CALL หรือ GOTO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



รูปที่ 2.6 การถ่ายทอดข้อมูลภายในโปรแกรมเคาน์เตอร์

2.3.5 สแต็ก (Stack)

ในไมโครคอนโทรลเลอร์ PIC16F8XX ได้จัดสรรสแต็กหรือพื้นที่ในหน่วยความจำ เพื่อใช้ในการเก็บค่าของโปรแกรมเคาน์เตอร์ชั่วคราวไว้ 8 ระดับ หากมีการพุชเป็นครั้งที่ 9 ซีพียูจะนำค่าของ PC ในครั้งที่ 9 นี้เก็บลงในสแต็กที่เก็บค่า PC ในครั้งที่ 1

2.3.6 รีจีสเตอร์ W

ในตัวไมโครคอนโทรลเลอร์ PIC16F8XX มีรีจีสเตอร์ที่ใช้ในการทำงานหลักคือ รีจีสเตอร์ W หากเปรียบเทียบกับ ไมโครคอนโทรลเลอร์เบอร์อื่นๆ รีจีสเตอร์ W เทียบได้กับแอกคิวมูเลเตอร์ (Accumulator) เมื่อ PIC16F8XX กระทำคำสั่งทางคณิตศาสตร์ รีจีสเตอร์ W จะเป็นรีจีสเตอร์ที่ซีพียูติดต่อด้วยการโอนข้อมูล หรือการตรวจสอบข้อมูลจะกระทำที่รีจีสเตอร์ W

2.3.7 รีจีสเตอร์ไฟล์ (File Register)

เนื่องจาก PIC16F8XX มีรีจีสเตอร์ที่เกี่ยวข้องอยู่หลายตัว จึงมีการจัดรวบรวมรีจีสเตอร์ทั้งหมดที่ต้องใช้การเข้าถึงแบบโดยอ้อมไว้ในลักษณะแฟ้มข้อมูล

รีจีสเตอร์ได้รับการนำมารวมไว้มีชื่อว่า รีจีสเตอร์ไฟล์ (Register File) มีขนาด 8 บิต และมีรีจีสเตอร์ฟังก์ชันพิเศษ ที่ถูกกำหนดหน้าที่และตำแหน่งไว้แล้ว และอีก 68 ตัวจะได้รับการกำหนดให้ใช้งานอย่างอิสระ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

43163

2.4 ชุดคำสั่งของ PIC16F86 และการเข้าถึงรีจีสเตอร์

2.4.1 กลุ่มคำสั่งจัดการข้อมูลระดับไบต์กับรีจีสเตอร์ไฟล์ (Byte Oriented File Register Operations)

เป็นกลุ่มคำสั่งที่เกี่ยวข้องกับรีจีสเตอร์ไฟล์โดยตรงหรือโดยอ้อม มีทั้งสิ้น 18 คำสั่ง

2.4.2 กลุ่มคำสั่งจัดการข้อมูลระดับบิตกับรีจีสเตอร์ไฟล์ (Bit Oriented File Register Operations)

เป็นกลุ่มคำสั่งที่ต้องเกี่ยวข้องหรือกระทำกับรีจีสเตอร์ไฟล์โดยตรงหรือโดยอ้อม โดยขนาดของข้อมูลที่ต้องประมวลผลอยู่ในระดับบิต มีทั้งสิ้น 4 คำสั่ง

2.4.3 กลุ่มคำสั่งจัดการกับค่าคงที่และควบคุมการทำงาน (Literal and Control Operations)

เป็นกลุ่มคำสั่งที่ผสมกันระหว่างการประมวลผลกับค่าคงที่ คำสั่งเกี่ยวกับการกระโดดไปยังโปรแกรมย่อยและออกจากโปรแกรมย่อย และคำสั่งกำหนดโหมดการทำงาน มีทั้งสิ้น 13 คำสั่ง

ตารางที่ 2.5 สรุปการแบ่งกลุ่มคำสั่งของ PIC16F8XX ตามการกำหนดโดย MPASM

กลุ่มคำสั่งจัดการข้อมูลระดับไบต์ของรีจีสเตอร์ไฟล์					
นีโมนิก โอเปอร์แรนด์	รายละเอียด	ไซเคิล	ออปโค้ด 14 บิต	บิตสถานะที่เกี่ยวข้อง	หมายเหตุ
Addwf f, d	W + f	1	00 0111 dfff ffff	C, DC, Z	1
Andwf f,d	W AND f	1	00 0101 dfff ffff	Z	1
Clrf f	เคลียร์ค่าของ f	1	00 0001 1fff ffff	Z	1
Clrf -	เคลียร์ค่าของ W	1	00 0001 0000 0011	Z	1
Comf f, d	คอมพลีเมนต์ f	1	00 1001 dfff ffff	Z	1
Decf f, d	ลดค่า f	1	00 0011 dfff ffff	Z	1
Decfsz f, d	ลดค่า f ข้ามหนึ่งแอดเครส ถ้าค่าเป็น 0	1 (2)	00 1011 dfff ffff	ไม่มี	1, 2
incf f, d	เพิ่มค่า f	1	00 1010 dfff ffff	Z	1
incfsz f, d	เพิ่มค่า f ข้ามหนึ่งแอดเครส ถ้าค่าเป็น 0	1 (2)	00 1111 dfff ffff	ไม่มี	1, 2
iorwf f, d	W OR f	1	00 0100 dfff ffff	Z	1
movf f, d	โอนย้ายข้อมูลไปยัง f	1	00 1000 dfff fff	Z	1
โอนย้ายข้อมูลจาก W ไปยัง f		1	00 0000 1fff ffff	ไม่มี	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.5 (ต่อ) สรุปการแบ่งกลุ่มคำสั่งของ PIC16F8XX ตามการกำหนดโดย MPASM

กลุ่มคำสั่งจัดการข้อมูลระดับไบต์ของรีจิสเตอร์ไฟล์					
nop -	ไม่มีการทำงาน	1	00 0000 0x00 0000	ไม่มี	
rlf f, d	เลื่อนข้อมูลไปทางซ้าย 1 บิตผ่านบิต C	1	00 1101 dfff ffff	C	1
rrf f, d	เลื่อนข้อมูลไปทางขวา 1 บิตผ่านบิต C	1	00 1101 dfff ffff	C	1
subwf f, d	f-W	1	00 0010 dfff ffff	C,DC,Z	1
swapf f, d	สลับค่า 4 บิตบนกับ 4 บิตล่างของ f	1	00 1110 dfff ffff	ไม่มี	1
xorwf f, d	W OR f	1	00 0110 dfff ffff	Z	1
กลุ่มคำสั่งจัดการข้อมูลระดับบิตของรีจิสเตอร์ไฟล์					
นีโมนิก โอเปอร์แรนด์	รายละเอียด	ไซเคิล	ออปโค้ด 14 บิต	บิตสถานะที่เกี่ยวข้อง	หมายเหตุ
bcf f, b	เคลียร์บิตของ f	1	01 00bb bfff ffff	ไม่มี	1
bsf f, b	เซตบิตของ f	1	01 01bb bfff ffff	ไม่มี	1
btfsc f, b	ตรวจสอบบิต ข้ามหนึ่งแอดเดรสถ้าเป็น "0"	1 (2)	01 10bb bfff ffff	ไม่มี	2
btfss f, b	ตรวจสอบบิต ข้ามหนึ่งแอดเดรสถ้าเป็น "1"	1 (2)	01 11bb bfff ffff	ไม่มี	2
กลุ่มคำสั่งจัดการกับข้อมูลค่าคงที่และควบคุมการทำงาน					
นีโมนิก โอเปอร์แรนด์	รายละเอียด	ไซเคิล	ออปโค้ด 4 บิต	บิตสถานะที่เกี่ยวข้อง	หมายเหตุ
Addlw k	ค่าคงที่ + W	1	11 111x kkkk kkkk	C, DC, Z	
Andlw k	ค่าคงที่ AND W	1	11 1001 kkkk kkkk	Z	
call k	เรียกโปรแกรมย่อย	2	10 0kkk kkkk kkkk	ไม่มี	
Clrwdt -	เคลียร์ค่าของ WDT	1	00 0000 0110 0100	TO, PD	
goto k	กระโดดไปยังแอดเดรสที่กำหนด	2	10 1kkk kkkk kkkk	ไม่มี	
iorlw k	ค่าคงที่ OR W	1	11 1000 kkkk kkkk	Z	
Movlw k	นำค่าคงที่ไปเก็บไว้ใน W	1	11 00xx kkkk kkkk	ไม่มี	
retfie -	ออกจากโปรแกรมย่อยบริการอินเตอร์รัพต์	2	00 0000 0000 1001	ไม่มี	
retlw k	กลับสู่โปรแกรมด้วยค่าคงที่ใน W	2	11 01xx kkkk kkkk	ไม่มี	
Return -	ออกจากโปรแกรมย่อย	2	00 0000 0000 1000	ไม่มี	
sleep -	เข้าสู่โหมดสลีป	1	00 0000 0110 0011	TO, PD	
Sublw k	ค่าคงที่ – W	1	11 110x kkkk kkkk	C, DC, Z	
Xorlw k	ค่าคงที่ XOR W	1	11 1010 kkkk kkkk	Z	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความหมายของตัวแปรที่ควรทราบ

f เป็นค่าแอดเดรสของรีจีสเตอร์ไฟล์ ปกติมีค่าตั้งแต่ 0-127 ในการเขียนโปรแกรมต้องทำการเลือกแบงค์ให้ถูกต้องเสียก่อน โดยรีจีสเตอร์ไฟล์ที่ใช้งานทั่วไป มักจะอยู่ในแบงค์ 0 ของหน่วยความจำข้อมูล

d ใช้ระบุปลายทางเก็บผลลัพธ์การกระทำคำสั่งนั้นๆ มีค่าได้ 2 ค่า คือ "0" และ"1"
d = "0" หลังจากกระทำคำสั่งแล้ว ผลลัพธ์ให้เก็บไว้ที่รีจีสเตอร์ W
d = "1" หลังจากกระทำคำสั่งแล้ว ผลลัพธ์ให้เก็บไว้ที่รีจีสเตอร์ไฟล์ที่คำสั่งนั้นๆ

k หมายถึง ค่าคงที่ใดๆ ที่ใช้กับคำสั่งประมวลผลของทางคณิตศาสตร์และลอจิก และยังหมายถึงตำแหน่งแอดเดรสที่ต้องการกระโดดไป ในกรณีที่ใช้กับคำสั่งการกระโดด เช่น Call หรือ Goto

b หมายถึง ข้อมูลระดับบิตหรือบิตภายในรีจีสเตอร์ไฟล์ขนาด 8 บิต

W หมายถึง รีจีสเตอร์ W (แอกคิวมูเลเตอร์)

X หมายถึง ตำแหน่งที่ไม่สนใจ เมื่อใช้กับแอสเซมเบลอร์ MPASM ของไมโครชิปจะกำหนดค่า x นี้เท่ากับ "0" โดยอัตโนมัติ เว้นแต่จะมีการกำหนดเพจหรือหน้าของหน่วยความจำเป็นอย่างอื่น

Label หมายถึง ตำแหน่งของแอดเดรสที่ต้องการกระโดดไปทำงาน

<b> หมายถึง ค่าในระดับบิต

TOS หมายถึง ตำแหน่งบนสุดของสแต็ก (Top Of Stack)

PC หมายถึง โปรแกรมเคาน์เตอร์

[] หมายถึง ออปชั่น หรือส่วนที่เพิ่มเติม

() หมายถึง ค่าของตัวแปรหรือข้อมูล

สำหรับค่าคงที่ของข้อมูลเลขฐานสิบหกที่เกี่ยวข้องในการเขียนโปรแกรมและอธิบายการทำงานของชุดคำสั่งของไมโครคอนโทรลเลอร์ตระกูล PIC จะแตกต่างจากไมโครคอนโทรลเลอร์ตระกูล MCS-51 โดยจะเขียนเป็น 0x00 ตัวเลข 2 หลักท้ายจะเป็นค่าของเลขฐานสิบหก ส่วน x โดยปกติจะมีค่าเป็น "0" การเขียนในลักษณะนี้ได้รับการกำหนดมาจากโปรแกรมแอสเซมเบลอร์ MPASM ของบริษัทไมโครชิป จะเห็นได้ว่าไม่ต้องมีอักษร H ต่อท้าย แต่ถ้าหากค่าเป็นเลขฐานสิบใส่อักษร d ต่อท้ายด้วยเสมอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 การเข้าถึงรีจีสเตอร์และข้อมูลของ PIC16F8XX

2.5.1 การเข้าถึงข้อมูลแบบทันทีทันใด (Immediate Addressing Mode)

การเข้าถึงข้อมูลแบบทันทีทันใดนี้จะเป็นการเข้าถึงค่าคงที่โดยตรง ด้วยการใช้ชุดคำสั่งที่เกี่ยวข้องกับค่าคงที่ (Literal) ร่วมกับรีจีสเตอร์ W หรือแอกคิวมูเลเตอร์ ตัวอย่างของคำสั่งที่แสดงให้เห็นถึงการเข้าถึงข้อมูลทันทีทันใดได้แก่ movlw k โดยที่ k คือค่าคงที่

2.5.2 การเข้าถึงข้อมูลแบบโดยตรง (Direct Addressing Mode)

การเข้าถึงข้อมูลแบบโดยตรงนี้เป็นการเข้าถึงข้อมูลหรือรีจีสเตอร์ด้วยการกำหนดแอดเดรสที่ต้องการเข้าถึงอย่างเจาะจง หรือระบุผ่านชื่อของรีจีสเตอร์ก็ได้ ดังตัวอย่าง

clrf Temp เป็นคำสั่งที่ต้องการเคลียร์ค่าของรีจีสเตอร์ตั้งชื่อว่า Temp

2.5.3 การเข้าถึงข้อมูลแบบโดยอ้อม (Indirect Addressing Mode)

การเข้าถึงแบบนี้จะใช้ในกรณีที่ต้องมีการติดต่อกับรีจีสเตอร์หลายๆ ตัว ในคราวเดียวกัน ไมโครคอนโทรลเลอร์ PIC16F8XX จะมีรีจีสเตอร์ FSR (File Select Register) และรีจีสเตอร์ INDF ช่วยในการติดต่อแบบโดยอ้อม รีจีสเตอร์ FSR ทำหน้าที่เป็นตัวชี้แอดเดรสของหน่วยความจำที่ต้องการเข้าถึง แล้วทำการอ่านข้อมูลนั้นๆ จากรีจีสเตอร์ INDF หรืออาจกล่าวได้ว่าใช้รีจีสเตอร์ FSR เก็บค่าแอดเดรส และรีจีสเตอร์ INDF เก็บค่าของข้อมูล

2.5.4 การเข้าถึงค่าข้อมูลแบบสัมพัทธ์

การเข้าถึงค่าข้อมูลแบบสัมพัทธ์นี้จะประกอบด้วยการคำนวณค่าสัมพัทธ์ของระยะห่างระหว่างแอดเดรสที่เริ่มต้นทำงานกับแอดเดรสที่ต้องการเข้าถึง โดยมีการใช้ค่าของโปรแกรมเคาน์เตอร์เข้ามาช่วยคำสั่งหนึ่งที่นำมาใช้ในการเข้าถึงข้อมูลสัมพัทธ์ คือ Retlw

กระบวนการจะเริ่มต้นด้วยการกำหนดค่าออฟเซต (Offset) ลงในรีจีสเตอร์ W แล้วนำไปรวมกับค่าของโปรแกรมเคาน์เตอร์ก็จะได้ค่าของแอดเดรสของหน่วยความจำที่ต้องการเข้าถึง การเข้าถึงแบบนี้มีข้อจำกัดคือ ระยะห่างของแอดเดรสเริ่มต้นกับแอดเดรสที่ต้องการเข้าถึงต้องไม่เกิน 256 ตำแหน่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 การใช้งาน Input / Output Port

ไมโครคอนโทรลเลอร์ PIC16F8XX มีพอร์ตสำหรับติดต่อกับอุปกรณ์ภายนอก 3 พอร์ต คือ พอร์ต A มีขนาด 6 บิต และพอร์ต B กับ พอร์ต C มีขนาด 8 บิต ขาแต่ละของ PIC16F8XX สามารถจ่ายกระแสออกได้สูงสุด 25 มิลลิแอมป์ และสามารถรับกระแสสูงสุดต่อขาได้ 20 มิลลิแอมป์ เมื่อใช้ไฟเลี้ยง +5 โวลต์ แต่ถ้าจะนำไปขับ LED จะต้องจำกัดกระแสโดยการต่อค่าความต้านทานเข้าไป แต่ถ้าใช้ไฟเลี้ยง +3 โวลต์ ก็สามารถที่จะขับ LED ได้โดยตรง

2.6.1 พอร์ต A และรีจีสเตอร์ TRISA

พอร์ต A เป็นพอร์ตแบบสองทิศทาง มีขนาด 6 บิต มีรีจีสเตอร์กำหนดทิศทางการถ่ายทอดข้อมูลของพอร์ต A คือ TRISA เมื่อต้องการใช้งานเป็นอินพุตต้องเขียนข้อมูล “1” ไปยังบิตที่ต้องการกำหนดให้เป็นอินพุต และถ้าหากต้องการใช้งานเป็นเอาต์พุตก็จะต้องเขียนข้อมูล “0” ไปยังบิตที่ต้องการให้เป็นเอาต์พุต

สำหรับขาของพอร์ต A บิตที่ 4 หรือ RA / TOCKI มีความแตกต่างจากขาอื่นๆ ตรงที่ว่าขานี้จะมีวงจรชมิตทริกเกอร์ (Schmitt Trigger) ต่อเข้าที่ขาอินพุต เพราะขานี้ใช้เป็นขาอินพุตรับสัญญาณนาฬิกาภายนอกสำหรับขาไทม์เมอร์เคาน์เตอร์ TMRO ภายใน

ขาของพอร์ต A ยังมัลติเพล็กซ์กับแอนะลอกอินพุต (Analog Input) และแอนะลอก V_{REF} อินพุต เมื่อต้องการใช้งานก็จะทำการกำหนดโดยเลือกเคลียร์ หรือเซต (Clearing / Setting) ที่บิตคอนโทรล (Control Bit) ในรีจีสเตอร์ ADCON1 (A/D Control Register 1)

ขณะเพาเวอร์ออนรีเซตขาของพอร์ต A จะถูกกำหนดให้เป็นแอนะลอกอินพุต และกำหนดลอจิกเป็น “0”

2.6.2 พอร์ต B และรีจีสเตอร์ TRISB

พอร์ต B เป็นพอร์ตแบบสองทิศทาง มีขนาด 8 บิต มีรีจีสเตอร์กำหนดทิศทางการถ่ายทอดข้อมูลของพอร์ต B คือ TRISB การกำหนดทิศทางจะกำหนดเช่นเดียวกับ TRISA

แต่ละขาของพอร์ต B สามารถเลือกให้พูลอัปโดยทำการเคลียร์บิต RBPU ในรีจีสเตอร์ OPTION และการพูลอัปจะถูกยกเลิกโดยอัตโนมัติเมื่อกำหนดให้พอร์ต B เป็นเอาต์พุต และนอกจากนี้การพูลอัปจะถูกยกเลิกเมื่อเกิดเพาเวอร์ออนรีเซต

ในส่วนของขาอีก 4 ขา คือ (RB4-RB7) ยังใช้เป็นขาเพื่อทำการตรวจสอบการเปลี่ยนแปลงข้อมูลหรือระดับสัญญาณของพอร์ต B เพื่อกระตุ้นให้เกิดการอินเตอร์รัพต์ กล่าวคือ จะสามารถรับสัญญาณอินเตอร์รัพต์ได้เฉพาะขาที่เป็นอินพุต (ถ้าบิต RB4-RB7 บิตใดบิตหนึ่งเป็นเอาต์พุตจะไม่สามารถรับสัญญาณอินเตอร์รัพต์ได้)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขณะที่อยู่ในโหมดสลีปสามารถทำให้อุปกรณ์เกิดการอินตอร์รัพต์ได้โดยผู้ใช้ สู่การบริการอินเตอร์รัพต์ย่อยสามารถทำการเคลียร์อินเตอร์รัพต์ได้ดังนี้

1) อ่านหรือเขียนจากพอร์ต B อย่างใดอย่างหนึ่งจะทำให้หยุดการทำงาน

2) เคลียร์บิต RBIF

2.6.3 พอร์ต C และรีจีสเตอร์ TRISC

พอร์ต C เป็นพอร์ตแบบสองทิศทาง มีขนาด 8 บิต มีรีจีสเตอร์กำหนดทิศทางการถ่ายทอดข้อมูลของพอร์ต C คือ TRISC

พอร์ต C จะมัลติเพล็กซ์กับฟังก์ชันอุปกรณ์เสริมมากมาย และพอร์ต C นี้ยังมีบัฟเฟอร์ (Buffer) ทางอินพุตแบบชมิตทริกเกอร์ เมื่อทำการอีนาเบิล (Enable) ฟังก์ชันอุปกรณ์เสริม ควรระวังในการกำหนดค่าของบิต TRISC และในบางอุปกรณ์เสริมจะทำการโอเวอร์ไรด์ (Override) บิต TRIS เพื่อกำหนดให้ขานี้เป็นอินพุตหรือเอาต์พุต และควรจะหลีกเลี่ยงตำแหน่งปลายทาง ผู้ใช้ควรจะอ้างอิงถึงส่วนอุปกรณ์เสริมที่คล้ายคลึงสำหรับการตั้งค่าบิต TRIS ให้ถูกต้อง

การเขียนโปรแกรมเพื่อกำหนดทิศทางของพอร์ต

เริ่มจากการเลือกแบงค์ของรีจีสเตอร์ไฟล์ไปที่แบงค์ 1 โดยการเซตบิต RP0 ในรีจีสเตอร์ STATUS จากนั้นเขียนค่า "0" หรือ "1" ซึ่งเป็นการกำหนดทิศทางว่าเป็นอินพุตหรือเอาต์พุต ไปยังรีจีสเตอร์ พอร์ต A พอร์ต B และ พอร์ต C ตามตำแหน่งบิตที่ต้องการกำหนดโดยแอดเดรสที่อ้างถึงจะตรงกับ 0x05, 0x06 และ 0x07 ตามลำดับ ซึ่งก็คือแอดเดรสของรีจีสเตอร์ TRISA, TRISB และ TRISC เนื่องจากไม่สามารถเข้าถึงรีจีสเตอร์ TRISA, TRISB และ TRISC ได้โดยตรง

หลังจากการกำหนดทิศทางของพอร์ตแล้ว ก็ต้องทำการเลือกแบงค์ของรีจีสเตอร์ไฟล์ให้กลับมายังแบงค์ 0 เพื่อเขียนหรืออ่านข้อมูลกับพอร์ตจริงๆ โดยผ่านทางรีจีสเตอร์ พอร์ต A พอร์ต B และ พอร์ต C วิธีการก็คือ การเคลียร์บิต RP0 ของรีจีสเตอร์ STATUS

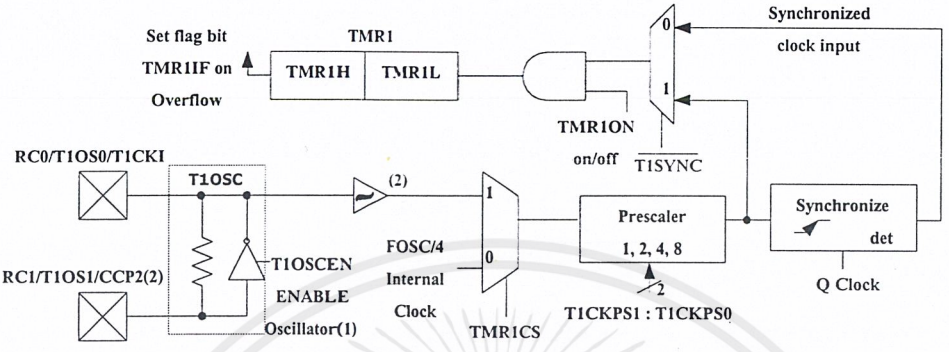
2.7 ไทม์เมอร์เคาน์เตอร์ภายใน PIC16F8XX

2.7.1 การเลือกโหมดไทม์เมอร์หรือเคาน์เตอร์

การกำหนดให้โมดูลไทม์เมอร์ 0 ทำงานในลักษณะเป็นไทม์เมอร์หรือตัวตั้งเวลากับเคาน์เตอร์ หรือวงจรนับ จะต้องทำการกำหนดบิต TOCS อันเป็นบิตที่ 5 ของรีจีสเตอร์ OPTION เมื่อกำหนดให้บิต TOCS เป็น "0 " โมดูลไทม์เมอร์ 0 จะทำงานในโหมดไทม์เมอร์ ซึ่งรับสัญญาณนาฬิกาจากภายในของ PIC16F8XX และหากกำหนดให้บิต TOCS เป็น "1" โมดูลไทม์เมอร์ 0 จะทำงานในโหมดเคาน์เตอร์ ซึ่งจะรับสัญญาณนาฬิกาจากภายนอกโดยผ่านมาทางขา RA4/TOCK1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อโมดูลไทม์เมอร์ 0 ทำงานในโหมดเคาน์เตอร์ TMRO จะเพิ่มค่าขึ้น เมื่อมีสัญญาณนาฬิกานั้นขึ้นอยู่กับการกำหนดที่บิต TOSE อันเป็นบิต 4 ของ รีจีสเตอร์ OPTION



รูปที่ 2.7 แผนผังการทำงานของไทม์เมอร์เคาน์เตอร์ภายใน PIC16F8XX

2.7.2 การอินเตอร์รัพต์ของ TMRO

จะเกิดขึ้นเมื่อค่าของรีจีสเตอร์ TMRO เกิดการเปลี่ยนแปลงจาก 0xff มาเป็น 0x00 หรือเรียกว่า เกิดโอเวอร์โฟลว เมื่อเกิดโอเวอร์โฟลว บิต TOIF ซึ่งเป็นบิต 2 ของรีจีสเตอร์ INTCON จะเซตเป็น "1" และถ้าหากไม่ต้องการให้เกิดอินเตอร์รัพต์อันเนื่องมาจากการเกิดโอเวอร์โฟลวของ TMRO ให้ทำการเคลียร์บิต TOIE อันเป็นบิต 5 ของรีจีสเตอร์ INTCON

ถ้าหากเกิดโอเวอร์โฟลวแล้วต้องทำการเคลียร์บิต TOIF ด้วยวิธีการทางซอฟต์แวร์เท่านั้นก่อนที่จะออกจากการบริการอินเตอร์รัพต์ มิฉะนั้นจะเกิดการอินเตอร์รัพต์ซ้อนขึ้นเป็นระยะ ซึ่งทำได้โดยการเขียนโปรแกรมเพื่อเคลียร์บิต TOIF รวมเข้าไปในโปรแกรมย่อยบริการอินเตอร์รัพต์อันเนื่องมาจาก TMRO เกิดโอเวอร์โฟลว

2.7.3 การใช้ TMRO กับสัญญาณนาฬิกาภายนอก

เมื่อต้องการให้ TMRO ทำงานด้วยสัญญาณนาฬิกาจากภายนอก สิ่งที่จะต้องให้ความสำคัญมาก จังหวะเวลาของสัญญาณจะต้องสอดคล้องกับจังหวะการทำงานภายในของ PIC16F8XX หรือเกิดการซิงโครไนเซชั่น (Syncronization) เมื่อมีการป้อนสัญญาณนาฬิกาจากภายนอกให้แก่ TMRO ผ่านทางขา RA4/TOCKI และไม่มีการใช้ปรีสเกลเลอร์ภายใน PIC16F8XX การซิงโครไนเซชั่นระหว่างสัญญาณนาฬิกาจากภายนอกที่ขา RA4 / TOCKI กับสัญญาณนาฬิกาภายในจะเกิดขึ้นจากการสุ่มที่ไซเคิล Q2 และ Q4 ของสัญญาณนาฬิกาภายใน สัญญาณที่ขา TOCKI จะต้องมีคาบเวลามากกว่าสัญญาณนาฬิกาภายในหรือ TOSC อย่างน้อย 2 เท่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีที่ใช้ปรีสเกลเลอร์สัญญาณนาฬิกาภายนอกจะถูกหารด้วยปรีสเกลเลอร์ แบบวงจรนับอะซิงโครนัส (Asynchronous Ripple Counter Type Prescaler) ทำให้เอาต์พุต ปรีสเกลเลอร์จะมีความสมมาตรกัน สัญญาณนาฬิกาภายนอกที่จะป้อนเข้าสู่ TMRO ต้องมีคาบเวลาอย่างน้อย 4 เท่าของ TOSC และความกว้างของสัญญาณนาฬิกาจะต้องไม่น้อยกว่า 10 นาโนวินาที

2.7.4 ผลทางเอาต์พุตของไทม์เมอร์เคาน์เตอร์

ไทม์เมอร์เคาน์เตอร์จะให้ผลทางเอาต์พุต 2 ลักษณะ คือ อ่านค่าของโมดูลไทม์เมอร์ที่รีจีสเตอร์ TMRO ที่แอดเดรส 0x01 และกำเนิดสัญญาณอินเตอร์รัพต์เมื่อเกิดโอเวอร์โฟลว จากการเปลี่ยนแปลงข้อมูล 0xff เป็น 0x00

2.7.5 ปรีสเกลเลอร์

เป็นตัวนับขนาด 8 บิตที่ทำหน้าที่ 2 อย่าง คือ เป็นปรีสเกลเลอร์เมื่อทำงานร่วมกับไทม์เมอร์เคาน์เตอร์ และเป็นโพสต์-สเกลเลอร์ (Post Scaler) เมื่อทำงานร่วมกับวอตช์ด็อกไทม์เมอร์

ปรีสเกลเลอร์จะหารสัญญาณนาฬิกาอินพุตด้วยค่า 1 ใน 8 ค่าซึ่งผู้ใช้งานสามารถกำหนดได้ โดยค่าของปรีสเกลเลอร์ที่สามารถกำหนดได้ประกอบด้วย 2, 4, 8, 16, 32, 64, 128 และ 256

การกำหนดค่าของปรีสเกลเลอร์ เพื่อทำการลดทอนสัญญาณนาฬิกาสามารถทำได้โดยกำหนดที่รีจีสเตอร์ OPTION 4 บิต คือ บิต PSA PS2 และ PS0

ค่าของปรีสเกลเลอร์จะถูกเคลียร์เมื่อ PIC16F8XX กระทำคำสั่งที่เกี่ยวข้องกับการเขียนข้อมูลมายัง TMRO เช่นคำสั่ง Clrf คำสั่ง Movwf และคำสั่ง Bsf เป็นต้น

2.8 การใช้งาน EEPROM ภายใน

PIC16F8XX มีหน่วยความจำแบบ EEPROM โดยสามารถอ่านและเขียนในขณะทำงานปกติได้แต่ต้องโดยการเข้าถึงนั้นจะต้องทำผ่านรีจีสเตอร์ฟังก์ชั่นพิเศษซึ่งต้องใช้ถึง 4 ตัวดังนี้

2.8.1 EECON1 และ EECON2

EECON1 เป็นรีจีสเตอร์ที่ใช้ควบคุมการเข้าถึงหน่วยความจำ และ EECON2 ใช้เป็น ซีเควน (Sequence) ของการเขียนเท่านั้นไม่สามารถอ่านและเขียนในขณะทำงานปกติ

บิต EEPGO เป็นบิตที่ใช้กำหนดว่าจะเข้าถึง EEPROM หรือหน่วยความจำแบบแฟลช (Flash Memory) คือ ถ้าเป็น "0" จะเป็นการเข้าถึงอีอีพรอม (EEPROM) ถ้าเป็น "1" เป็นการเข้าถึงหน่วยความจำแบบแฟลช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต RD และ WR เป็นบิตที่ใช้กำหนดว่าจะเป็นการอ่านหรือเขียน โดยทั้งสองบิตนี้สามารถเซตได้โดยซอฟต์แวร์ อย่างเดียวแล้วจะถูกเคลียร์โดยฮาร์ดแวร์ (Hardware) เมื่ออ่านหรือเขียนเสร็จเรียบร้อย

บิต WREN เป็นบิตที่ใช้อีนาเบิล การเขียนและปกติจะถูกดีสเอเบิล (Disable) หรือเป็น "0"

บิต WRERRใช้บอกว่าการเขียนเกิดขัดจังหวะจากการรีเซต (Reset) หรือวอตซ์ด็อก ทำงาน ซึ่งบิตนี้จะเซต "1" ถ้าเกิดกรณีทั้งสองขณะเขียนข้อมูลลงอีอีพรอมและถ้าไม่เกิดกรณีทั้งสองขณะเขียนข้อมูลจะเป็น "1"

ตารางที่ 2.6 รายละเอียดของรีจีสเตอร์ EECON1

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0	R :อ่านค่าได้ W : เขียนค่าได้ U : ไม่ใช้งาน "0" -n ค่าที่เกิดหลังเพาเวอร์ออนรีเซต
U	U	U	R/W-0	R/W-X	R/W-0	R/S-0	R/S-X	
-	-	-	EEIF	WRERR	WREN	WR	RD	

บิต 7-5 : ใน PIC16F8XX ไม่ใช้บิตนี้ อ่านค่าได้เป็น "0"

บิต 4 : EEIF (EEPROM Write Operation Interrupt Flag Bit) บิตแสดงการเขียนข้อมูลลงในหน่วยความจำอีอีพรอมสมบูรณ์ จะทำให้เกิดการอินเตอร์รัพต์ขึ้น หากมีการอีนาเบิลไว้

"1" เมื่อการเขียนข้อมูลลงในหน่วยความจำอีอีพรอมสมบูรณ์ (เคลียร์ด้วยกระบวนการทางซอฟต์แวร์)

"0" เมื่อการเขียนข้อมูลลงในหน่วยความจำอีอีพรอมไม่สมบูรณ์หรือยังไม่เริ่มต้นการเขียนข้อมูล

บิต 3 : WRERR (EEPROM Error Flag Bit) บิตแสดงความผิดพลาดในการเขียนข้อมูลลงในหน่วยความจำอีอีพรอม

"1" เมื่อการเขียนข้อมูลถูกขัดจังหวะ ทำให้การเขียนข้อมูลเกิดความผิดพลาด

"0" เมื่อการเขียนข้อมูลสมบูรณ์

บิต 2 : WREN (EEPROM Write Enable Bit) บิตอีนาเบิลการเขียนข้อมูลยอมให้เกิดการเขียนข้อมูล

"1" ยอมให้เกิดการเขียนข้อมูล

"0"ไม่ยอมให้เกิดการเขียนข้อมูลยอมให้เกิดการเขียนข้อมูล

บิต 1 : WR (Write Control Bit) บิตควบคุมการเขียนข้อมูลลงในหน่วยความจำอีอีพรอม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

"1" แสดงถึงไซเคิลของการอินนิเชียล (Initial) การเขียนข้อมูล หรือใช้เป็นบิตเริ่มต้นการเขียนข้อมูลสามารถเซตได้ทางซอฟต์แวร์และบิตนี้จะเคลียร์อัตโนมัติ เมื่อเกิดการเขียนข้อมูลเสร็จสมบูรณ์

"0" เมื่อไซเคิลของการเขียนข้อมูลเกิดขึ้นอย่างสมบูรณ์

บิต 0 : RD (Read Control Bit) บิตควบคุมการอ่านข้อมูลจากหน่วยความจำอีอีพรอม

"1" แสดงถึงไซเคิลของการกำหนดค่าเริ่มต้นการอ่านข้อมูล หรือใช้เป็นบิตเริ่มต้นการอ่านข้อมูลสามารถเซตได้ได้ทางซอฟต์แวร์ บิตนี้จะเคลียร์อัตโนมัติเมื่อเกิดการเขียนข้อมูลเสร็จสมบูรณ์

"0" เมื่อไซเคิลของการเขียนข้อมูลเกิดขึ้นอย่างสมบูรณ์

2.8.2 EEDATA

EEDATA ใช้เป็นที่พักข้อมูลขนาด 8 บิตที่ต้องการอ่านหรือเขียนมีจำนวน 64 ไบต์ มีแอดเดรสอยู่ที่ 0x00 – 0x3F การเขียนและอ่านข้อมูลจะกระทำในระดับไบต์หรือครั้งละ 8 บิตเท่านั้น การเขียนข้อมูลลงในหน่วยความจำอีอีพรอมทุกครั้งต้องทำการลบข้อมูลออกก่อนเสมอ ซึ่งอัตราเร็วในการลบข้อมูลจะสูง ในขณะที่อัตราเร็วในการเขียนข้อมูลจะขึ้นอยู่กับไทม์เมอร์ในตัว PIC16F8XX ซึ่งจะเปลี่ยนแปลงตามแรงดันและอุณหภูมิในขณะที่ทำการเขียนข้อมูลนั้น

เมื่อทำการป้องกันข้อมูลในหน่วยความจำอีอีพรอมแล้ว ซีพียูสามารถอ่านและเขียนข้อมูลในหน่วยความจำได้เป็นปกติ แต่เครื่องโปรแกรมภายนอกจะไม่สามารถเข้าถึงหน่วยความจำส่วนนี้

2.8.3 EEADR

เป็นรีจีสเตอร์ที่ใช้เก็บค่าแอดเดรสของหน่วยความจำข้อมูลอีอีพรอมที่ต้องการเขียนโดยมีค่าตั้งแต่ 00H ถึง FFH (256 ไบต์)

2.9 คุณสมบัติพิเศษของซีพียู PIC16F8XX

2.9.1 การทำงานในโหมดสลีปหรือเพาเวอร์ดาวน์

การเข้าสู่โหมดการทำงานแบบนี้ จะเกิดขึ้นหลังจากกระทำคำสั่ง Sleep เมื่อเริ่มทำงานวอตช์ด็อกไทม์เมอร์จะถูกเคลียร์ (แต่ยังทำงานอยู่) บิต PD ในรีจีสเตอร์ STATUS จะถูกเคลียร์ด้วย ในขณะที่บิต TO ในรีจีสเตอร์ STATUS จะเซต วงจรออสซิลเลเตอร์ภายในหยุดทำงาน ส่วนขาสัญญาณทุกพอร์ตจะคงสถานะเดิมก่อนหน้าที่จะกระทำคำสั่ง Sleep

การออกจากโหมดสลีปหรือการเวกอัป (Wake Up)

PIC16F8XX จะออกจากโหมดสลีปได้ หากมีเหตุการณ์ 1 ใน 3 เกิดขึ้น คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) เกิดสัญญาณรีเซตจากภายนอกป้อนเข้าที่ขา MCLR

2) วอตซ์ด็อกไทม์เมอร์เวกอัป (ถ้าหากมีการอีนาเบิลวอตซ์ด็อกไทม์เมอร์ไว้)

3) เกิดอินเตอร์รัพต์ขึ้นจากการป้อนสัญญาณเข้าที่ขา RB0/INT จากการเปลี่ยนแปลงข้อมูลที่พอร์ต B บิต 7- บิต 4 หรือจากการที่เขียนข้อมูลในอีอีพรอมสมบูรณ์

เมื่อไมโครคอนโทรลเลอร์ กระทำคำสั่ง Sleep แล้วคำสั่งต่อไปนี้จะถูกเฟตซ์ขึ้นมา ถ้าหากคำสั่งต่อไปนี้เป็นคำสั่งที่ทำให้เกิดการอินเตอร์รัพต์ ไมโครคอนโทรลเลอร์จะเวกอัปกระทำคำสั่งและต้องตรวจสอบสถานะที่บิต GIE ด้วยหากบิต GIE เป็น “0” อยู่ แล้วไมโครคอนโทรลเลอร์เกิดการเวกอัปซีพียูจะทำงานในคำสั่งที่ถัดจาก Sleep ต่อไปตามลำดับ แต่ถ้ากำหนดบิต GIE เป็น “1” เมื่อเกิดการเวกอัปไมโครคอนโทรลเลอร์จะทำคำสั่งที่ถัดจาก Sleep แล้วกระโดดไปยังอินเตอร์รัพต์แอดเดรสที่ตำแหน่ง 0x04 ดังนั้นคำสั่งที่ควรใช้หลังคำสั่ง Sleep คือ Nop (No Operation)

ถ้าเกิดการอินเตอร์รัพต์เกิดขึ้นระหว่างหรือหลังจากเอ็กซิคิวต์คำสั่ง Sleep ซีพียูจะเวกอัปโหมดสลีปทันที ทั้งนี้ซีพียูจะกระทำคำสั่ง Sleep จนเสร็จสิ้นก่อน ในขณะที่วอตซ์ด็อกไทม์เมอร์จะเคลียร์ค่าทั้งหมด บิต T0 จะเซต และ PD จะได้รับการเคลียร์

ดังนั้นก่อนเข้าสู่โหมดสลีปจะมีการตรวจสอบสถานะของบิต INTE และ INTF ก่อนและเพื่อให้แน่ใจว่าวอตซ์ด็อกไทม์เมอร์ได้รับการเคลียร์จริงๆ ควรเอ็กซิคิวต์คำสั่ง Clrwdt ก่อนที่จะกระทำการเอ็กซิคิวต์คำสั่ง Sleep

2.9.2 การโปรแกรมแบบอนุกรมภายในวงจร

PIC16F8XX สามารถที่จะโปรแกรมข้อมูลลงบนตัวมันได้ แม้ว่าจะอยู่ในวงจรก็ตาม โดยใช้ขาสัญญาณสองขา สำหรับสัญญาณนาฬิกาและข้อมูลอีก 3 ขา สัญญาณสำหรับไฟเลี้ยง กราวด์ และแรงดันสำหรับโปรแกรม

PIC16F8XX ถูกกำหนดให้ทำงานในโหมดโปรแกรม (Program / Verify Mode) โดยการทำให้ขา RB6 และ RB7 ได้รับลอจิก ”0” ขา RB6 จะถูกใช้สำหรับรับสัญญาณนาฬิกา ส่วนขา RB7 รับข้อมูลที่ต้องการโปรแกรม ในขณะที่ขา MCLR ใช้รับแรงดันสำหรับการโปรแกรม

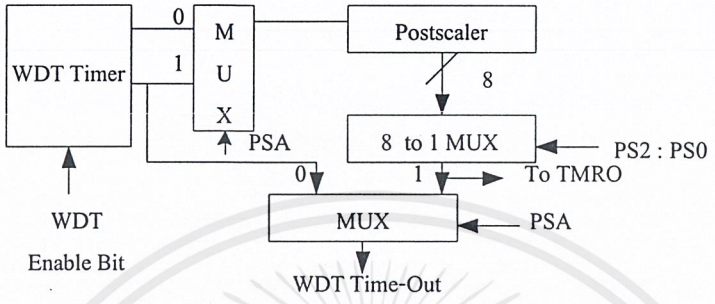
หลังจากการรีเซตโปรแกรมเคาน์เตอร์ของ PIC16F8XX จะถูกกำหนดให้ไปที่ตำแหน่ง 0x00 คำสั่งขนาด 6 บิต และข้อมูลขนาด 14 บิต จะถูกส่งเข้าไปในหน่วยความจำโปรแกรมภายใน PIC16F8XX โดยการใช้คำสั่งเกี่ยวกับการถ่ายทอดข้อมูล

2.9.3 วอตซ์ด็อกไทม์เมอร์

เป็นวงจรกำเนิดสัญญาณอย่างต่อเนื่องหรือที่เรียกว่า ฟรีรันนิ่งออสซิลเลเตอร์ (Free Running Oscillator) แบบ RC บรรจุอยู่ภายในไมโครคอนโทรลเลอร์ PIC16F8XX ในการทำงานจึงไม่ต้องอาศัยอุปกรณ์ภายนอกแต่อย่างใด วงจรออสซิลเลเตอร์ในส่วนของวอตซ์ไทม์เมอร์นี้แยกการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำงานออกจากวงจรออสซิลเลเตอร์ที่ขา OSC1 / CLKIN อย่างสิ้นเชิง ดังนั้นวอตช์ด็อกไทม์เมอร์ หรือ WDT จะยังคงทำงานต่อเนื่องไป แม้ว่าไม่มีการป้อนสัญญาณนาฬิกาเข้ามาที่ขา OSC1/CLKIN และ OSC2/CLKOUT ก็ตาม โดยยังคงทำงานตราบเท่าที่ยังมีการจ่ายไฟเลี้ยงให้ไมโคร



คอนโทรลเลอร์

รูปที่ 2.8 แผนผังการทำงานของวอตช์ด็อกไทม์เมอร์

เมื่อไมโครคอนโทรลเลอร์ PIC16F8XX กระทำคำสั่ง Sleep สัญญาณนาฬิกาที่ป้อนให้ไมโครคอนโทรลเลอร์ PIC16F8XX จะถูกปลดออกแต่วอตช์ด็อกไทม์เมอร์ภาย PIC16F8XX ยังคงทำงานต่อไป กล่าวคือยังคงมีการกำเนิดสัญญาณและนับค่าต่อไปเรื่อยๆ จนถึงค่าที่กำหนดไว้ ก็จะเกิดสภาวะที่เรียกว่า ไทม์เอาต์ (Time Out) ถ้าหากอยู่ในสภาวะการทำงานปกติเมื่อวอตช์ด็อกเกิดไทม์เอาต์ จะสร้างสัญญาณรีเซตขึ้น แต่ถ้าอยู่ในโหมดสลีปสัญญาณที่เกิดขึ้นจากการไทม์เอาต์จะกระตุ้นให้ไมโครคอนโทรลเลอร์ PIC16F8XX กลับมาทำงานในโหมดปกติต่อไป

2.9.4 คาบเวลาของวอตช์ด็อกไทม์เมอร์

วอตช์ด็อกจะมีคาบเวลาไทม์เอาต์เท่ากับ 18 มิลลิวินาที ในกรณีที่ไม่มีการใช้ปรีสเกลเลอร์ คาบเวลานี้อาจคลาดเคลื่อนได้ตามอุณหภูมิใช้งานและค่าของแรงดันไฟเลี้ยง ถ้าต้องการให้คาบเวลาของวอตช์ด็อกยาวนานขึ้น จะต้องนำปรีสเกลเลอร์มาช่วยซึ่งสามารถกำหนดได้ด้วยกระบวนการซอฟต์แวร์โดยการกำหนดค่าลงในรีจีสเตอร์ OPTION เมื่อนำปรีสเกลเลอร์มาช่วยเพิ่มคาบเวลาจะสามารถเพิ่มได้สูงถึง 128 เท่า หรือประมาณ 2.3 วินาที

คำสั่ง Clrwdt และ Sleep เป็นคำสั่งที่ใช้ในการเคลียร์ค่าของวอตช์ด็อก เมื่อกระทำคำสั่งนี้ค่าของวอตช์ด็อกจะกลายเป็นศูนย์ โดยที่ไม่เกิดการไทม์เอาต์และกำเนิดสัญญาณเพื่อทำการรีเซตระบบ บิต T0 ในรีจีสเตอร์ STATUS จะถูกเคลียร์เมื่อวอตช์ด็อก เกิดการไทม์เอาต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.5 การรีเซตของ PIC16F8XX

กระบวนการรีเซตที่เกิดขึ้นกับไมโครคอนโทรลเลอร์ PIC16F8XX มีด้วยกัน 5 แบบ ดังนี้

1) เพาเวอร์ออนรีเซต (Power-On Reset : POR)

2) การเกิดรีเซตที่ขา MCLRในขณะที่ทำงานปกติ

3) เกิดการรีเซตที่ขา MCLR ในขณะที่อยู่ในโหมดสลีป

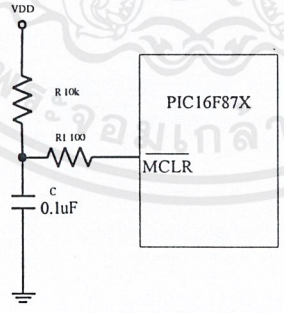
4) การรีเซตอันเนื่องมาจากวอตช์ด็อกไทม์เมอร์ในขณะทำงานปกติ

5) การรีเซตอันเนื่องมาจากวอตช์ด็อกไทม์เมอร์ในขณะทำงานอยู่ในโหมดสลีป

เมื่อเกิดการรีเซตจะส่งผลต่อค่าของรีจีสเตอร์ภายในตัวไมโครคอนโทรลเลอร์ PIC16F8XX โดยจะเกิดการเคลียร์ค่าเป็น 0 แต่ทั้งนี้ก็ต้องขึ้นอยู่กับว่าสาเหตุของการรีเซตมาจากอะไร เพราะจะมีรีจีสเตอร์บางตัวที่ยังคงค่าเดิมอยู่แม้ว่าจะเกิดการรีเซตก็ตาม

2.9.6 เพาเวอร์ออนรีเซต (POR)

การรีเซตแบบนี้จะเกิดขึ้นเมื่อเกิดการตรวจจับแรงดันที่ขา V_{DD} ได้ในระหว่างแรงดันอยู่ในช่วง 1.2-1.7 โวลต์ ในขณะที่ขา MCLR จะต่ออยู่กับไฟเลี้ยงโดยตรงหรือผ่านตัวต้านทานทำให้ขา R (Reset) ของ RS ฟลิปฟลอป (Flip Flop) ที่อยู่ภายในไมโครคอนโทรลเลอร์ได้รับลอจิก "1" ซึ่งจะส่งผลให้ฟลิปฟลอปเกิดการรีเซต การรีเซตแบบนี้จะไม่เกิดขึ้นหากไม่มีการจ่ายไฟเลี้ยงให้แก่ไมโครคอนโทรลเลอร์ PIC16F8XX



รูปที่ 2.9 การต่อวงจรรีเซตจากภายนอก

2.9.7 เพาเวอร์อัปไทม์เมอร์ (PWRT)

เมื่อเกิดเพาเวอร์ออนรีเซต จะทำให้ไทม์เมอร์ตัวหนึ่งภายใน PIC16F8XX ทำงาน ไทม์เมอร์ตัวนั้น คือ เพาเวอร์อัปไทม์เมอร์ (Power Up Timer : PERT) จะทำการกำหนดคาบเวลาไทม์เอาต์ (Time-Out : TPWRT) ไว้ที่ 72 มิลลิวินาที หลังจากที่เกิดเพาเวอร์ออนรีเซต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คาบเวลาของเพาเวอร์อัปไทม์เมอร์ได้รับการกำหนดโดยวงจรออสซิลเลเตอร์ RC ภายในตัวไมโครคอนโทรลเลอร์ โดยจะคำรงสถานะการรีเซตไว้ จนกว่าเพาเวอร์อัปไทม์เมอร์จะแอกทีฟ

การอีนาเบิลและดิสเอเบิลเพาเวอร์อัปไทม์เมอร์สามารถทำได้โดยกำหนดที่บิต PWRTE ใน Configuration Bit

2.9.8 ออสซิลเลเตอร์ สตาร์ทอัปไทม์เมอร์ (OST)

OST ทำหน้าที่กำหนดสัญญาณหน่วงเวลาขนาด 1,024 ไซเคิล หลังจากสิ้นสุดการทำงานของ PWRT เพื่อให้แน่ใจว่าวงจรจะกำเนิดสัญญาณนาฬิกา แบบคริสตอลและเซรามิกเรโซเนเตอร์พร้อมจะเริ่มทำงานแล้ว

2.10 การอินเตอร์รัพต์ของ PIC16F8XX

การอินเตอร์รัพต์ (Interrupt) คือ การขัดจังหวะการทำงานปกติของซีพียู เมื่อเกิดการอินเตอร์รัพต์ขึ้น ซีพียูจะทำงานจนจบไซเคิลนั้น แล้วจึงมาตอบสนองการอินเตอร์รัพต์โดยกระโดดมาทำงานยังตำแหน่งที่กำหนดไว้ จนกระทั่งเมื่อตอบสนองการอินเตอร์รัพต์เรียบร้อยแล้ว ซีพียูจะกลับไปทำงานต่อจากเหตุการณ์ที่เกิดขึ้นก่อนหน้าการอินเตอร์รัพต์

การเกิดอินเตอร์รัพต์ในระบบไมโครคอนโทรลเลอร์ PIC16F8XX มีทั้งจากภายในตัวซีพียูและจากภายนอกหรือจากกระบวนการประมวลผลของซอฟต์แวร์ก็ได้ ซึ่งในไมโครคอนโทรลเลอร์ PIC16F8XX สามารถรับการอินเตอร์รัพต์โดยสรุปได้ 4 แหล่งกำเนิดดังนี้

1) สัญญาณอินเตอร์รัพต์จากภายนอกป้อนเข้าที่ขา RB0/INT

2) การอินเตอร์รัพต์อันเนื่องจาก TMRO เกิดโอเวอร์โฟลว

3) การอินเตอร์รัพต์อันเนื่องมาจากเกิดการเปลี่ยนแปลงระดับสัญญาณที่ขา RB4-RB7

4) การอินเตอร์รัพต์อันเนื่องมาจากการเขียนข้อมูลในหน่วยความจำข้อมูลอีอีพรอมเสร็จ

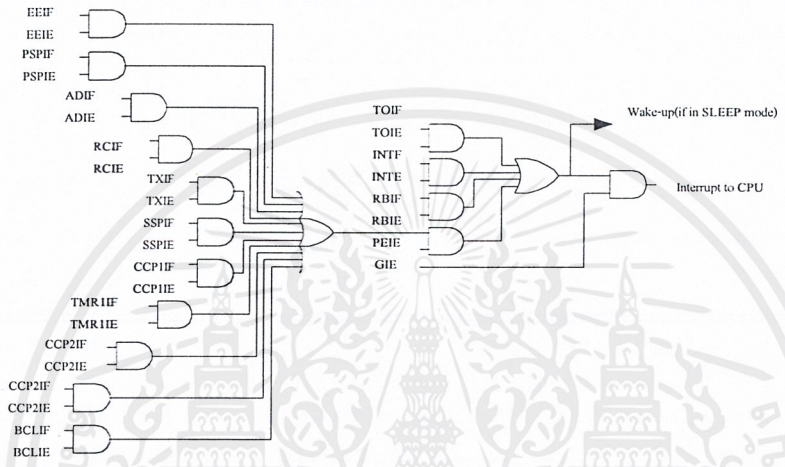
PIC16F8XX สามารถที่จะเลือกการตอบสนองการอินเตอร์รัพต์ทั้ง4รูปแบบ หรือจะเลือกตอบสนองบางแหล่งกำเนิดก็ได้โดยการกำหนดที่รีจีสเตอร์ INTCON

ระดับความสำคัญของการตอบสนองการอินเตอร์รัพต์ บิต GIE จะเป็นบิตที่มีนัยสำคัญสูงสุด ถ้าหากบิต GIE เป็น “0” จะเป็นการดิสเอเบิลการอินเตอร์รัพต์จากทุกแหล่งกำเนิด ในขณะเดียวกันถ้าหากบิต GIE เป็น “1” จะกำหนดให้ PIC16F8XX พร้อมที่จะตอบสนองการอินเตอร์รัพต์จากทุกแหล่งกำเนิด ขึ้นกับว่าบิตควบคุมการอินเตอร์รัพต์อื่นๆ ในรีจีสเตอร์ INTCON จะมีการอีนาเบิลไว้หรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเกิดการอินเตอร์รัพต์ขึ้น PIC16F8XX จะกระโดดไปทำงานที่แอดเดรส 0x04 ของหน่วยความจำโปรแกรมเสมอ ซึ่งโปรแกรมย่อยที่แอดเดรส 0x004 จะเรียกว่า โปรแกรมย่อยของการบริการอินเตอร์รัพต์ (Interrupt)

ทุกครั้งที่เกิดอินเตอร์รัพต์ ก่อนที่จะทำการเอ็กซิคิวต์ โปรแกรมย่อยบริการอินเตอร์รัพต์ควรจะมีการหน่วงเวลาอย่างน้อย 3-4 ไซเคิลของการกระทำคำสั่ง (Instruction Cycle)



รูปที่ 2.10 แผนผังลอจิกนัยสำคัญของตัวแปรที่ใช้ควบคุมการอินเตอร์รัพต์

2.10.1 รายละเอียดของแหล่งกำเนิดอินเตอร์รัพต์

1) แหล่งกำเนิดอินเตอร์รัพต์จากภายนอก (External Interrupt)

การใช้แหล่งกำเนิดอินเตอร์รัพต์แบบนี้จะต้องใช้ทั้งกระบวนการทางฮาร์ดแวร์และซอฟต์แวร์ โดยฮาร์ดแวร์ หมายถึง การกำหนดระดับสัญญาณหรือระดับลอจิกที่ป้อนเข้าสู่ขา RB0 / INT เพื่อทำให้ซีพียูภายใน PIC16F8XX รับทราบถึงการเกิดอินเตอร์รัพต์

การกำหนดขอบขาของสัญญาณที่ทำให้เกิดการอินเตอร์รัพต์ต้องกระทำด้วยกระบวนการทางซอฟต์แวร์ โดยการเขียนโปรแกรมเพื่อกำหนดข้อมูลลงในบิต INTED (บิต 6) ของรีจีสเตอร์ OPTION ผู้ใช้งานสามารถอีนาเบิลการอินเตอร์รัพต์แบบนี้ได้ โดยกำหนดที่บิต 4 ในรีจีสเตอร์ INTCON

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) อินเตอร์รัพต์จากไทม์เมอร์เคาน์เตอร์

เมื่อไทม์เมอร์เคาน์เตอร์ TMR0 เกิดการโอเวอโฟลว กล่าวคือ ถ้าการนับของเคาน์เตอร์เปลี่ยนจาก 0xff มาเป็น 0x00 จะเกิดการอินเตอร์รัพต์ขึ้น การอินเตอร์รัพต์นี้เป็นการอินเตอร์รัพต์จากแหล่งกำเนิดภายใน PIC16F8XX

3) อินเตอร์รัพต์จากการเปลี่ยนแปลงที่ขา RB4-RB7 ของพอร์ต B

การอินเตอร์รัพต์แบบนี้เกิดขึ้นได้ก็ต่อเมื่อมีการเซตบิต RBIE ซึ่งเป็นบิต 3 ของ รีจีสเตอร์ INTCON เมื่อมีการอินเตอร์รัพต์แบบนี้ ขาสัญญาณของพอร์ต B ทั้ง 4 ขา คือ RB4-RB7 จะเริ่มมีการตรวจสอบเปรียบเทียบระดับลอจิกที่ขาสัญญาณว่าเกิดการเปลี่ยนแปลงหรือไม่ ถ้าหากเกิดการเปลี่ยนแปลงขึ้นก็จะเกิดการอินเตอร์รัพต์ ซีพียูจะทำการเซตบิต RBIF อันเป็นบิต 0 ของรีจีสเตอร์ INTCON

การเคลียร์อินเตอร์รัพต์นี้ สามารถกระทำได้ด้วยกระบวนการทางซอฟต์แวร์ได้ 2 วิธี คือ

1) รีเซตบิต RBIE ซึ่งเป็นบิต 3 ของรีจีสเตอร์ INTCON

2) อ่านค่าพอร์ต B หลังเกิดการอินเตอร์รัพต์ และรีเซตบิต RBIF ของรีจีสเตอร์ INTCON

2.10.2 การเก็บข้อมูลเมื่อเกิดการอินเตอร์รัพต์

เมื่อเกิดการอินเตอร์รัพต์ค่าของโปรแกรมเคาน์เตอร์ (PC) จะถูกเก็บไว้ในสแต็กโดยอัตโนมัติ ทั้งนี้เพราะไมโครคอนโทรลเลอร์จะต้องทราบว่าก่อนเกิดการอินเตอร์รัพต์ทำงานอยู่ที่แอดเดรสใด เมื่อตอบสนองการอินเตอร์รัพต์แล้ว ซีพียูจะต้องย้อนกลับมาทำงานต่อตามที่เก็บไว้ในสแต็ก

2.10.3 ตำแหน่งเริ่มต้นของการบริการอินเตอร์รัพต์

ค่าอินเตอร์รัพต์เวกเตอร์ (Interrupt Vector) ของ PIC16F8XX คือ 0x004 เมื่อเกิดการอินเตอร์รัพต์ขึ้น อินเตอร์รัพต์เวกเตอร์จะชี้มาที่หน่วยความจำโปรแกรมตำแหน่ง 0x004 ซึ่งที่ตำแหน่งแอดเดรสนี้คำสั่งแรกของโปรแกรมย่อยของการบริการอินเตอร์รัพต์จะต้องถูกเก็บไว้

2.10.4 การบริการอินเตอร์รัพต์เมื่อมีแหล่งกำเนิดอินเตอร์รัพต์ภายนอกหลายแหล่ง

ถ้าหากในระบบไมโครคอนโทรลเลอร์มีแหล่งกำเนิดอินเตอร์รัพต์มากกว่า 1 แหล่ง ฮาร์ดแวร์จะกำเนิดสัญญาณ เพื่ออินเตอร์รัพต์ไมโครคอนโทรลเลอร์ และแจ้งให้ทราบถึงสาเหตุของการอินเตอร์รัพต์ เนื่องจากไมโครคอนโทรลเลอร์มีการจัดโปรแกรมย่อยที่แตกต่างกันไปสำหรับการอินเตอร์รัพต์ในแต่ละลักษณะ

สำหรับ PIC16F8XX มีขาสำหรับรับสัญญาณอินเตอร์รัพต์จากภายนอกเพียง 1 ขา เท่านั้น ถ้าหากต้องทำงานร่วมกับแหล่งกำเนิดอินเตอร์รัพต์จากภายนอกหลายแหล่ง จะต้องใช้ขาสัญญาณพอร์ตอินพุตรับสัญญาณจากแหล่งกำเนิดอินเตอร์รัพต์ภายนอก แล้วใช้วิธีการพูลลิ่ง (Pulling) ซึ่งคือ

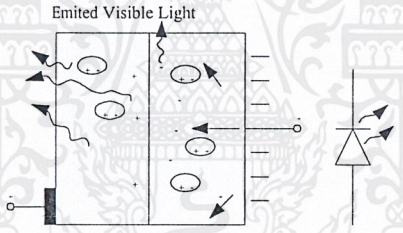
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้ซอฟต์แวร์ตรวจสอบการเปลี่ยนแปลงของขาสัญญาณอินพุตแต่ละขา ที่ต่อกับแหล่งกำเนิดอินเตอร์รัพต์ภายนอก โดยจะต้องมีการกำหนดนัยสำคัญของการตอบสนองการอินเตอร์รัพต์ด้วย

2.11 ไดโอดเปล่งแสง

ไดโอดเปล่งแสง ย่อมาจากคำว่า ไลท์ อิมิตติ้งไดโอด (Light Emitting Diode : LED) เป็นไดโอดที่สามารถเปล่งแสงสว่างออกมาด้วยคลื่นความถี่เดียวและมีเฟสต่อเนื่องได้ ซึ่งต่างจากแสงธรรมดาที่คนเรามองเห็น ประกอบไปด้วยคลื่นที่มีความถี่เฟสต่างๆมารวมกัน

โครงสร้างของไดโอดเปล่งแสงเหมือนกับไดโอดทั่วๆไป ประกอบขึ้นมาจากการเอาสารพี (P) และสารเอ็น (N) มาประกอบกันโดยผิวข้างหนึ่งเป็นมันคล้ายกระจก เมื่อเราให้ฟอร์เวิร์ด (Forward) แก่ไดโอดเปล่งแสงจะทำให้อิเล็กตรอนที่สารกึ่งตัวนำชนิดเอ็นมีพลังงานสูงขึ้น จนสามารถวิ่งข้ามรอยต่อไป รวมกับโฮล (Hold) ในสารพีได้ ก่อให้เกิดพลังงานที่เราเรียกว่าพลังงาน “โฟตอน” (Photon) เปล่งออกมา



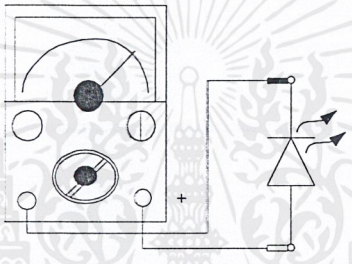
รูปที่ 2.11 โครงสร้างและสัญลักษณ์ของไดโอดเปล่งแสง

ปรากฏการณ์ใหม่ที่เกิดขึ้นกับตัวไดโอดชนิดนี้บังเกิดขึ้นไม่ได้ถ้าเราไม่ใช้กระบวนการซึ่งเรียกว่า อิเล็กโตรลูมิเนสเซ็นส์ (Electroluminescence) เราต้องสร้างให้โลหะหน้าสัมผัสชนิดพีมีขนาดเล็กกว่าชนิดเอ็น อันจะทำให้พลังงานโฟตอนมีจำนวนมากที่สุด และที่สำคัญการที่จะทำให้พลังงานนี้เรืองแสงขึ้นมาสว่างสุกใสนั้น จำเป็นจะต้องเติมสารเจือบางอย่างลงไปในซิลิกอน (Silicon) หรือเยอรมันเนียม (Germanium) ด้วย สารนั้นก็คือ แกเลียมอาร์เซไนด์ฟอสไฟด์ (Gallium Arsenide Phosphide : GaAsP) หรือแกเลียมฟอสไฟด์ (Gallium Phosphide : GaP) และนี่เองที่ทำให้อิเล็กทรอนิกส์ทางด้านแสงได้รับการพัฒนาขึ้นอย่างรวดเร็ว จนได้อุปกรณ์อิเล็กทรอนิกส์ทางแสงเพิ่มขึ้นมาอย่างเช่น ไฟเบอร์ออพติค (Fiber Optic) ไอซีประเภทออพติคอล (Optical) รวมไปถึงการเกิดวงจรที่มีการเชื่อมโยงพลังงานต่างๆด้วยกรรมวิธี “ออพโตไอโซแลต” (Opto Isolate)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยสำหรับการวัดความเข้มของแสงของไดโอดเปล่งแสง เราจะวัดกันเป็นแคนเดลลา (Candella) โดย 1 แคนเดลลามีความเข้มเท่ากับ 4 π ลูเมนส์

ไดโอดเปล่งแสงอาจมีสีสันต่างๆกันไปเช่น สีเขียว แดง เหลือง ส้ม และขาว และมีการคาดหวังกันว่าอีกไม่นานอาจจะมีไดโอดเปล่งแสงสีน้ำเงินออกมาใช้ โดยทั่วไปไดโอดเปล่งแสงที่มีอยู่ในปัจจุบันใช้กับระดับโวลต์ 1.7 ถึง 3.3 โวลต์ ซึ่งจะเป็นค่าแรงดันที่ทำให้ไดโอดเปล่งแสงทำงานได้สมบูรณ์ที่สุดเมื่อนำมาใช้กับอุปกรณ์โซลิดสเตท (Solid State)โดยจะมีความเร็วในการทำงานและความสว่างในระดับพอดี โดยมีกำลังตั้งแต่ 10 ถึง 150 มิลลิวัตต์ อายุการใช้งานมากกว่า 100,000 ชั่วโมง



รูปที่ 2.12 การตรวจสอบไดโอดเปล่งแสง

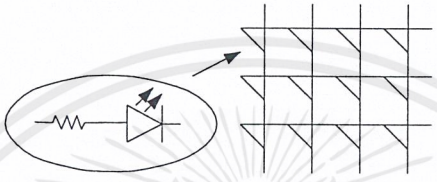
วิธีทดสอบไดโอดเปล่งแสง เราสามารถทดสอบได้ว่าดีหรือเสีย โดยการจ่ายไบอัส (Bias) ให้กับไดโอดเปล่งแสง โดยตั้งมิเตอร์ Rx1วัด เข็มมิเตอร์จะขึ้นครั้งหนึ่งและไม่ขึ้นอีกครั้งหนึ่ง เหมือนกับการวัดไดโอดทั่วไป ครั้งที่วัดขึ้นก็คือ ฟอร์เวิร์ด ไดโอดจะเปล่งแสงออกมาให้เราเห็น ฉะนั้นเวลาวัดเราอาจจะสังเกตจากแสงก็ได้

ความเป็นจริงแล้วพลังงานโฟตอนนั้นมีอยู่สองแบบ คือ พลังงานที่มนุษย์สามารถมองเห็นได้ด้วยตาเปล่า กับพลังงานแสงที่มนุษย์ไม่สามารถมองเห็นด้วยตาเปล่าได้ ปัจจุบันการใช้พลังงานทางแสงได้รับการพัฒนาไปมาก เช่น ในวงการไฟเบอร์ออพติค (Fiber Optic) ซึ่งเป็นอิเล็กทรอนิกส์ที่ใช้พลังงานของแสงเป็นตัวหลัก และมีบทบาทในวงการสื่อสารโทรคมนาคมหรือในงานเล็กๆ เราก็ยังนำอุปกรณ์เหล่านี้มาประยุกต์ใช้งานอย่างกว้างขวาง เช่น นำไดโอดเปล่งแสง มาประกอบกับโฟโต้ทรานซิสเตอร์ (Photo Transistor) โฟโต้ไดโอด (Photo Diode) เพื่อให้อุปกรณ์เหล่านี้ทำงานตามความเข้มของแสงที่ออกมา เราเรียกวิธีการเหล่านี้ว่า ออพโต้ไอโซเลต (Opto Isolate)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.12 ดอตเมตริกซ์

การแสดงผลอย่างง่ายๆ คือ การแสดงผลแบบแถวเดียวจะควบคุมสัญญาณเพียงด้านเดียว คือ ส่งแต่ข้อมูลแล้วหน่วงเวลาไว้ระยะหนึ่งแล้วจึงส่งสัญญาณตัวต่อไปก็จะทำให้เราเห็นเป็นไฟวิ่ง ได้จากนั้นเราสามารถแสดงให้เห็นเป็นตัวอักษรหรือรูปภาพก็ได้ถ้าเราเพิ่มลอจิกการควบคุมซึ่งแทนที่จะควบคุมการติดดับเพียงด้านเดียวก็เป็น 2 ด้าน



รูปที่ 2.13 การต่อไดโอดเปล่งแสงแบบเมตริกซ์

จะเห็นว่าถ้าเราส่งข้อมูลออกไปทางแถวและให้คอลัมน์ใดเป็นศูนย์ (0) จะทำให้ไดโอดเปล่งแสงถูกไบอัสตรง และติดสว่างตามบิตข้อมูลที่ส่งออกไปทางแถว จากหลักการนี้จึงสามารถควบคุมไดโอดเปล่งแสงทุกดวง ในแผงเมตริกซ์ (Matrix) ให้ติดหรือดับได้ตามต้องการ

หลักการเกิดภาพบนไดโอดเปล่งแสงแบบเมตริกซ์เนื่องจากการมองเห็นของตามนุษย์นั้นจะมีการคงภาพ คือ เห็นภาพอยู่นาน 1/6 วินาที ถึงแม้ว่าภาพนั้นจะไม่ปรากฏแล้วก็ตาม จากหลักการของการมองเห็นนี้ ถ้าเราทำให้ไดโอดเปล่งแสง ติดและดับตามจุดต่างๆ ได้เร็วกว่า 1 / 6 เราก็จะเห็นว่าไดโอดเปล่งแสงนั้นติดค้างอยู่

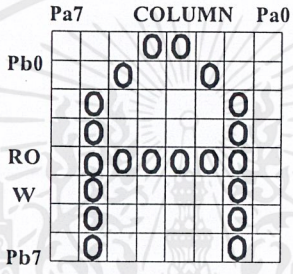
แต่ซีพียูทำงานด้วยความเร็วสูงการที่จะติดและเปลี่ยนตำแหน่งการติดไปเรื่อยๆ ตามความเร็วของซีพียูจะเห็นว่าไดโอดเปล่งแสงนั้นติดสว่างเรื่องๆ ทุกดวง ก็เพราะว่าความสัมพันธ์ของกระแสความอิ่มตัวของทรานซิสเตอร์ รวมทั้งกระแสที่ทำให้ไดโอดเปล่งแสงติด เป็นปัจจัยทำให้เห็นไดโอดเปล่งแสงติดทุกดวง เมื่อไดโอดเปล่งแสงดวงใดติดแล้ว ควรทำการหน่วงเวลาเพื่อให้มีกระแสเพียงพอที่จะทำให้ไดโอดเปล่งแสงติดสว่าง ยิ่งหน่วงเวลานานเท่าไรยิ่งทำให้กระแสเข้าใกล้จุดสูงสุด แต่เนื่องจากเราต้องทำการกวาด (Scan) หลายๆ คอลัมน์ถ้าทำการหน่วงเวลามากเกินไปจะทำให้ไดโอดเปล่งแสงในลำดับถัดไปมีวงรอบในการนำกระแสเป็นช่วงๆ ซึ่งระยะเวลาในการนำกระแสของไดโอดเปล่งแสงแต่ละดวงห่างไกลจากความคงแสงของตามนุษย์ ซึ่งมีผลทำให้เห็นเป็นภาพกระพริบหรือพริ้ว ในการกวาดแบบคอลัมน์นี้จะเห็นว่าเวลาที่จะทำให้ ไดโอดเปล่งแสงแต่ละดวงติดนานเท่าไรนั้นขึ้นอยู่กับจำนวนคอลัมน์ของไดโอดเปล่งแสงด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถแก้ไขได้โดยเมื่อจำนวนคอลัมน์มากเกินไปเราควรเปลี่ยนมาเป็นการกวาดทางแถวแทนซึ่งจะทำให้เวลาของไดโอดเปล่งแสงแต่ละดวงที่จะติด มีเวลาคงที่มากยิ่งขึ้น และความสว่างคงที่มากขึ้น

การคิดข้อมูลที่จะนำมาแสดงเป็นตัวอักษร มิได้ถูกกำหนดเป็นมาตรฐานหรือเป็นทฤษฎีแต่ขึ้นอยู่กับ ฮาร์ดแวร์นั้นๆ ฉะนั้นข้อมูลที่ทำให้ไดโอดเปล่งแสงติดได้ คือ ข้อมูลจะต้องเป็นลอจิกบวก และถ้าจะทำให้ไดโอดเปล่งแสงดับข้อมูลจะต้องเป็นลอจิกลบ

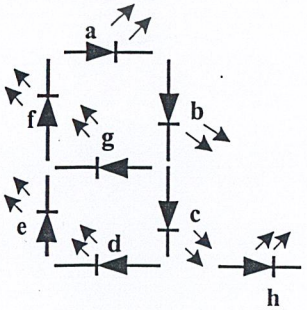
ตัวอย่าง การกำหนดตัวอักษร (Character Gen)



รูปที่ 2.14 การสร้างตัวอักษร

2.13 LED ตัวเลข 7 ส่วน

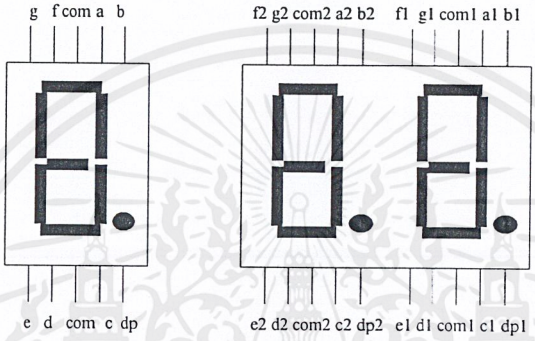
LED ตัวเลข 7 ส่วนประกอบขึ้นจาก LED จำนวน 7 ตัวที่บรรจุอยู่ในตัวถังเดียวกันและได้รับการจัดเรียงเป็นรูปตัวเลข LED แต่ละตัวจะถูกเรียกว่า ส่วน หรือ เซกเมนต์ (Segment) แต่ละส่วนหรือเซกเมนต์มีชื่อเรียกแตกต่างกันตามตำแหน่งที่ได้รับการจัดวาง คือ a, b, c, d, e, f และ g ส่วน db เป็น LED อีก 1 ตัวที่บรรจุอยู่ใน LED ตัวเลข 7 ส่วนนี้ใช้เป็นตัวแสดงจุดทศนิยมในกรณีที่มีการแสดงผลในลักษณะเลขที่มีทศนิยม



รูปที่ 2.15 การจัดวางไดโอดเปล่งแสงเพื่อแสดงผลแบบ 7 ส่วน

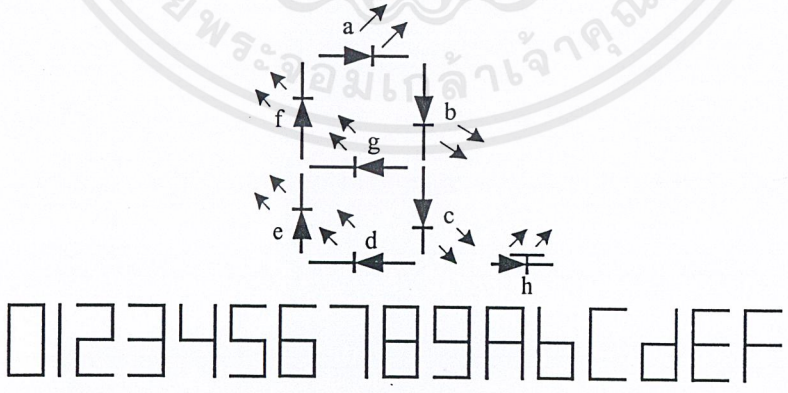
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LED ทุกตัวที่บรรจุอยู่ใน LED ตัวเลข 7 ส่วนนี้มีขาต่อร่วมกัน ซึ่งก็มีทั้งแบบต่อขาคาโถดร่วมกันเรียกว่า แบบคาโถดร่วม (Common Cathode) และแบบต่อขาอาโนดร่วมกันเรียกว่า แบบอาโนดร่วม (Common Anode) การขับให้ LED ตัวเลข 7 ส่วนแบบคาโถดร่วมสว่างจะต้องจ่ายไฟลบเข้าที่ขาร่วม แล้วจ่ายไฟบวกเข้าที่ขาอาโนด ซึ่งก็คือขาของแต่ละเซกเมนต์ ในขณะที่ LED ตัวเลข 7 ส่วนแบบอาโนดร่วมจะต้องจ่ายไฟบวกเข้าที่ขาร่วม แล้วจ่ายไฟลบเข้าที่ขาคาโถด ซึ่งเป็นขาของแต่ละเซกเมนต์



รูปที่ 2.16 การจัดขาของ LED ตัวเลข 7 ส่วน ทั้งแบบตัวถังเดี่ยวและตัวถังคู่

การนำข้อมูลออกไดโอดเปล่งแสงแบบ 7 ส่วน ไดโอดเปล่งแสง เป็นอุปกรณ์ที่ใช้แสดงผลของเอาต์พุตอย่างง่ายๆ และไดโอดเปล่งแสงแบบ 7 ส่วนจะใช้สำหรับแสดงตัวเลขฐานสิบ 0 ถึง 9 หรือแสดงแลขฐานสิบหก 0 ถึง F



รูปที่ 2.17 ไดโอดเปล่งแสงแบบ 7 ส่วน และการแสดงตัวเลขของฐานสิบและฐานสิบหก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้ากำหนดให้การแสดงผลแบบ 7 ส่วนนี้เป็นแบบคาโถดร่วม และถ้าให้พอร์ตเอาต์พุตบิต D0 ต่ออยู่กับส่วน a บิต D1 ต่ออยู่กับส่วน b ตามลำดับ เมื่อบิตใดมีระดับเป็น 1 จะทำให้ส่วนนั้นติด คือ เมื่อต้องการให้การแสดงผลแบบ 7 ส่วน ติดเป็นเลขใด ก็ส่งรหัสนั้นออกทางพอร์ตเอาต์พุต

ตารางที่ 2.7 ข้อมูลดิจิตอลเอาต์พุตและค่าเลขฐานสิบหกเพื่อการแสดงตัวเลขแบบ 7 ส่วน

ข้อมูลดิจิตอลเอาต์พุตสำหรับขับLEDตัวเลข 7ส่วน								ค่าเลขฐานสิบหกที่ใช้กับ	ค่าตัวเลข
D7	D6	D5	D4	D3	D2	D1	D0	PIC16F876	
0	0	1	1	1	1	1	1	X03F	0
0	0	0	0	0	1	1	0	0x06	1
0	1	0	1	1	0	1	1	0x5B	2
0	1	0	0	1	1	1	1	0x4F	3
0	1	1	0	0	1	1	0	0x66	4
0	1	1	0	1	1	0	1	0x6D	5
0	1	1	1	1	1	0	1	0x7D	6
0	0	0	0	0	1	1	1	0x07	7
0	1	1	1	1	1	1	1	0x7F	8
0	1	1	0	1	1	1	1	0x6F	9
0	1	1	1	0	1	1	1	0x77	A
0	1	1	1	1	1	0	0	0x7C	b
0	0	1	1	1	0	0	1	0x39	C
0	1	0	1	1	1	1	0	0x5E	d
0	1	1	1	1	0	0	1	0x79	E
0	1	1	1	0	0	0	1	0x71	F

2.14 สวิตช์เมตริกซ์

การต่อสวิตช์ (Switch) แบบปกติจะใช้ 1 บิตต่อสวิตช์ 1 ตัว แต่ในการใช้งานจริงของระบบไมโครคอมพิวเตอร์ต้องการใช้สวิตช์หรือตัวอักษรบนแป้นเป็นจำนวนมาก ดังนั้นถ้าใช้ 1 สวิตช์ต่อ 1 บิต จะทำให้สิ้นเปลืองพอร์ตเป็นจำนวนมาก เช่น ถ้าต้องการใช้ 64 สวิตช์ จะต้องใช้พอร์ตแบบขนาน 8 บิต ถึง 8 พอร์ต วิธีการต่อสวิตช์แบบที่ประหยัดพอร์ตและเป็นที่นิยมใช้แบบหนึ่ง คือ การต่อแบบเมตริกซ์ ซึ่งจะทำให้ลดจำนวนพอร์ตลง

เมื่อจำนวนของสวิตช์มากขึ้นก็จะทำให้ประหยัดสายข้อมูลมากขึ้นไปด้วย ข้อยุ่งยากทางด้านฮาร์ดแวร์จะลดลง และการตรวจสอบสวิตช์ที่กดจะใช้วิธีทางซอฟต์แวร์ช่วยจัดการลักษณะการทำงานของวงจรแป้นพิมพ์ขนาด 3 x 3 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.8 เปรียบเทียบจำนวนของสายเมื่อต่อแบบธรรมดาและแบบเมตริกซ์

จำนวนสวิตช์	จำนวนสาย	
	ต่อแบบธรรมดา	ต่อแบบเมตริกซ์
1 x 1	1	1
2 x 2	4	4
3 x 3	9	6
4 x 4	16	8
5 x 5	25	10
8 x 8	64	16

หลักการทำงานของโปรแกรมจะใช้วิธีการกวาด (Scanning) เพื่อหาว่าสวิตช์ใดที่ถูกกดโดยอาจเริ่มต้นที่แถวที่ 0 โดยกำหนดให้แถวที่ 0 มีระดับเป็น 0 ส่วนแถวอื่นๆ มีระดับเป็น 1 แล้วทำการตรวจสอบที่ละคอลัมน์ (Column) โดยเริ่มจากคอลัมน์ที่ 0 ตรวจสอบว่ามีระดับเป็น 0 หรือไม่ ถ้ามีระดับเป็น 0 แสดงว่าสวิตช์ที่ตำแหน่งนี้ถูกกด ถ้าไม่มีก็เลื่อนไปคอลัมน์ถัดไป และเมื่อพบว่าสวิตช์ใดถูกกด แล้วต้องทราบว่าสวิตช์นั้นคือสวิตช์ตำแหน่งใด เพื่อที่จะได้ตีความหมายหรือสร้างรหัสขึ้นมาตามสวิตช์

2.15 จอแสดงผลแบบผลึกเหลว

จอแสดงผลแบบผลึกเหลว (Liquid Crystal Display : LCD) ซึ่งเราจะใช้สำหรับวงจรแสดงผลระบบตัวเลขในเครื่องใช้อิเล็กทรอนิกส์ในยุคปัจจุบัน ซึ่งอยู่ในภาคแสดงผล (Display) ของเครื่องคิดเลข นาฬิกา วงจรแสดงผลทางดิจิตอล

ผลึกเหลว (Liquid Crystal) อันเป็นตัวหลักของจอแสดงผลแบบผลึกเหลวนี้ การรวมตัวแบบนี้ เรียกว่า การรวมตัวแบบนีแมติคลิควิดคริสตอล (Nematic Liquid Crystal)

2.15.1 ประเภทของจอแสดงผลแบบผลึกเหลว

1) จอแสดงผลแบบผลึกเหลวชนิดอักขระ

เป็นจอแสดงผลแบบผลึกเหลวที่สามารถแสดงตัวอักษร ตัวเลขและเครื่องหมายต่างๆ ได้โดยสร้างจากจุดเล็กๆ ทางแนวตั้งและแนวนอน หรือเรียกว่า (Dot Matrix) โดยทั่วๆ ไปมี 2 ขนาด คือ 5 x 7 จุด และ 5 x 10 จุด (ตัวเลขตัวแรกเป็นจำนวนจุดทางแนวนอน ส่วนตัวหลังเป็นจำนวนจุดทางแนวตั้ง) นอกจากนั้นจอแสดงผลแบบผลึกเหลวแบบนี้ยังสามารถแสดงข้อความได้ 1 บรรทัด หรือมากกว่าก็ได้ขึ้นอยู่กับรุ่นของจอแสดลผลแบบผลึกเหลวนั้นๆ

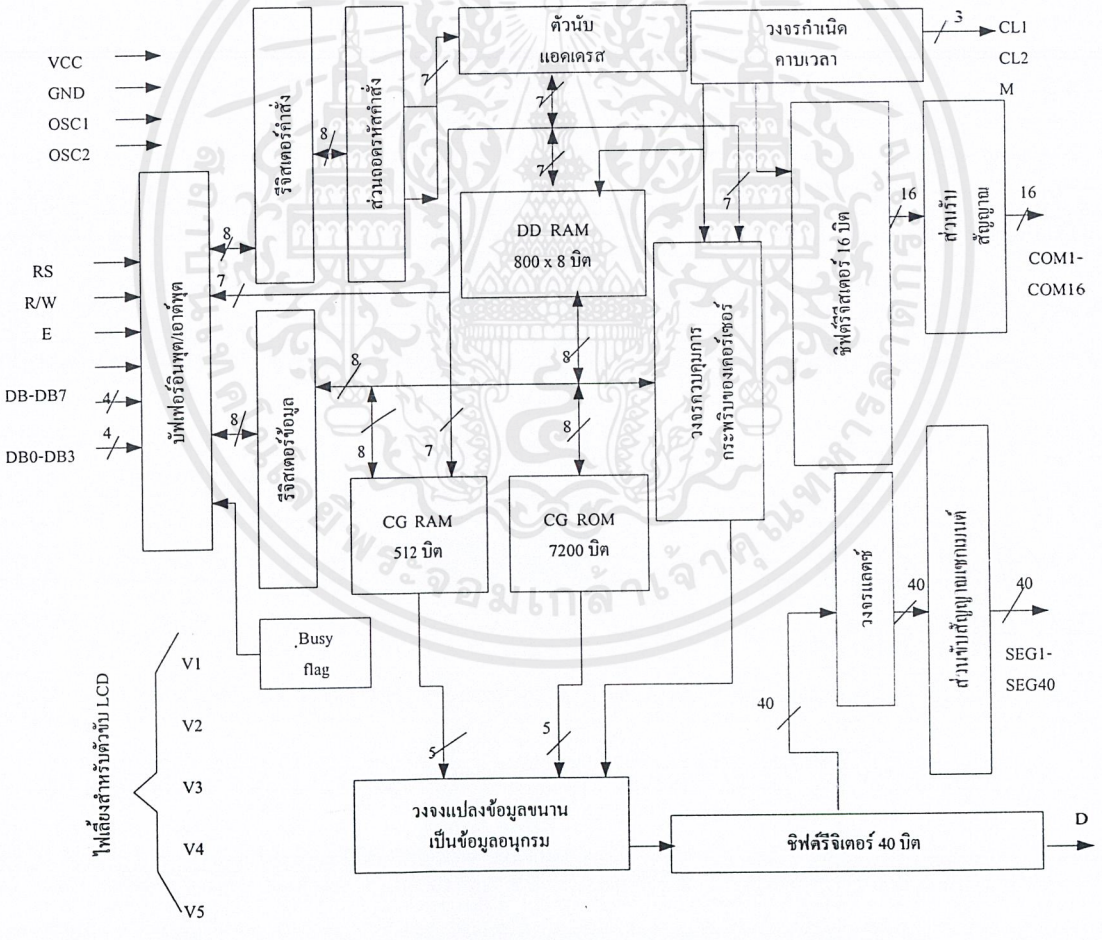
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) จอแสดงผลแบบผลึกเหลวชนิดกราฟิก

มีลักษณะโครงสร้างคล้ายกับจอแสดงผลแบบผลึกเหลวชนิดอักขระ สามารถที่จะแสดงข้อมูลเป็นทั้งตัวอักษร ตัวเลข เครื่องหมายและรูปภาพได้ ความละเอียดของภาพก็จะขึ้นอยู่กับความละเอียดของดอตเมตริกซ์ของจอแสดงผลแบบผลึกเหลวตัวนั้นๆ และขนาดของจอแสดงผลแบบผลึกเหลวชนิดนี้มีหลายขนาดให้เลือกใช้ในปัจจุบันมีการพัฒนาเป็นสีแล้ว

3) จอแสดงผลแบบผลึกเหลวชนิดเซกเมนต์

เป็นจอแสดงผลแบบผลึกเหลวชนิดที่เล็กที่สุด มีลักษณะการแสดงผลเป็นเซกเมนต์คล้ายกับตัวเลขแสดงผล 7 ส่วน โดยปกติมักจะมีมากกว่า 1 หลัก พบเห็นทั่วไปในดิจิตอลมัลติมิเตอร์ (Digital Multimeter)



รูปที่ 2.18 แผนผังการทำงานของจอแสดงผลแบบผลึกเหลวชนิดอักขระ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.15.2 ส่วนประกอบหลักในจอแสดงผลแบบผลึกเหลว

1) ตัวแสดงผล (Display)

ภายในเป็นผลึกเหลวที่สามารถแสดงผลให้เห็นโดยอาศัยแสงจากภายนอก ดังนั้นจึงต้องมีมุมในการมองข้อมูลที่แสดงผลบนจอแสดงผลแบบผลึกเหลว

2) ตัวควบคุม (Controller)

เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอกมาควบคุมการทำงานของจอแสดงผลแบบผลึกเหลว เช่น ลบจอภาพ แสดงตัวอักษร หรือเลื่อน เคอร์เซอร์ (Cursor) เป็นต้น ตัวควบคุมนี้ใช้ชิป (Chip) ควบคุมโดยเฉพาะ

3) ตัวขับ (Driver)

เป็นตัวรับสัญญาณจากตัวควบคุมมาขับให้ตัวแสดงผลแสดงข้อมูลตามที่กำหนด

2.15.3 โครงสร้างภายในของตัวควบคุมจอแสดงผลแบบผลึกเหลว

จอแสดงผลแบบผลึกเหลว แบบอักขระ ประกอบด้วย

1) บัฟเฟอร์อินพุตเอาต์พุต

เป็นส่วนที่ใช้ในการติดต่อรับส่งข้อมูลกับอุปกรณ์ภายนอก เพื่อที่จะถ่ายทอดข้อมูลเข้าออกภายในตัวควบคุม

2) รีจีสเตอร์คำสั่ง (Instruction Register :IR)

เป็นรีจีสเตอร์ใช้รับคำสั่งจากอุปกรณ์ภายนอกเพื่อนำไปควบคุมการแสดงผล

3) รีจีสเตอร์ข้อมูล (Data Register :DR)

ใช้รับข้อมูลจากอุปกรณ์ภายนอกเพื่อถ่ายทอดต่อไปยังหน่วยความจำทำหน้าที่เก็บข้อมูลแสดงผล หรือนำข้อมูลไปสร้างตัวอักษรเพิ่มเติมในแรมเก็บตัวอักษร

แรมเก็บข้อมูลแสดงผล (Display Data RAM : DDRAM) เป็นหน่วยความจำแรมทำหน้าที่เก็บข้อมูลที่มาจากรีจีสเตอร์ DR ตัวควบคุมจะนำข้อมูลใน DDRAM นี้ไปเปิดตาราง (Look-up Table) ของตัวอักษรที่เก็บไว้ในหน่วยความจำรอมและแรมเก็บตัวอักษรเพื่อแสดงที่ตัวแสดงผล

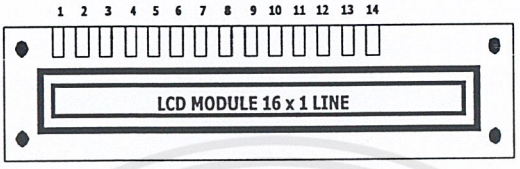
รอมเก็บตัวอักษร (Character Generator ROM : CGROM) เป็นหน่วยความจำรอมที่ใช้เก็บข้อมูลตัวอักษรหรือสัญลักษณ์ที่สามารถอ่านออกไปแสดงที่ตัวแสดงผลได้ มีขนาด 7200 บิต โดยจะถูกอ่านด้วยค่าของข้อมูลใน DDRAM

แรมเก็บตัวอักษร (Character Generatoe RAM : CGRAM) เป็นหน่วยความจำแรมที่ใช้เก็บอักษรที่มีการสร้างเพิ่มเติมขึ้นใหม่ในกรณีที่ตัวอักษรใน CGROM ไม่เพียงพอ มีขนาด 512 บิต การเขียนและอ่านค่าไปใช้นั้นทำได้เช่นเดียวกับ CGROM คือเขียนข้อมูลลงใน DDRAM แล้วตัวควบคุมจะมาอ่านค่าจาก CGRAM เอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แฟลก BUSY ทำหน้าที่แจ้งสถานะการทำงานของตัวควบคุมให้อุปกรณ์ภายนอกทราบว่าตัวควบคุมพร้อมที่จะรับข้อมูลหรือคำสั่งหรือไม่ ดังนั้นก่อนการส่งข้อมูลหรือคำสั่งมายังตัวควบคุมต้องตรวจสอบสถานะของแฟลก BUSY นี้เสียก่อน

2.15.4 จอแสดงผลแบบผลึกเหลว ขนาด 16 ตัวอักษร 1 บรรทัด (LCD 16x1)



รูปที่ 2.19 รูปร่างและการจัดขาจอแสดงผลแบบผลึกเหลว แบบอักขระ

จอแสดงผลแบบผลึกเหลว ขนาด 16x1 มีขาต่อใช้งานทั้งสิ้น 14 ขา รายละเอียดการทำงานของแต่ละขามีดังนี้

V_{SS} (ขา 1) : ต่อกราวด์

V_{DD} (ขา 2) : ต่อไฟเลี้ยง +5 โวลต์

V_O (ขา 3) : เป็นขาอินพุตรับแรงดันเพื่อปรับความเข้มของการแสดงผล

RS (ขา 4) : เป็นขาอินพุตใช้ในการแยกชนิดของข้อมูลที่ทำการประมวลผลในขณะนั้นว่าเป็นคำสั่งสำหรับรีจีสเตอร์ IR หรือเป็นข้อมูลสำหรับรีจีสเตอร์ DR โดยถ้าขานี้เป็น “0” ข้อมูลที่ส่งมาจะเป็นคำสั่งแต่ถ้าขานี้เป็น “1” ข้อมูลที่ส่งมาจะเป็นข้อมูลสำหรับการแสดงผล

R / W (ขา 5) : เป็นขาที่ใช้เลือกการอ่านหรือเขียนข้อมูลกับ LCD ถ้าเป็น “0” เป็นการกำหนดให้เขียนข้อมูล แต่ถ้าเป็น “1” จะเป็นการอ่านข้อมูล

E (ขา 6) : เป็นขาอีนาเบิล LCD ให้ทำงาน

D0-D7 (ขา 7-14) : เป็นขาที่ใช้เป็นทางผ่านของข้อมูลระหว่าง LCD กับอุปกรณ์ภายนอก

2.16 I^2C

I^2C (Inter-IC Communication) หมายถึง การติดต่อสื่อสารระหว่างไอซีโดยบัส I^2C การติดต่อสื่อสารแบบ นี้ ได้รับการพัฒนาขึ้นโดยฟิลิปส์ (Philips) ด้วยจุดมุ่งหมายหลักคือ ต้องการติดต่อให้ไอซี หรือโมดูลสามารถติดต่อสั่งงานและควบคุมภายใต้สายสัญญาณเพียง 2 เส้น เส้นหนึ่งคือ สายข้อมูล อีกเส้นคือ สายสัญญาณนาฬิกาที่ใช้กำหนดจังหวะการทำงาน การต่อร่วมกันของอุปกรณ์บนบัส I^2C ทำได้ง่ายมาก เพียงต่อสายข้อมูลและสายสัญญาณนาฬิกาของอุปกรณ์แต่ละตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขนานหรือพ่วงกันไป ส่วนการกำหนดแอดเดรสหรือตำแหน่งสำหรับติดต่ออุปกรณ์แต่ละตัวจะใช้รหัสข้อมูลและการกำหนดสภาวะลอจิกที่ขาแอดเดรสของอุปกรณ์แต่ละตัว

สายข้อมูลบนบัส I^2C มีชื่อเรียกอย่างเป็นทางการว่า สายข้อมูลอนุกรมหรือ SDA (Serial Data Line) ส่วนสายสัญญาณนาฬิกามีชื่อเรียกว่า สายสัญญาณนาฬิกาอนุกรมหรือ SCL (Serial Clock Line) อุปกรณ์ที่ทำการเชื่อมต่อบนบัส I^2C มีหลากหลายไม่ว่าจะเป็นไอซีขยายพอร์ตอินพุตเอาต์พุต (I/O Expander) ไอซีแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอล (ADC) และแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะลอก (DAC) ไอซีเรียลไทม์คล็อก (RTC) ไอซีขับจอแสดงผลแบบผลึกเหลว หน่วยความจำอีพรอม (EPROM) และไมโครคอนโทรลเลอร์

คุณสมบัติโดยทั่วไปของบัส I^2C สาย SDA และ SCL เป็นสายสัญญาณแบบ 2 ทิศทาง (Bi-Directional Line) ต้องมีการต่อตัวต้านทานพูลอัพกับแรงดัน +5 โวลต์ เพื่อให้สถานะลอจิกสูงขณะที่ไม่มีการติดต่อใช้งาน และมีลักษณะเป็นวงจรเดรนเปิด (Open-Drain) หรือคอลเล็กเตอร์เปิด (Open-Collector)

อัตราการถ่ายทอดข้อมูลบนบัส I^2C สูงถึง 100 กิโลไบต์ ต่อ วินาทีในโหมดปกติ (Standard Mode) และสูงถึง 400 กิโลบิต ต่อ วินาทีในโหมดความเร็วสูง (Fast Mode) อุปกรณ์ที่ต่อร่วมอยู่บนบัสแบบ I^2C นี้จะต้องมีค่าความจุไฟฟ้ารวมที่เกิดขึ้นระหว่างสาย SDA และ SCL ไม่เกิน 400 พิโคฟารัด การเข้าถึงอุปกรณ์บนบัส I^2C ใช้ข้อมูลสำหรับการเข้าถึง 2 ค่า คือ 7 บิต (7-Bit Addressing) หรือ 10 บิต (10-Bit Addressing)

ข้อเด่นอีกประการหนึ่งของบัส I^2C คือ สามารถเชื่อมต่ออุปกรณ์ที่ใช้ไฟเลี้ยงไม่เท่ากันให้ใช้ไฟเลี้ยง +12 โวลต์ การต่อร่วมกันบนบัส I^2C สามารถกระทำได้ในลักษณะเดียวกับกรณีที่อุปกรณ์ทั้งสองใช้ไฟเลี้ยงเท่ากัน

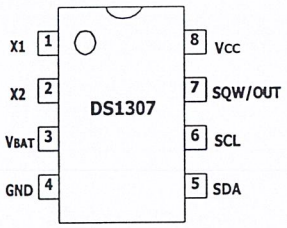
2.17 DS1307 ไอซีสร้างฐานเวลาหรือเรียลไทม์คล็อก (RTC)

ผู้ผลิตคือ ดัลลัสเซมิคอนดัคเตอร์ (Dallas Semiconductor) มีหน้าที่สร้างฐานเวลาจริงให้แก่ระบบไมโครคอนโทรลเลอร์โดย DS1307 จะให้ข้อมูลเกี่ยวกับเวลาทั้งหมด

เป็นไอซีเรียลไทม์คล็อกให้ข้อมูลตั้งแต่วินาทีจนถึงปี รวมถึงการปรับวันในปีอธิกสุรทิน สามารถให้ข้อมูลเวลาได้อย่างเที่ยงตรงถึงปีคริสตศักราช 2100 มีหน่วยความจำนอนโวลาไทม์แรม (Non- Vola Time RAM) 56 ไบต์ อยู่ภายใน สามารถใช้เก็บข้อมูลทั่วไปได้และใช้การเชื่อมต่อแบบระบบบัส I^2C มีวงจรตรวจจับไฟเลี้ยงต่ำหรือหายไปอย่างอัตโนมัติ และสามารถรักษาข้อมูลเวลาไว้ได้แม้ไม่มีไฟเลี้ยง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.17.1 รายละเอียดขาต่อใช้งานของ DS1307



รูปที่ 2.20 การจัดขาของไอซี DS1307

V_{cc} และ GND (ขา 8,4) ต่อกับไฟเลี้ยง +5 โวลต์

V_{BAT} (ขา 3) ใช้ต่อกับแบตเตอรี่ 3 โวลต์ เพื่อรักษาการทำงานของวงจรสร้างฐานเวลาของ DS1307 ให้คงอยู่ต่อไปแม้ว่าไม่มีไฟเลี้ยงจ่ายให้แก่ DS1307 ชนิดของแบตเตอรี่ที่เหมาะสมคือแบตเตอรี่แบบลิเธียม (Lithium) หรือมากกว่า จะสามารถรักษาข้อมูลได้นาน 10 ปีที่อุณหภูมิ 25 องศาเซลเซียส

SDA และ SCL (ขา 5 และ 6) สำหรับเชื่อมต่อกับไมโครคอนโทรลเลอร์บนระบบบัส I^2C

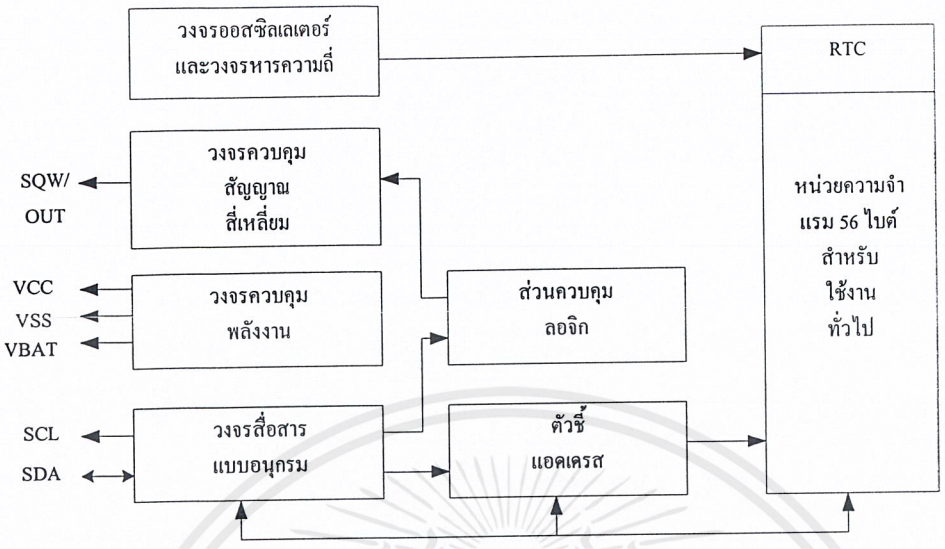
SQW/OUT (ขา 7) ที่ขานี้จะมีสัญญาณรูปสี่เหลี่ยมส่งออกมา โดยสามารถเลือกความถี่ได้ 1 เฮริตซ์ 4.096 กิโลเฮริตซ์ 9.192 กิโลเฮริตซ์ ในการใช้งานต้องต่อตัวต้านทาน 1 กิโลโอห์มพูลอัพที่ขานี้ด้วย

X1 และ X2 (ขา 1 และ 2) ใช้ต่อกับคริสตอลความถี่มาตรฐาน 32.768 กิโลเฮริตซ์ เพื่อใช้เป็นฐานเวลาในการสร้างค่าเวลาจริง ในการใช้งานต้องต่อคริสตอลเข้ากับขาทั้งสองนี้และที่แต่ละขาต้องต่อตัวเก็บประจุค่า ต่ำๆ ประมาณ 15 พิโคฟารัดคร่อมกับขากราวด์

2.17.2 การทำงานของ DS1307

ไอซี DS1307 จัดการเชื่อต่อในแบบบัส I^2C โดยจะทำงานเป็นอุปกรณ์สเลฟ (Slave) เสมอ ดังนั้นการติดต่อเพื่อใช้งานจึงต้องกำหนดรูปแบบตามที่กำหนดไว้ในการติดต่อแบบ I^2C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.21 โครงสร้างภายในของ DS1307

วงจรออสซิลเลเตอร์ถือเป็นหัวใจหลักของไอซี เนื่องจากเป็นจุดเริ่มต้นของการสร้างข้อมูลเวลาจริง ในขณะที่ DS1307 ทำงานที่ขา SQW/OUT จะมีสัญญาณพัลซ์สี่เหลี่ยมส่งออกมาตลอดเวลาในกรณีที่มีการอีนาเบิลวงจรกำเนิดสัญญาณพัลซ์ที่รีจีสเตอร์ควบคุม ค่าความถี่ของสัญญาณนี้สามารถเลือกได้ 4 ค่า คือ 1 เฮริตซ์ 4.096 กิโลเฮริตซ์ 9.192 กิโลเฮริตซ์ และ 32 กิโลเฮริตซ์ พร้อมกันนั้นก็จะมีการเก็บค่าของเวลาในหน่วยความจำนอนโวลาไทล์แรมซึ่งมีขนาดรวมทั้งหมด 64 ไบต์ แต่จัดสรรให้ใช้เก็บข้อมูลเวลา 8 ไบต์ และเป็นหน่วยความจำสำหรับเก็บข้อมูลทั่วไปสำหรับผู้ใช้งานอีก 56 ไบต์

วงจรสื่อสารอนุกรมภายใน DS1307 ได้รับการกำหนดให้ทำงานตามรูปแบบของบัส I^2C เป็นช่องทางการสื่อสารระหว่าง DS1307 กับอุปกรณ์มาสเตอร์ (Master) ผู้ใช้งานสามารถเข้าถึงหน่วยความจำที่ใช้เก็บค่าเวลาและหน่วความจำใช้งานทั่วไปได้ โดยการเขียนข้อมูลตามรูปแบบที่กำหนดในระบบบัส I^2C

2.17.3 โหมดการทำงานของ DS1307

มีด้วยกัน 2 โหมดคือ โหมดเขียนและโหมดอ่านข้อมูล ในการใช้งาน DS1307 ตามปกติจะใช้งานเฉพาะโหมดอ่านข้อมูลเท่านั้น เนื่องจากไมโครคอนโทรลเลอร์จะติดต่อกับ DS1307 เพื่ออ่านข้อมูลของเวลาไปใช้งานโหมดการเขียนข้อมูลจะถูกใช้งานก็ต่อเมื่อต้องการตั้งค่าเวลาใหม่และต้องการเขียนข้อมูลลงในหน่วยความจำใช้งานทั่วไป เมื่อเริ่มต้นติดต่อกับ DS1307 จำเป็นอย่างยิ่งที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะต้องเข้าสู่โหมดการเขียนข้อมูลก่อนเพื่อกำหนดแอดเดรสที่ต้องการอ่านข้อมูล จากนั้นจึงเปลี่ยนโหมดการทำงานมาเป็นโหมดการอ่านข้อมูลต่อไป

1) โหมดการเขียนข้อมูล

เริ่มต้นเมื่อไมโครคอนโทรลเลอร์ทำการกำหนดสภาวะเริ่มต้น (START : S) ส่งข้อมูลกำหนดแอดเดรส 11010000 ตามด้วยข้อมูลเลือกการเขียน นั่นคือค่า 0 จากนั้นจะรอการตอบรับจาก DS1307 ขั้นตอนต่อมาคือ ส่งข้อมูลเพื่อเลือกแอดเดรสที่ต้องการเขียนจากนั้นรอการตอบรับจาก DS1307 เมื่อมีการตอบรับมาเรียบร้อยแล้ว ก็เริ่มทยอยเขียนข้อมูลลงไปครั้งละแอดเดรส หลังจากการเขียนข้อมูลในแต่ละแอดเดรสจะต้องหยุดรอการตอบรับจาก DS1307 ทุกครั้งจึงจะสามารถเขียนข้อมูลต่อไปได้ เมื่อเขียนเรียบร้อยแล้วให้ส่งสภาวะหยุด (STOP : P) เป็นอันสิ้นสุดกระบวนการเขียนข้อมูล

2) โหมดการอ่านข้อมูล

เริ่มต้นการทำงานเหมือนกับโหมดการเขียนข้อมูลคือ ไมโครคอนโทรลเลอร์กำหนดสภาวะเริ่มต้นแล้วส่งข้อมูลกำหนดแอดเดรสต่อด้วยข้อมูลเลือกการอ่านซึ่งเท่ากับ 1 จากนั้นรอการตอบรับจาก DS1307 เมื่อตอบรับเรียบร้อยแล้ว DS1307 จะทยอยส่งข้อมูลออกมาให้ไมโครคอนโทรลเลอร์คราวละ 1 แอดเดรส หรือคราวละ 1 ไบต์ โดยแอดเดรสที่เลือกอ่านข้อมูลจะต้องมีการกำหนดมาก่อนล่วงหน้าด้วยการเขียนข้อมูล วิธีการง่ายๆ คือ เข้าสู่โหมดการเขียนข้อมูลก่อนเมื่อถึงจังหวะที่ต้องเขียนข้อมูลให้ทำการสร้างสภาวะเริ่มต้นและส่งข้อมูลกำหนดแอดเดรสใหม่อีกครั้ง ตามด้วยเลือกโหมดการอ่านข้อมูล ข้อมูลที่ออกมาจาก DS1307 ก็จะเป็นข้อมูลจาก แอดเดรสที่กำหนดไว้ก่อนหน้านี้

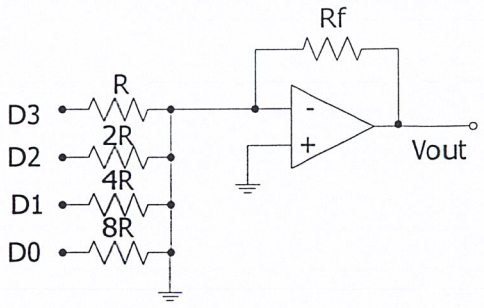
2.18 การแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอล

ปัจจุบันคอมพิวเตอร์คืออุปกรณ์ที่สามารถช่วยแบ่งเบาและทำงานแทนมนุษย์ได้เป็นอย่างดี แต่การที่จะติดต่อกับคอมพิวเตอร์ หรือการนำข้อมูลดิจิตอลแปลงเป็นข้อมูลแบบแอนะลอกจำเป็นต้องใช้วงจร Digital to Analog หรือเรียกสั้นๆว่า A to D วงจร A/D นี้แบ่งเป็นประเภทต่างๆ ดังนี้

2.18.1 แบบกำหนดน้ำหนักตัวต้านทาน

วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะลอกแบบกำหนดน้ำหนักตัวต้านทาน คือวิธีการกำหนดน้ำหนักตัวต้านทาน (Weighted resistor) ซึ่งก็คือ วงจรรวมสัญญาณ (Summing Amplifier)

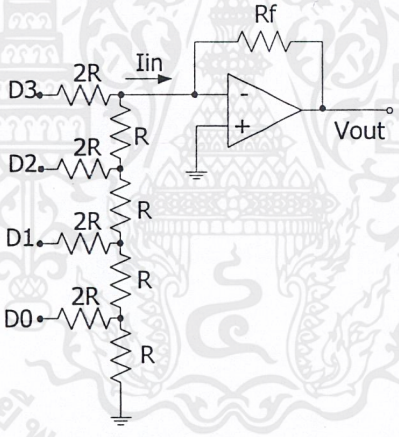
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.22 วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะลอกแบบกำหนดน้ำหนักตัวต้านทาน

จากรูปเป็นวงจร DAC 4 บิต โดยวงจรจะประกอบไปด้วยตัวต้านทาน 4 ตัว มีค่าความต้านทานแตกต่างกันตามสัดส่วน คือ 1 เท่า 2 เท่า 3 เท่า 4 เท่า และ 8 เท่า ตามลำดับ

2.18.2 วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะลอกแบบ R-2R แลดเดอร์



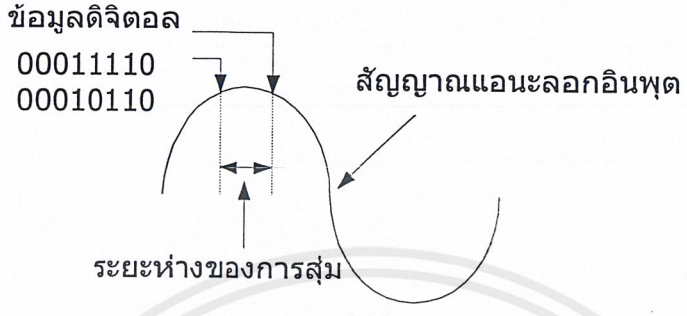
รูปที่ 2.23 วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะลอกแบบ R-2R แลดเดอร์

เป็นวงจรที่ได้รับความนิยมเป็นอย่างมาก เนื่องจากสามารถหาตัวต้านทานได้ง่ายกว่าแบบกำหนดน้ำหนัก โดยจะใช้ค่าความต้านทานเพียง 2 ค่า คือ R และ 2R เช่น ถ้า R มีค่า 10 กิโลโอห์ม 2R ก็จะมีค่า 20 กิโลโอห์ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

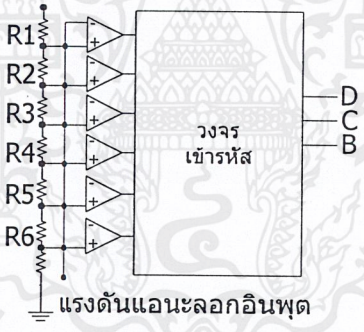
2.19 การแปลงสัญญาณดิจิตอลเป็นแอนะลอก

การแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอล โดยวิธีการสุ่ม (Sampling)



รูปที่ 2.24 การแปลงสัญญาณดิจิตอลเป็นแอนะลอก

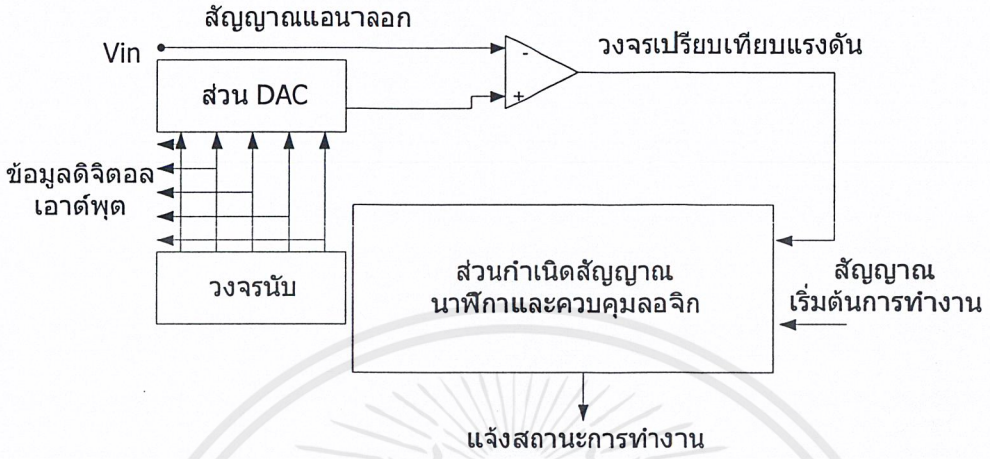
2.19.1 แบบเปรียบเทียบแรงดัน (Voltage Comparator ADC)



รูปที่ 2.25 วงจรแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.19.2 แบบวงจรนับแรมป์ (Simple Counter-Ramp ADC)



รูปที่ 2.26 แบบวงจรนับแรมป์ (Simple Counter-Ramp ADC)

1) ส่วนประกอบของวงจรนับแรมป์ (Ramp)

1.1) วงจรกำเนิดสัญญาณนาฬิกาและควบคุมลอจิก (Clock And Control Logic) ส่วนนี้จะได้รับสัญญาณให้เริ่มการทำงาน แล้วส่งสัญญาณนาฬิกาไปยังวงจรนับ

1.2) วงจรนับ (Counter) ส่วนนี้จะทำการนับค่าตามสัญญาณนาฬิกาที่ส่งเข้ามมาแล้วส่งต่อออกไปยังวงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะลอก

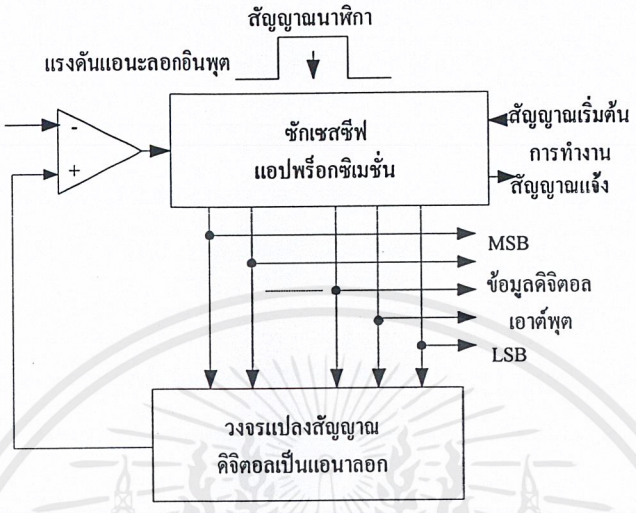
1.3) วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะลอก (DAC) ในส่วนนี้จะแปลงค่าของการนับจากวงจรนับเป็นแรงดันไฟตรง เพื่อส่งออกไปยังวงจรเปรียบเทียบแรงดัน

1.4) วงจรเปรียบเทียบแรงดัน (Comparator) จะทำหน้าที่เปรียบเทียบแรงดันจากวงจร DAC กับแรงดันแอนะลอกที่ส่งเข้ามาทางอินพุตของวงจร (Vin) หากยังไม่เท่ากันวงจรเปรียบเทียบแรงดันจะส่งสัญญาณไปยังวงจรกำเนิดสัญญาณนาฬิกาและส่วนควบคุมลอจิกให้ทำงานต่อไป เพื่อกระตุ้นให้วงจรนับและวงจร DAC ทำงาน จนกระทั่งแรงดันจากวงจร DAC มีค่าเท่ากับแรงดันอินพุต วงจรเปรียบเทียบจะทำการส่งสัญญาณควบคุมให้วงจรทั้งหมดหยุดทำงาน ค่าของวงจรนับครั้งสุดท้ายคือผลลัพธ์จากวงจร ADC

ข้อเสียของการแปลงสัญญาณแบบนี้คือ มีความเร็วต่ำ เวลาที่ใช้ในการแปลงจะขึ้นอยู่กับปริมาณของแรงดันอินพุตหากแรงดันอินพุตมีค่ามาก เวลาที่ใช้ก็จะมากตาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.19.3 แบบซักเซสซีฟแอปพร็อกซิเมชั่น



รูปที่ 2.27 แบบซักเซสซีฟแอปพร็อกซิเมชั่น

อาจเรียกว่าเป็นการแปลงแบบประมาณค่าใกล้เคียง ส่วนสำคัญหลักจะต้องมีวงจรเปรียบเทียบแรงดัน DAC สัญญาณนาฬิกา และส่วนควบคุมลอจิกเช่นเดียวกับวงจรแบบนับแรมป์

จุดแตกต่างอยู่ที่ในวงจรแบบซักเซสซีฟแอปพร็อกซิเมชั่นนี้จะใช้รีจีสเตอร์เลขฐานสอง หรือ ไบนารี่รีจีเตอร์แทนวงจรนับเลขฐานสอง แต่ละบิตของรีจีสเตอร์จะเซต และรีเซตโดยการควบคุมจากวงจรควบคุม

2.20 มาตรฐานพอร์ตอนุกรมแบบ RS-232

มาตรฐานการเชื่อมต่อแบบอนุกรม RS-232 เป็นมาตรฐานอุตสาหกรรมที่ออกแบบมาเพื่อใช้ในการส่งข้อมูลอนุกรมแบบอะซิงโครนัส 2 ทิศทาง มาตรฐานนี้ในช่วงแรกจะใช้คอนเนคเตอร์ (Connector) เป็นแบบ DB-25 โดยกำหนดความยาวสูงสุดของสายสัญญาณไว้ที่ 50 ฟุต มีระดับสัญญาณตั้งแต่ –3 ถึง –12 โวลต์ แสดงว่ามีข้อมูล (Mark) และ +3 โวลต์ ถึง +12 โวลต์ แสดงว่าเป็นช่องว่าง (Space)

มาตรฐาน RS-232 ได้กำหนดรูปแบบของอุปกรณ์เชื่อมต่อข้อมูล (Data Terminal Equipment : DTE) กับวงจรข้อมูลปลายทาง (Data Circuit Terminationg : DCE) ไว้ว่าอุปกรณ์ DTE จะต้องเป็นอุปกรณ์ที่มีการประมวลผลในตัวเช่น ไมโครคอนโทรลเลอร์ หรือ ไมโครคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Micro Computer) ซึ่งมีความสามารถในการสร้างบิตข้อมูลแบบอนุกรมได้ ส่วนอุปกรณ์ DCE จะทำหน้าที่เป็นเพียงตัวรับข้อมูลที่ส่งมาจาก DTE เท่านั้น โดยการรับส่งข้อมูลระหว่างอุปกรณ์ทั้งสองจะกระทำผ่านมาตรฐาน RS-232

ข้อแตกต่างของอุปกรณ์ DTE และอุปกรณ์ DCE อย่างหนึ่งที่เห็นได้ชัดคือ คอนเนคเตอร์ของ DTE จะเป็นตัวผู้ ส่วนคอนเนคเตอร์ของ DCE จะเป็นตัวเมีย ซึ่งพอร์ตอนุกรมของคอมพิวเตอร์ที่ใช้กันอยู่ทั่วไปจะเป็นแบบ DTE ส่วนคอนเนคเตอร์ที่อยู่ที่โมเด็มจะเป็นแบบ DCE

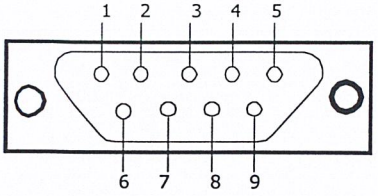
สำหรับการใช้งานบนคอมพิวเตอร์ พอร์ตอนุกรม RS-232 มักถูกใช้เชื่อมต่อกับโมเด็มหรือเมาส์ (Mouse) โดยสามารถรับส่งข้อมูลได้ที่ความยาวของสายสัญญาณสูงสุดถึง 20 เมตร

พอร์ตติดต่อสื่อสารกับอุปกรณ์ภายนอกของคอมพิวเตอร์แบบอนุกรมมาตรฐาน RS-232 นี้เป็นระบบการส่งข้อมูลในรูปแบบอนุกรม คือ ข้อมูลจะส่งไปได้ทีละ 1 บิต โดยจะมีอัตราการส่งเป็นบิตต่อวินาที หรืออัตราการเปลี่ยนแปลงของสัญญาณใน 1 วินาที เรียกว่า Baud Rate โดยจะมีการส่งบิตเริ่มต้น (Start Bit) มีระดับสัญญาณเป็น “0” และบิตข้อมูล (Data Bit) ซึ่งอาจจะมีข้อมูล 7 หรือ 8 บิต และอาจตามด้วยบิตพาริตี้ (Parity Bit) ซึ่งอาจจะเป็นแบบคู่ (Even) หรือแบบคี่ (Odd) เพื่อตรวจความถูกต้องของข้อมูล บิตพาริตี้นี้อาจจะมีหรือไม่มีก็ได้ และสุดท้ายจะตามด้วยบิตจบการทำงาน (Stop Bit) ซึ่งอาจจะมีความกว้างของสัญญาณเป็น 1 1.5 หรือ 2 บิตก็ได้ ซึ่งการส่งข้อมูลแบบนี้ จำเป็นต้องมีข้อตกลงกันระหว่างการรับกับการส่งคือ

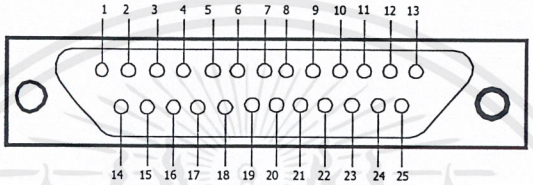
1) ความเร็วในการส่ง Baud Rate
2) จำนวนข้อมูล
3) มี บิตพาริตี้หรือไม่ ถ้ามีจะเป็นแบบคู่หรือคี่
4) จำนวนบิตเริ่มต้นเป็น 1 1.5 หรือ 2 บิต

มาตรฐานการเชื่อมต่อแบบ RS-232 จะใช้คอนเนคเตอร์แบบ DB-25 ตัวผู้หรือ DB-9 ตัวผู้ ซึ่งคอนเนคเตอร์แบบ DB-25 จะมีขาต่อใช้งานเพียง 9 เส้นเช่นเดียวกับคอนเนคเตอร์แบบ DB-9 เนื่องจากขาอื่นๆ ที่เคยใช้งานในอดีต ปัจจุบันมีการใช้งานไม่มากนัก จึงถูกยกเลิกไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.28 คอนเนคเตอร์อนุกรม 9 ขา หรือแบบ DB-9 (มองจากด้านหลังคอมพิวเตอร์)



รูปที่ 2.29 คอนเนคเตอร์อนุกรม 25 ขา หรือแบบ DB-25 (มองจากด้านหลังคอมพิวเตอร์)

สำหรับรายละเอียดหน้าที่การทำงานในแต่ละขาของพอร์ตอนุกรม RS-232 มีดังนี้

ขา 1 DCD : Carrier Detect หรืออาจเรียกว่า Carrier Detect : CD ขานี้จะแอกตีฟ (Active) เมื่อมีการส่งสัญญาณพาห์จากอุปกรณ์สื่อสารข้อมูล เช่น โมเด็ม สำหรับการใช้งานปกติขานี้จะไม่ได้ถูกใช้งานมากนัก

ขา 2 RXD : Receive Data ขานี้ใช้เพื่อรับสัญญาณอนุกรมเข้ามาคอมพิวเตอร์ โดยนำข้อมูลที่อ่านได้เก็บไว้ในรีจิสเตอร์บัฟเฟอร์

ขา 3 TXD : Transmited Data ขานี้ใช้เพื่อส่งข้อมูลออกจากคอมพิวเตอร์ โดยนำข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูลส่งออกไป

ขา 4 DTR : Data Terminal Ready เป็นขาสัญญาณที่ส่งออกจากคอมพิวเตอร์เพื่อให้อุปกรณ์ปลายทางรับรู้ว่า ต้องการติดต่อด้วย โดยขา DTR นี้จะต้องเชื่อมต่อกับขา DSR ของอุปกรณ์ปลายทาง และขา DTR ของอุปกรณ์ปลายทางจะต้องเชื่อมต่อกับขา DSR ของคอมพิวเตอร์ ถ้าใช้การเชื่อมต่อเป็นแบบ Null Modem ซึ่งใช้สายในการเชื่อมต่อเพียง 3 เส้น จะต้องต่อขา DTR และ DSR ของตัวมันเองเข้าด้วยกันและต้องต่อกับขา DCD ด้วย

ขา 5 GND : Signal Ground ขากราวด์ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขา 6 DSR : Data Set Readyใช้คู่กับขา DTR เพื่อตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง ซึ่งขา DSR นี้จะเป็นขาสำหรับรับข้อมูลจากภายนอกซึ่งถูกส่งมาจากขา DTR

ขา 7 RTS : Request To Send เป็นขาสำหรับส่งสัญญาณร้องขอให้ทางอุปกรณ์ปลายทางส่งข้อมูลกลับมายังคอมพิวเตอร์ โดยขาที่รับสัญญาณ RTS ก็คือขา CTS ในกรณีที่ใช้การเชื่อมต่อแบบ Null Modem 3 สาย จะต้องเชื่อมต่อขา RTS และ CTS ของตัวมันเองเข้าด้วยกัน เพื่อจะให้การับและส่งข้อมูลสามารถเกิดขึ้นได้ตลอดเวลา

ขา 8 CTS : Clear To Send ขารับสัญญาณจากขา RTS เมื่อรับสัญญาณได้ ข้อมูลที่ขา TXD จะถูกส่งออกไป ดังนั้นขานี้จึงถูกใช้เพื่อตรวจสอบอุปกรณ์ต่อพ่วงว่าพร้อมที่จะรับข้อมูลหรือไม่

ขา 9 RI : Ring Indicator ใช้แสดงสถานะสัญญาณเรียกจากสายโทรศัพท์ ในการสื่อสารโดยทั่วไปสายนี้จะไม่ถูกใช้งาน จะใช้งานก็ต่อเมื่อมีการเชื่อมต่อกับโมเด็มและโปรแกรมมีการตรวจสอบสัญญาณนี้เท่านั้น

ตารางที่ 2.9 การจัดขาของคอนเนคเตอร์พอร์ตอนุกรมตามมาตรฐาน RS-232 แบบ DB-9 และDB-25

DB-9	DB-25	ชื่อของสายสัญญาณ	ชนิดของสายสัญญาณ
1	8	DCD : Carrier Detect	อินพุต
2	3	RXD : Receive Data	อินพุต
3	2	TXD : Transmited Data	เอาต์พุต
4	20	DTR : Data Terminal Ready	เอาต์พุต
5	7	GND : Signal Ground	-
6	6	DSR : Data Set Ready	อินพุต
7	4	RTS : Request To Send	เอาต์พุต
8	5	CTS : Clear To Send	อินพุต
9	22	RI : Ring Indicator	อินพุต

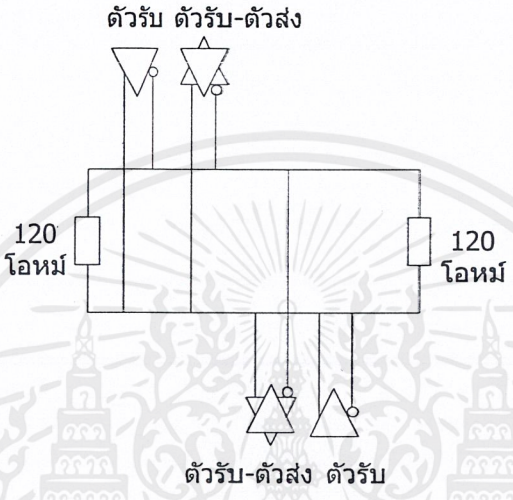
2.21 การสื่อสารข้อมูลแบบอนุกรมด้วย RS-485

2.21.1 มาตรฐานการสื่อสารข้อมูล

สำหรับมาตรฐาน RS-485 นั้นเป็นมาตรฐานที่อาศัยหลักการทำงานของการส่งสัญญาณแบบดิฟเฟอเรนเชียล (Differential) เช่นเดียวกับมาตรฐาน RS-422A แต่จะสามารถสื่อสารข้อมูลได้ 2 ทิศทาง ในสายนำสัญญาณเพียงคู่เดียวซึ่งเป็นการสื่อสารข้อมูลแบบฮาล์ฟดูเพล็กซ์ (Half Duplex)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากผลของการใช้สัญญาณในลักษณะดิฟเฟอเรลเชียล จะทำให้ระยะทางและความเร็วในการสื่อสารข้อมูลสูง เช่นเดียวกับมาตรฐาน RS-422A แต่มาตรฐาน RS-485 นั้นสามารถที่จะสื่อสารระหว่างอุปกรณ์ทั้งการรับและการส่งได้สูงสุดถึง 32 ตัว หรือ การสื่อสารข้อมูลตามมาตรฐาน RS-485 นั้นเป็นการสื่อสารแบบหลายจุด (Multi-point Communication)



รูปที่ 2.30 โครงสร้างของการสื่อสารข้อมูลแบบอนุกรมด้วย RS-485

ตารางที่ 2.10 การเปรียบเทียบมาตรฐานการสื่อสารข้อมูลของ EIA

พารามิเตอร์	RS-232	RS-423	RS-422A	RS-485
โหมดการทำงาน	Single-ended	Single-ended	Differential	Differential
จำนวนของตัวรับและตัวส่งที่ยอมรับได้	1 ตัวส่ง 1 ตัวรับ	1 ตัวส่ง 10 ตัวรับ	1 ตัวส่ง 10 ตัวรับ	32 ตัวส่ง 32 ตัวรับ
จำนวนของตัวรับและตัวส่งที่ยอมรับได้	1 ตัวส่ง 1 ตัวรับ	1 ตัวส่ง 10 ตัวรับ	1 ตัวส่ง 10 ตัวรับ	32 ตัวส่ง 32 ตัวรับ
ความยาวของคู่สายสัญญาณรับส่งข้อมูล	50 ฟุต	4000 ฟุต	4000 ฟุต	4000 ฟุต
แรงดันไฟฟ้าโหมดร่วมสูงสุด	$\pm$2.5 V	$\pm$6 V	+6 V - 2.5 V	12 V -7 V
Driver Load (W)	3 k ถึง 7 k	450 ต่ำสุด	100 ต่ำสุด	60 ต่ำสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.10 (ต่อ) การเปรียบเทียบมาตรฐานการสื่อสารข้อมูลของ EIA

พารามิเตอร์	RS-232	RS-423	RS-422A	RS-485
โหมดการทำงาน	Single-ended	Single-ended	Differential	Differential
Driver slew rate	30 V / us สูงสุด		NA	NA
กระแสลิมิตเมื่อเอาต์พุตลัดวงจร	500 mA ลัดวงจรกับ Vcc หรือ Gnd	150 mA ลัดวงจรกับ Gnd	150 mA ลัดวงจรกับ Gnd	150 mA ลัดวงจรกับGnd 250 mA ลัดวงจรกับ 8 V หรือ 12 V
ค่าความต้านทานเอาต์พุตของตัวส่ง (W)	NA-Power On 300- Power Off	NA-Power On 60 K- Power Off	NA-Power On 60 K – Power Off	60 K Power On ,Off
ค่าความต้านทานอินพุตของตัวรับ (W)	3 k ถึง 7 k	4 k	4 k	4 k
ความไวตัวรับ	±3 V	±200 mV	±200 mV	±200 mV

ตารางที่ 2.11 ข้อเปรียบเทียบระหว่างการส่งข้อมูลแบบขนานและแบบอนุกรม

รูปแบบการสื่อสารข้อมูล	แบบขนาน	แบบอนุกรม
1. ระยะทาง	ปกติจะน้อยกว่า 100 ฟุต	ส่งได้ตั้งแต่ระยะทางสั้นๆ ไปจนถึงระยะทางเป็นไมล์
2. ความเร็ว	อัตราความเร็วสูงมากในระยะที่ไม่ไกลมากนักกำหนดได้เป็นจำนวนบิตต่อวินาที	อัตราความเร็วของข้อมูลที่ใช้กันอยู่ทั่วไปอยู่ในช่วง 0 ถึง 2 ล้านบิตต่อวินาที
3. ระดับของสัญญาณ	ในการอินเตอร์เฟสจะใช้ระดับของสัญญาณที่ใช้กับอุปกรณ์ (TTL) คือ สัญญาณลอจิก 1 และ 0 จะแทนด้วยระดับแรงดัน 5 V และ 0 V ตามลำดับ	ใช้มาตรฐานของ EIA RS-232 คือมีระดับสัญญาณไฟฟ้า ขนาด 12 V หรืออาจจะใช้มาตรฐาน 20 mA current loop หรืออาจจะใช้ระดับสัญญาณ TTL ก็ได้ (ใช้กันน้อยมาก)
4. ความผิดพลาดของสัญญาณ	ถ้าส่งในระยะทางที่ไกลความผิดพลาดของข้อมูลจะเกิดขึ้นง่าย	การผิดพลาดของสัญญาณนั้นจะมีน้อยลง
5. ค่าใช้จ่าย	จะสิ้นเปลืองค่าใช้จ่ายมาก	สิ้นเปลืองน้อยกว่าหลายเท่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.21.2 รูปแบบของการสื่อสารข้อมูลแบบ RS-485

รูปแบบของ RS-485 ในลักษณะเครือข่าย สามารถแบ่งได้เป็น 2 รูปแบบ คือ

1) เครือข่ายแบบที่ใช้สายนำสัญญาณสองเส้น (Two Wires)

การสื่อสารข้อมูล RS-485 แบบสองสายนี้ ถือเป็นการสื่อสารแบบซิมเพล็กซ์ (Simplex) คือ ลักษณะของการรับ-ส่ง ข้อมูล จะเป็นไปในลักษณะผลัดกันรับ-ส่ง โดยการกำหนดว่าจะให้เป็นการรับหรือการส่ง จะถูกกำหนดโดยตัวแม่ข่าย (Master) ข้อดีของเครือข่ายแบบนี้ก็คือ จะช่วยประหยัดสายสัญญาณที่ใช้ในการวางเครือข่าย ส่วนข้อเสีย คือความเร็วในการรับส่งจะช้าลง

2) เครือข่ายแบบที่ใช้สายนำสัญญาณสี่เส้น (Four Wires)

การสื่อสารข้อมูลแบบ RS-485 แบบที่ใช้สายนำสัญญาณสี่เส้นนี้ ถือเป็นการสื่อสารข้อมูลแบบฟลูดูเพล็กซ์ (Full Duplex) ลักษณะของการรับ-ส่งข้อมูลสามารถทำการรับและส่งข้อมูลได้ในเวลาเดียวกัน ทั้งนี้เพราะว่ามีบัสข้อมูล (Data Bus) อยู่จำนวน 4 เส้น ข้อดีของเครือข่ายแบบนี้ คือ ความเร็วของการรับและการส่งมากกว่าแบบที่ใช้สายนำสัญญาณสองเส้น 2 ชุด โดยที่โหลด 1 ชุด จะประกอบด้วยตัวส่ง 1 ตัวและตัวรับ 1 ตัว และค่าความต้านทานที่ต่อคร่อมระหว่างสายนำสัญญาณมีค่า 60 โอห์ม

เอาต์พุตของตัวส่งในสภาวะออฟ (Off) มีกระแสรั่วไหลไม่เกิน 100 ไมโครแอมป์ ในช่วงแรงดันไฟฟ้าระหว่าง –7 โวลต์ ถึง 12 โวลต์

เอาต์พุตของตัวส่งให้แรงดันไฟฟ้าเอาต์พุต 1.5 โวลต์ ถึง 5 โวลต์ในช่วงแรงดันไฟฟ้าระหว่าง –7 โวลต์ ถึง 12 โวลต์ ตัวส่งจะมีวงจรป้องกันตัวเองที่ส่วนเอาต์พุต ในกรณีที่ตัวส่งหลายๆ ตัวส่งข้อมูลออกมาพร้อมๆ กัน

2.21.3 มาตรฐานของ RS-485

1) คุณสมบัติเฉพาะตัวของตัวรับ RS-485

1.1) ค่าความต้านทานอินพุตมีค่าสูง โดยมีค่าไม่น้อยกว่า 12 กิโลโอห์ม

1.2) มีค่าแรงดันอินพุต ระหว่าง –7 โวลต์ ถึง 12 โวลต์

1.3) ตอบสนองต่อสัญญาณที่แตกต่างจากสัญญาณโหมดรวมได้ $\pm$ 200 มิลลิโวลต์

2) คุณสมบัติเฉพาะของคู่สัญญาณ RS-232

คู่สายสัญญาณที่ใช้ในการรับ-ส่งสัญญาณ ในระบบ RS-485 นั้นควรพันสลับเป็นเกลียวเพื่อการลดทอนสัญญาณรบกวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ความหมายของยูนิตโหลด (Unit Load : U. L.)

ยูนิตโหลด เป็นจำนวนมากที่สุดของตัวรับและตัวส่ง ที่สามารถใช้งานบนคู่สายสัญญาณรับ-ส่ง คู่หนึ่ง โดยจะขึ้นอยู่กับยูนิตโหลด ซึ่ง RS-485 ยอมรับได้ที่ 32 ยูนิตโหลด ต่อคู่สายสัญญาณ 1 คู่สาย

นิยามของยูนิตโหลด เป็นโหลดที่ใช้กระแส 1 มิลลิแอมป์ ที่แรงดันโหมดร่วม 12 โวลต์ ซึ่งโหลดนี้ประกอบด้วยตัวส่ง และตัวรับ แต่ไม่รวมค่าความต้านทานที่ต่อคร่อมคู่สายสัญญาณรับส่ง

4) คุณสมบัติเฉพาะของตัวรับส่ง

4.1) เป็นไปตามมาตรฐาน RS-485 RS-422 CCITT V.11 และ X.27

4.2) เอาต์พุตของตัวส่งเป็นแบบ 3 สเตท (State) ยกเว้น SN75179B

4.3) เอาต์พุตตัวส่งสามารถขับกระแสได้สูงสุด 60 มิลลิแอมป์

4.4) ค่าความต้านทานอินพุตของตัวรับมีค่าประมาณ 20 กิโลโอห์ม (น้อยที่สุด)

4.5) ตัวรับมีค่าความไวอินพุต (Input Sensitivity) ประมาณ $\pm$ 200 มิลลิโวลต์

4.6) ใช้ไฟเลี้ยง 5 โวลต์

2.21.4 การคำนวณหาจำนวนคู่รับ – ส่ง RS-485

1) ตัวส่ง

ค่าของกระแสรั่วไหล เมื่ออยู่ในสภาวะออฟที่ 12 V มีค่าไม่เกิน 0.1 mA

ตัวส่งมีค่า U. L. = 0.1 mA / 1.0 mA = 0.1 U. L.

2) ตัวรับ

ค่าของกระแสอินพุตที่แรงดันไฟฟ้าอินพุต 12 V มีค่าไม่เกิน 0.6 mA

ตัวรับมีค่า U. L. = 0.6 mA / 1.0 mA = 0.6 U. L.

เมื่อพิจารณาโหลด 1 (ตัวรับและตัวส่ง) มีค่า = (0.1 + 0.6) mA / 1.0 mA

= 0.7 U. L.

สายสัญญาณรับ-ส่ง ข้อมูลคู่หนึ่งสามารถรองรับตัวรับ – ส่งได้ = 32 / 0.7 = 45 คู่

2.21.5 RS-485 Protocol

RS-485 Protocol เป็นเงื่อนไขของการรับและส่งข้อมูลที่ได้พัฒนาขึ้นมาเพื่อใช้กับระบบการสื่อสารข้อมูลแบบ RS-485 โดยได้ประยุกต์มาจาก DDCMP Protocol BSC Protocol HDLC Protocol และ Xebee Protocol เพื่อสร้างชุดของข้อความในการรับ–ส่งข้อมูล และการประยุกต์

เฟรมของข้อมูลประกอบด้วย HEADER PORTION, และ DATA PORTION

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.12 เฟรมของข้อมูล HEADER PORTION และ DATA PORT

SYN	SYN	SYN	HEADER	DATA PORTION	SYN	SYN	SYN

HEADER PORTION

ในส่วนของ HEADER PORTION ของ RS-485 Protocol ได้ประยุกต์มาจากบางส่วนของ DDCMP (Digital Data Communication Message Protocol)

HDLC (High level Data Link Control)

Xebee Protocol

ตารางที่ 2.13 รายละเอียดของ HEADER PORTION

SOH	TYD	EXC	CDAT	FSEQTF	FSERR	ADDRESS	LRC#1
8 Bit	2 Bit	4Bit	10it	8 Bit	8 Bit	24 Bit	8 Bit

SOH : Start of Header (8 Bit)

เป็นส่วนแสดงการเริ่มต้นของข้อความของ HEADER PROTION มีขนาด 8 บิต

TYD : Type of Data (2 Bit)

เป็นส่วนที่ใช้แสดงถึงชนิดของข้อมูลที่บรรจุอยู่ในส่วนของ DATA PORTION

ตารางที่ 2.14 ชนิดของข้อมูลที่บรรจุอยู่ในส่วนของ DATA PORTION

TPD	ลักษณะของข้อความ
00	ข้อความเป็นข้อมูล
01	ข้อความเป็นคำสั่ง
TPD	ลักษณะของข้อความ
10	ข้อความเป็นคำสั่งและพารามิเตอร์
11	ข้อความเป็นข้อความตอบรับจากตัวตอบรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DCAT : Count of Data (10 Bit)

เป็นส่วนแสดงจำนวนข้อมูลที่บรรจุอยู่ใน DATA FIELD ของ DATA PORTION โดยนับข้อมูลแบบไบนารี (Binary)ได้สูงสุด 1 กิโลไบต์ / 1 เฟรม (สามารถขยายได้ 81 กิโลไบต์ / 1 เฟรม)

EXC : Expansion of Count (4 Bit)

เป็นส่วนแสดงบิตขยายของ CDAT มีขนาด 4 บิต

บิตแรก = 0 แสดงว่า CDAT มีขนาด 10 บิต

บิตแรก = 1 แสดงว่า CDAT มีขนาด 13 บิต (เพิ่มจาก 3 บิต ของ EXC)

FSEQTF : File Sepuence Transfer (8 Bit)

เป็นส่วนแสดงการรับ – ส่งข้อมูลอย่างต่อเนื่องครั้งละหลายๆ เฟรม มีขนาด 8 บิต ซึ่งจะประกอบด้วย

ตารางที่ 2.15 การรับ – ส่งข้อมูลอย่างต่อเนื่องของ File Sepuence Transfer

FSQ	NFS	P / F	Nfr
1 Bit	3 Bit	1 Bit	3 Bit

FSQ : Frame Sequence Bit (1 Bit)

0 คือ ส่งข้อความทีละเฟรม ไม่ต้องพิจารณาอีก บิตของ FSEQTF และ Fserr

1 คือ ส่งข้อความทีละหลายๆ เฟรม พิจารณาอีก 7 บิตของ FSEQTF และ Fserr (โดยส่งต่อเนื่องไม่เกิน 8 เฟรมต่อครั้ง)

NFs : Number of Frame Sequence Begin to Send (3 Bit)

แสดงลำดับเฟรมข้อความที่จะส่ง (0-7) มีขนาด 3 บิต

NFr : Number of The Next Frame be Receive (3 Bit)

แสดงลำดับเฟรมข้อความต่อไปที่จะรับ (0-7) มีขนาด 3 บิต

P / F : Poll / Final Bit (1 Bit)

กรณีเซนเตอร์ (Center) ส่งข้อความให้เทอร์มินอล บิต P / F เป็น P : Poll

P = 1 ยินยอมให้เทอร์มินอล (Terminal) สามารถตอบรับได้

P = 0 ยังไม่ยินยอมให้เทอร์มินอลสามารถตอบรับได้

กรณีเซนเตอร์ส่งข้อความให้เทอร์มินอล บิต P / F เป็น F : Final

P = 1 ยินยอมให้เซนเตอร์สามารถตอบรับได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

P = 2 ยังไม่ยินยอมให้เซนเตอร์ตอบรับได้

Fserr : Frame Sequence error

แสดงข้อความที่ผิดพลาด มีขนาด 8 บิต Fserr ประกอบด้วย

ตารางที่ 2.16 ข้อความที่ผิดพลาดของ Fserr

F7	F6	F5	F4	F3	F2	F1	F0
1 Bit	1 Bit	1 Bit	1 Bit	1 Bit	1 Bit	1 Bit	1 Bit

LRC # 1 : Longitudinal Redundacy Chec

เป็นส่วนแสดงความผิดพลาดและเป็นรหัสตรวจสอบความผิดพลาด ความผิดพลาดของ HEADER มีขนาด 8 บิต

DATA PORTION

ในส่วนของ DATA PORTION ของ RS-485 Protocol ได้ประยุกต์มาจากบางส่วนของDDCMP (Digital Data Communication Message Protocol) HDLC (High level Data Link Control)และ Xebee Protocol

ตารางที่ 2.17 DATA PORTION ของ RS-485 Protocol

STX	DATA FIELD	ETX	LRC # 2
8 Bit	1-8 Kbyte	8 Bit	8 Bit

STX : Start of Text (8 Bit)

เป็นส่วนแสดงการเริ่มต้นของข้อความส่วน DATA PORTION มีขนาด 8 บิต

ETX : End of Text (8 Bit)

เป็นส่วนแสดงการสิ้นสุดข้อความ ส่วน DATA PORTION มีขนาด 8 บิต

LRC # 2 : Longitudinal Redundacy Check (8 Bit)

เป็นส่วนแสดงรหัสตรวจสอบของ DATA PORTION มีขนาด 8 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DATA FIELD (1-8 Kbyte)

เป็นส่วนแสดงข้อมูลที่ใช้ในการติดต่อ มีขนาดไม่เกิน 1024 ตัวอักษรต่อเฟรมข้อความ (ขยายได้ 8192 ตัวอักษร)

ตารางที่ 2.18 ข้อมูลที่ใช้ในการติดต่อของ DATA FIELD

COMMAND	SEP	SDB	DATA	EDB
8 Bit	8 Bit	8 Bit	ASCII CODE	8 Bit

COMMAND (8 Bit)

คือ ชุดคำสั่งของ RS-485 Protocol

SEP : Seperator (8 Bit)

เป็นส่วนแสดงการเริ่มต้นของข้อมูลหรือพารามิเตอร์

SDB : Start of DATA Block (8 Bit)

เป็นส่วนแสดงการเริ่มต้นของข้อมูลหรือพารามิเตอร์

DATA : Data or Parameter (ASCII CODE)

เป็นข้อมูล หรือ รูปแบบเป็นรหัส ASCII

EDB : End of DATA Block (8 Bit)

เป็นส่วนแสดงการสิ้นสุดของข้อมูลหรือพารามิเตอร์

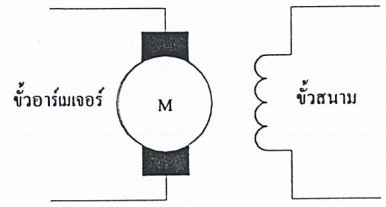
2.22 มอเตอร์ไฟฟ้ากระแสตรง

มอเตอร์ คือ อุปกรณ์ซึ่งทำหน้าที่เปลี่ยนพลังงานไฟฟ้าเป็นพลังงานกล และมอเตอร์ซึ่งขับเคลื่อนด้วยกระแสไฟฟ้าดีซีนี้เรียกว่า มอเตอร์ดีซี (DC Motor) นอกจากนั้นยังมีมอเตอร์ขนาดเล็กซึ่งขับเคลื่อนด้วยกระแสไฟฟ้าดีซี ทิศทางการหมุนของมอเตอร์ดีซีจะตรงข้ามกับกรณีเครื่องกำเนิดไฟฟ้าดีซีโดยสิ้นเชิง แต่สำหรับโครงสร้างแล้วจะเหมือนกันทุกประการจึงสามารถนำเครื่องกำเนิดไฟฟ้าดีซีมาใช้ทำหน้าที่ของมอเตอร์ดีซีได้

มอเตอร์ไฟฟ้ากระแสตรงสามารถจำแนกออกไปได้อีกหลายประเภท ซึ่งขึ้นอยู่กับวิธีการสร้างที่รู้จักกันเป็นส่วนใหญ่ในปัจจุบัน คือมอเตอร์ไฟฟ้ากระแสตรงแบบขนาน (Shunt) แบบอนุกรม (Series) แบบผสม (Compound) และแบบแม่เหล็กถาวร (PM Motor or Permanent Magnet Motor)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

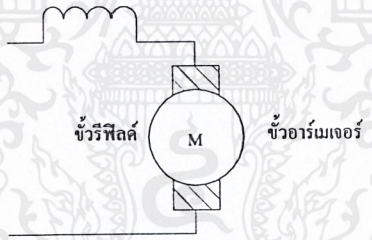
2.22.1 ลักษณะโครงสร้างของมอเตอร์แบบขนาน



รูปที่ 2.31 ลักษณะโครงสร้างของมอเตอร์แบบขนาน

มอเตอร์แบบนี้สามารถปรับเส้นแรงแม่เหล็กได้อย่างอิสระต่อกระแสของอาร์เมเจอร์ (Armature) เป็นผลให้สามารถควบคุมพารามิเตอร์ (Parameter) ให้มีค่าคงที่ได้ตลอดช่วงพิสัยที่กว้าง มอเตอร์ชนิดนี้จึงมักใช้ในงานระบบควบคุมการเคลื่อนที่ที่ต้องการแรงบิด

2.22.2 ลักษณะโครงสร้างของมอเตอร์แบบอนุกรม



รูปที่ 2.32 ลักษณะโครงสร้างของมอเตอร์แบบอนุกรม

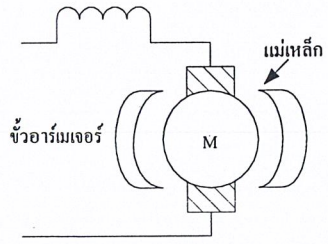
มีเส้นแรงแม่เหล็กเป็นสัดส่วนกับกระแสดังนั้นเส้นแรงของสนามแม่เหล็กจึงสามารถปรับค่าได้ ซึ่งเราจะได้ความสัมพันธ์ ระหว่างความเร็วและแรงบิดไม่เป็นเชิงเส้น จึงเหมาะที่จะนำไปใช้งานในภาวะจำเพาะ คือ เมื่อต้องการแรงบิดสูงที่ความเร็วต่ำ และแรงบิดต่ำที่ความเร็วสูง เช่น ระบบการขับเคลื่อนของรถลาก

2.22.3 ลักษณะโครงสร้างของมอเตอร์แม่เหล็กแบบถาวร

มอเตอร์แบบนี้จะใช้การกระตุ้นฟิลด์ (Field) ของมอเตอร์เป็นแม่เหล็กถาวร ใช้ขดลวดแบนจะใช้เส้นแรงของฟิลด์มีค่าคงที่ อัตราส่วนระหว่างกระแสอาร์เมเจอร์และแรงบิดจะมีค่าคงที่ด้วย ซึ่งมีข้อดี คือ ไม่มีกำลังสูญเสียในฟิลด์ มีประสิทธิภาพสูงกว่าและมีขนาดเล็กกว่าเมื่อเทียบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

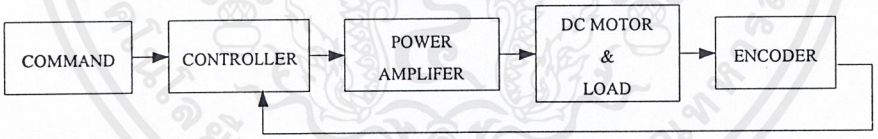
กับมอเตอร์แบบใช้ขดลวดในการกระตุ้นที่มีขนาดกำลังม้าเท่ากันจึงเหมาะกับงานที่ต้องการแรงบิดของโหลดสูง



รูปที่ 2.33 ลักษณะโครงสร้างของมอเตอร์แม่เหล็กแบบถาวร

มอเตอร์แบบนี้จะใช้การกระตุ้นฟิลด์ (Field) ของมอเตอร์เป็นแม่เหล็กถาวร ใช้ขดลวดแบนจะใช้เส้นแรงของฟิลด์มีค่าคงที่ อัตราส่วนระหว่างกระแสอาร์เมเจอร์และแรงบิดจะมีค่าคงที่ด้วย ซึ่งมีข้อดี คือ ไม่มีกำลังสูญเสียในฟิลด์ มีประสิทธิภาพสูงกว่าและมีขนาดเล็กกว่าเมื่อเทียบกับมอเตอร์แบบใช้ขดลวดในการกระตุ้นที่มีขนาดกำลังม้าเท่ากันจึงเหมาะกับงานที่ต้องการแรงบิดของโหลดสูง

2.22.4 ระบบการควบคุมมอเตอร์ไฟฟ้ากระแสตรงแบบพื้นฐาน



รูปที่ 2.34 ระบบการควบคุมมอเตอร์ไฟฟ้ากระแสตรงแบบพื้นฐาน

Controller ตัวควบคุมเป็นส่วนของระบบที่ทำให้เกิดสัญญาณควบคุมไปบังคับมอเตอร์ไฟฟ้ากระแสตรงและโหลดซึ่งอาจเป็นแอนะลอกหรือดิจิตอลก็ได้

POWER AMPLIFIER หรือส่วนขับทำหน้าที่ปรับปรุง และขยายสัญญาณให้เหมาะสมก่อนจะป้อนไปขับมอเตอร์ไฟฟ้ากระแสตรง ซึ่งอาจแบ่งเป็นลิเนียร์ (Linear) เพาเวอร์แอมปลิไฟล์ (Power Amplifier) และพัลซ์วิดท์มอดูเลชั่น (Pulse Width Modulation)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลิเนียร์เพาเวอร์แอมปลิไฟล์ เป็นการควบคุมมอเตอร์แบบต่อเนื่อง แต่จะมีความสูญเสียทางกำลังสูงเนื่องจากกำลังงานส่วนใหญ่จะสูญเสียในเอาต์พุตทรานซิสเตอร์ (Transistor) เป็นจำนวนมาก เพราะขณะมอเตอร์ไม่ทำงานทรานซิสเตอร์ส่วนนี้ก็ต้องแบกภาระเนื่องจากมีกระแสไหลผ่าน

พัลซ์วิดท์มอดูเลชั่น เป็นสวิตชิ่งแอมปลิไฟล์เออร์ (Switching Amplifier) คือ ควบคุมโวลต์เตจ (Voltage) ของมอเตอร์ โดยการปรับดิวตี้ไซเกิล (Duty Cycle) ของโวลต์เตจที่จ่ายให้กับมอเตอร์และให้มันทำงานทุกๆ ภาวะอิ่มตัว (On) หรือภาวะไม่นำกระแส (Off) ด้วยเหตุนี้กำลังสูญเสียน้อย เนื่องจากเมื่อทรานซิสเตอร์นำกระแสแรงดันตกคร่อมตัวมันจะน้อยจนตัดทิ้งได้ และเมื่อหยุดนำกระแส แรงดันตกคร่อมจะประมาณแหล่งจ่ายนั้น กระแสไหลผ่านจึงน้อยมากประมาณศูนย์ แต่จะใช้กับความถี่สูงได้ไม่ดี และความถี่ต้องคงที่ ถ้าไม่เช่นนั้นอาจเกิดออสซิลเลตขึ้นได้

DC MORTOR LOAD ระบบคอนโทรล (Control) ส่วนที่ออกแรงทำงานจะเป็นเครื่องจักรกล

ENCODER หรือฟีดแบ็คทรานสดิวเซอร์ (Feed Back Transducer) ใช้รับรู้หรือดีเทค (Detect) สัญญาณที่ต้องการโดยไม่มีผลของการโหลดดิ้ง (Loading) สัญญาณที่ดีเทคจะป้อนกลับไปเปรียบเทียบสัญญาณอ้างอิงเพื่อควบคุมมอเตอร์ไฟฟ้ากระแสตรงอีกที ซึ่งฟีดแบ็คทรานสดิวเซอร์ แบ่งเป็นแบบแอนะลอกและดิจิตอล

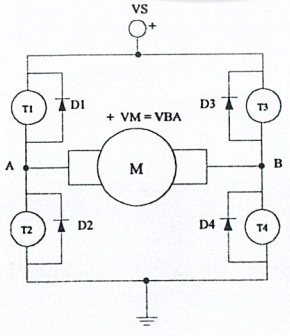
2.22.5 การทำงานของแอมปลิไฟล์แบบพัลซ์วิดท์มอดูเลชั่น

การขยายแบบพัลซ์วิดท์มอดูเลชั่น (Pulse Width Modulation : PWM) สามารถแบ่งได้เป็น 3 ชนิด ตามลักษณะการทำงาน คือ

1) ไบโพลาร์ (Bipolar)

2) ยูนิโพลาร์ (Unipolar)

3) ลิมิตยูนิโพลาร์ (Limit Unipolar)



รูปที่ 2.35 วงจรพื้นฐานของการขยายพัลซ์วิดท์มอดูเลชั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

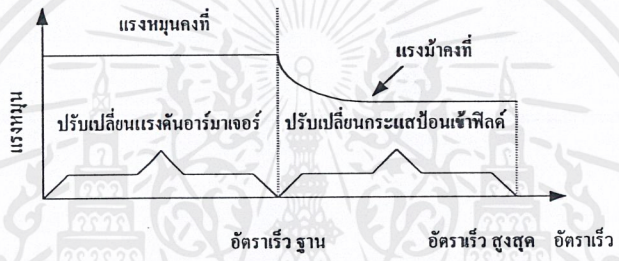
ซึ่งทั้ง 3 ชนิดสามารถอธิบายด้วยวงจรพื้นฐานนี้ได้โดยทั้ง 3 ชนิด ต่างกันตรงการควบคุมเปิด ปิด ของทรานซิสเตอร์ ซึ่งแบบไบโพลาร์ เพราะเป็นแบบที่ควบคุมและเข้าใจได้ง่าย คือ

เมื่อมอเตอร์อยู่ในเฟส On ก็ให้ T1 กับ T4 On และT2 กับT3 Off ดังนั้นกระแสไหลจาก VS ผ่าน T1 มอเตอร์และ T4 ลงกราวด์ ดังนั้น Vm = Vs (มอเตอร์หมุนตามเข็ม)

เมื่อให้มันมอเตอร์อยู่ในเฟส Off T2 กับT3 On และ T1 กับ T4 Off ดังนั้นกระแสไหลจาก VS ผ่าน T3 ขั้วลบ มอเตอร์ T2 และลงกราวด์ ดังนั้น Va = Vs (มอเตอร์หมุนทวนเข็ม)

2.22.6 การควบคุมความเร็วของมอเตอร์ไฟฟ้ากระแสตรง

1) การควบคุมแรงดันไฟตรงของอาร์เมเจอร์ (Armature Voltage Control)



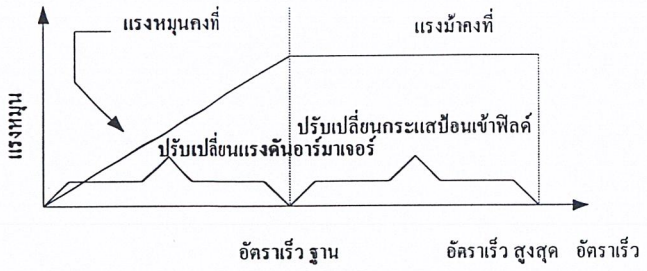
รูปที่ 2.36 การควบคุมแรงดันไฟตรงของอาร์เมเจอร์

ความเร็วมอเตอร์ไฟฟ้ากระแสตรง จะแปรผันตรงกับแรงดันที่ใส่ให้กับขดลวดอาร์เมเจอร์ ดังนั้นเราจึงสามารถควบคุมความเร็วของมอเตอร์ โดยการควบคุมแรงดันของอาร์เมเจอร์ วิธีการนี้จะใช้ในช่วงความเร็วที่ต่ำกว่าความเร็วที่กำหนด (Base Speed) หรือ N- Base

2) การควบคุมความเข้มของสนามแม่เหล็ก (Field Control)

การควบคุมความเร็วของมอเตอร์ไฟฟ้ากระแสตรง ในย่านความเร็วสูงกว่าความเร็วที่กำหนด จะทำได้โดยการควบคุมกระแสของขดลวดสร้างสนามแม่เหล็กของมอเตอร์ เมื่อเราต้องการเพิ่มความเร็วเราจะต้องลดขนาดของกระแสขดลวดลง การลดความเข้มของสนามแม่เหล็กของมอเตอร์ จะมีผลทำให้แรงบิดสูงสุดของมอเตอร์ลดลง ขณะที่กำลังออกสูงสุดของมอเตอร์จะไม่เปลี่ยนแปลง วิธีนี้จะใช้กับภาระที่ต้องการความเร็วสูงโดยที่แรงบิดของภาระ จะต้องลดลงเมื่อความเร็วเพิ่มขึ้น ไม่เช่นนั้นจะเป็นการโหลดเกิน (Over Load Motor)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.37 การควบคุมความเข้มของสนามแม่เหล็ก

การควบคุมความเร็วของมอเตอร์ไฟฟ้ากระแสตรง ในย่านความเร็วสูงกว่าความเร็วที่กำหนด จะทำได้โดยการควบคุมกระแสของขดลวดสร้างสนามแม่เหล็กของมอเตอร์ เมื่อเราต้องการเพิ่มความเร็วเราจะต้องลดขนาดของกระแสขดลวดลง การลดความเข้มของสนามแม่เหล็กของมอเตอร์ จะมีผลทำให้แรงบิดสูงสุดของมอเตอร์ลดลง ขณะที่กำลังออกสูงสุดของมอเตอร์จะไม่เปลี่ยนแปลง วิธีนี้จะใช้กับภาระที่ต้องการความเร็วสูงโดยที่แรงบิดของภาระ จะต้องลดลงเมื่อความเร็วเพิ่มขึ้น ไม่เช่นนั้นจะเป็นการโหลดเกิน (Over Load Motor)

2.23 สเตปเปอร์มอเตอร์ (Step Moter)

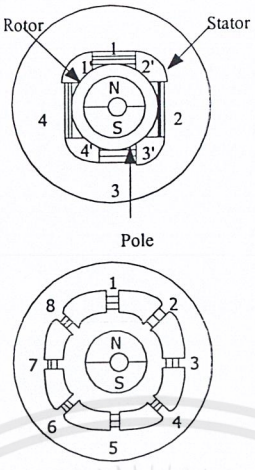
2.23.1 ความรู้เบื้องต้น และหลักการทำงาน

เมื่อเราป้อนไฟฟ้าให้กับมอเตอร์ทำให้หมุนเพียงเล็กน้อยตามเส้นรอบวงและหยุด ซึ่งต่างจากมอเตอร์ ทั่วไปที่จะหมุนทันทีและตลอดเวลาเมื่อป้อนแรงดันไฟฟ้าข้อดีของสเตปมอเตอร์สามารถกำหนด ตำแหน่งของการหมุนด้วยตัวเลข (องศาหรือระยะทาง) ได้อย่างละเอียดโดยใช้คอมพิวเตอร์หรือ ไมโครคอนโทรลเลอร์เป็น เครื่องกำหนดและจัดเก็บตัวเลข

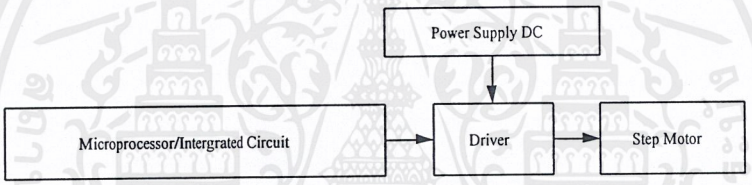
1) โครงสร้างของสเตปเปอร์มอเตอร์

โครงสร้างของขั้วแม่เหล็กบนสเตเตอร์ (Stator) ทำมาจากแผ่นเหล็กวงแหวนที่มีซี่ยื่นออกมาประกอบกันเป็นชั้นๆ โดยที่แต่ละซี่นั้นจะมีคอยล์ (Coil) ขดลวดพันสวมอยู่ เมื่อมีการป้อนกระแสผ่านคอยล์ทำให้เกิดสนามแม่เหล็กไฟฟ้า (Electromagnetic)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.38 โครงสร้างของสเตปเปอร์มอเตอร์



รูปที่ 2.39 การควบคุมระบบสเตปเปอร์มอเตอร์

ถ้าเราเพิ่มจำนวนของขั้วแม่เหล็กมากขึ้นจะเพิ่มจำนวนของสเตป (Step) ต่อวงจรรอบมากขึ้นตามด้วย ลักษณะการนำไปใช้งานสเตปมอเตอร์ ใช้งานลักษณะระบบเปิด (Open Loop System) คือ สเตปมอเตอร์สามารถทำงานได้โดยไม่ต้องมีการป้อนค่าพารามิเตอร์กลับมา (Feed back) แต่ทุกวิธีที่ต้องการกำหนดตำแหน่งที่แน่นอนนั้น จะต้องการป้อนกลับไปยังระบบและตัวบอกตำแหน่งว่าถูกต้องหรือผิดผลาดให้รับทราบ

วิธีที่ใช้กับสเตปมอเตอร์ คือนำลิมิทสวิตช์ (Limit Switch) ติดตามตำแหน่งที่จะตรวจจับเมื่อสเตปมอเตอร์ เริ่มหมุนแล้วหมุนไปจนถึงตำแหน่งของสวิตช์ตรวจจับสัญญาณ สวิตช์ทำงานก็จะป้อนกลับไปสู่ระบบ ซึ่งก็จะทำให้รู้การทำงานของสเตปมอเตอร์ตลอดเวลา จึงทำให้ตัววงจรไมโครคอนโทรลเลอร์เองจะมีจุดอ้างอิง ไว้ให้เริ่มต้นการทำงานและอ้างอิงตำแหน่งได้ถูกต้อง

โดยแนวทางสเตปมอเตอร์เป็นอุปกรณ์จำพวกเชิงกลทางไฟฟ้า โดยมีกรุ๊ป (Group) ของไบนารี่โวลต์เตทเปอินพุตและการเคลื่อนที่ แบบเชิงมุมเป็นเอาต์พุต หรือว่าหมุนทีละสเตปซึ่งอยู่ระหว่าง 0.1 - 30 องศา อยู่ที่โครงสร้างของสเตปมอเตอร์ โดยตามสัญญาณพัลซ์ที่จ่ายให้กับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขดสเตเตอร์ทำให้เกิดแรงผลักแก่โรเตอร์ (Rotor) หมุนไป สเตปมอเตอร์มีขดลวดหลายชุดในที่นี้เราเรียกว่าเฟส (Phase) ดังนั้นสัญญาณที่ต่อเนื่องเป็นลำดับ (Sequence) ลักษณะของไบนารี่ (Binary) ซึ่งจะต้องไปยังวงจรขับ (Driver) ก็จะทำให้โรเตอร์หมุนไปอย่างต่อเนื่อง

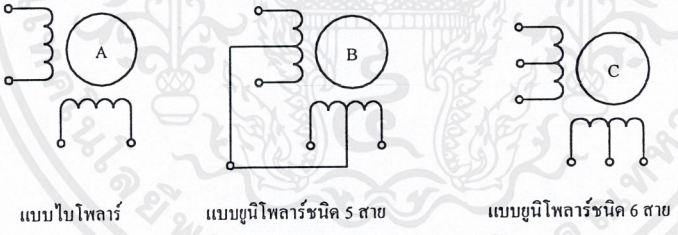
2) การพันขดลวดบนสเตเตอร์ของสเตปมอเตอร์

2.1) แบบไบโพล่าร์ (Bipolar)

จะมีการพันขดลวดหนึ่งขด (จำนวนรอบขึ้นอยู่กับสเปค (Space) ใช้งาน) ในแต่ขั้วแม่เหล็กของสเตเตอร์ โดยขั้วแม่เหล็กที่เกิดขึ้น ที่สเตเตอร์จะถูกกำหนดโดยทิศทางของการไหลของกระแสไฟฟ้า ซึ่งสามารถทำให้เกิดขั้วแม่เหล็กในทิศทางตรงกันข้ามได้เพียง การกลับทิศทางของการไหลในกระแสไฟฟ้า โดยมาจากการควบคุมของวงจรสวิทชิ่ง (Switching) ให้กลับขั้วไฟฟ้า

2.2) แบบยูนิโพล่าร์ (Unipolar)

แบบนี้มี 2 ขด บนแต่ละขั้วแม่เหล็กของสเตเตอร์ ทำให้แต่ละขดลวดเกิดขั้วแม่เหล็กในทิศทางตรงกันข้าม การกลับทิศทางขั้วแม่เหล็กทำได้โดยใช้วงจรสวิทชิ่งให้สลับหนึ่งไปยังอีกขั้วหนึ่งแบบยูนิโพล่าร์จะทำให้เกิดแรงบิดน้อยกว่าแบบไบโพล่าร์ สังเกตได้จากสายไฟที่ต่อมาจากตัวสเตปมอเตอร์ซึ่งแบบไบโพลาร์จะมี 4 สาย ส่วนเป็นแบบยูนิโพล่าร์จะมี 5 สายหรือ 6 สาย

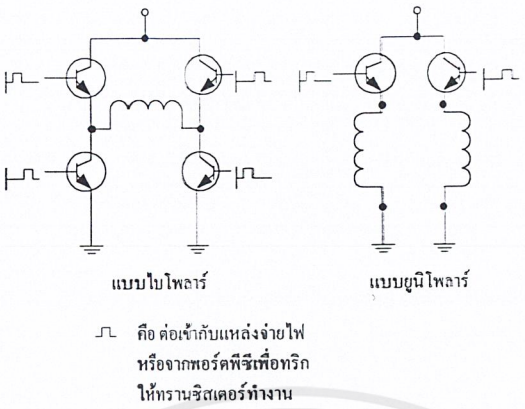


รูปที่ 2.40 การพันขดลวดบนสเตเตอร์ของสเตปมอเตอร์

3) การสั่งงานควบคุมการหมุนของสเตปมอเตอร์

การควบคุมและสั่งงานให้สเตปมอเตอร์ทำงาน ไปทีละสเตปสามารถทำได้โดยการจ่ายกำลังไฟไปยังขดลวด ในแต่ละขดบนสเตเตอร์ โดยการป้อนจะทำในลักษณะเป็นลำดับ หรือเรียกว่า ซีเควนเชียลในลูปที่ถูกต้อง ซึ่งจะแบบ ได้เป็น 3 รูปแบบ คือ แบบเวฟ (Wave) แบบ 2 เฟส (2 Phase) และแบบครึ่งสเตป (Half step) ทั้ง 3 แบบนี้ก็จะมีข้อดีและข้อเสียต่างกันออกไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.41 วงจรจ่ายไฟให้กับสเตปเปอร์มอเตอร์

3.1) แบบเวฟ (wave)

จะเป็นการกระตุ้นแบบที่ง่ายที่สุด ซึ่งจะทำการกระตุ้นขดลวดทีละขดในเวลาหนึ่งๆ เรียงกันไป ตัวอย่างเช่น ขดที่ 1, 2, 3, 4, 1, 2, 3, 4 เป็นลำดับอย่างนี้ หรือ ขด 1, 4, 3, 2, 1, 4, 3, 2 เป็นลำดับกันไป ทั้งนี้ขึ้นอยู่กับทิศทางที่เราต้องให้มอเตอร์หมุนไป วงจรควบคุมที่นำมากระตุ้นนั้นจะมีราคาค่อนข้างจะถูกกว่าและง่ายกว่า ขั้นตอนการทำงานมีดังนี้

ตารางที่ 2.19 ขั้นตอนการทำงานแบบเวฟ

สเตปที่	เฟส 1	เฟส 2	เฟส 3	เฟส 4
1	ทำงาน			
2		ทำงาน		
3			ทำงาน	
4				ทำงาน
5	ทำงาน			
6		ทำงาน		

3.2) แบบ 2 เฟส (2 Phase)

แบบนี้ก็จะคล้ายกับการกระตุ้นในแบบเวฟแต่จะต่างกันตรงที่ แบบ 2 เฟส จะกระตุ้นทีละ 2 ขด ที่อยู่ใกล้กันใน เวลาเดียวกัน และจะเรียงลำดับกันไป ดังเช่นแบบเดียวกับแบบเวฟ จะยกตัวอย่างการกระตุ้นขดลวดในลำดับดังนี้ 12, 23, 34.41, 12, 23, 34, 41 เรียงลำดับกันไปเรื่อยๆ หรือจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็น 14, 43, 32, 21, 14, 43, 32, 21 เรียงกันไปเรื่อยๆ ซึ่งข้อดีก็คือ การที่เราจะเพิ่มจำนวนขดลวดที่ถูกกระตุ้นจะทำให้แรงบิดได้มากกว่าแบบเวฟ ซึ่งโรเตอร์จะหมุนด้วยแรงดึงแบบเต็มๆ แรงจาก ทั้ง 2 ขดลวดที่กระตุ้นพร้อมกัน และข้อเสียก็คือจะกระตุ้นขดลวดนั้นต้องใช้กำลังไฟมากขึ้นเป็น 2 เท่าของแบบเวฟ เราสามารถเขียนลำดับการกระตุ้นของขดลวดแบบ 2 เฟส ได้ดังในตารางต่อไปนี้

ตารางที่ 2.20 ขั้นตอนการทำงานแบบ 2 เฟส

สเตปที่	เฟส 1	เฟส 2	เฟส 3	เฟส 4
1	ทำงาน	ทำงาน		
2		ทำงาน	ทำงาน	
3			ทำงาน	ทำงาน
4	ทำงาน			ทำงาน
5	ทำงาน	ทำงาน		
6		ทำงาน	ทำงาน	

3.3) แบบครึ่งสเตป

แบบนี้แบบรูปแบบผสมผสานของการกระตุ้นระหว่าง แบบเวฟ กับ แบบ 2 เฟส เพื่อให้จำนวนรอบของสเตปให้ มากขึ้นเป็น 2 เท่า ซึ่งในระบบนี้จะทำการกระตุ้นขดลวดเรียงกันไปเรื่อยๆ เป็นลำดับ ดังตัวอย่างต่อไปนี้ 1, 12, 2, 23, 3, 34, 4, 41, 1, 12, 2, 23, 3, 34, 4, 41, 1 เป็นลำดับอยู่อย่างนี้เรื่อยไป ถ้าจะกลับทิศทางการหมุนก็จะได้เป็น 1, 41, 4, 43, 3, 32, 2, 21, 1, 41, 4, 43, 3, 32, 2, 21, 1 เป็นลำดับซึ่งมีข้อดี คือ การกระตุ้นแบบนี้จะให้แรงบิดที่เพิ่มมากขึ้น เนื่องจากช่วง สเตปที่มีระยะสั้นลงอีกประการหนึ่งแต่ละ สเตปเกิดแรงดึงจากขดลวด 2 ขดที่ถูกกระตุ้นพร้อมกันเป็นผลให้ค่าตำแหน่งความถูกต้องมากขึ้นไปด้วย ส่วนข้อเสีย เช่นเดียวกับแบบ 2 เฟส ที่ต้องจ่ายกำลังไฟเป็น 2 เท่าของแบบเวฟหรือจะใช้เท่ากับแบบ 2 เฟส

ดังนั้นเราสามารถนำลำดับการทำงานของ แบบครึ่งเฟส ในรูปของตารางได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.21 ขั้นตอนการทำงานแบบครึ่งสเตป

สเตปที่	เฟส 1	เฟส 2	เฟส 3	เฟส 4
1	ทำงาน			
2	ทำงาน	ทำงาน		
3		ทำงาน		
4		ทำงาน	ทำงาน	
5			ทำงาน	
6			ทำงาน	ทำงาน
7				ทำงาน
8	ทำงาน			ทำงาน
9	ทำงาน			
10	ทำงาน	ทำงาน		

2.24 โซลิดสเตทรีเลย์

โซลิดสเตทรีเลย์ (Solid State Relay : SSR) คือ อุปกรณ์อิเล็กทรอนิกส์ที่ทำหน้าที่เหมือนกับรีเลย์ ไฟฟ้ากล โดยปราศจากชิ้นส่วนเชิงกลที่เคลื่อนไหว โดยพื้นฐานแล้วโซลิดสเตทรีเลย์จะมีขั้วอินพุตและเอาต์พุตอย่างละสองขั้ว ขั้วอินพุตเป็นขั้วสำหรับป้อนสัญญาณควบคุมหรือสัญญาณกระตุ้น เพื่อบังคับให้สวิตช์ทางด้านขั้วเอาต์พุตปิดหรือเปิดโดยจะต้องมีการแยกทางไฟฟ้าระหว่างขั้วอินพุตกับขั้วเอาต์พุตมักใช้การเชื่อมโยงด้วยแสง (Optocoupling) ซึ่งต่างจากในรีเลย์ไฟฟ้ากลที่ใช้การเชื่อมโยงแบบแม่เหล็กไฟฟ้า (Electromagnitic Coupling) วงจรทั้งหมดของโซลิดสเตทรีเลย์จะถูกบรรจุไว้ในตัวถังเดียวกัน

2.24.1 ส่วนประกอบและการทำงานของโซลิดสเตทรีเลย์

โซลิดสเตทรีเลย์ส่วนใหญ่จะเป็นชนิดที่สวิตช์ทางด้านเอาต์พุตเป็นแบบ ขั้วเดียวทางเดียวปกติเปิด (SPST : NO) ทั้งนี้สาเหตุสำคัญเนื่องมาจากราคาต่อหน้าสัมผัสหนึ่งชุดยังคงสูงกว่าของรีเลย์ไฟฟ้ากลมาก เมื่อกล่าวถึงภาวะนำกระแส และภาวะไม่นำกระแส ของโซลิดสเตทรีเลย์ จะหมายถึง ภาวะที่ด้านอินพุตถูกกระตุ้น และไม่ถูกระตุ้น จะทำการอ้างถึงรีเลย์แบบปกติเปิดเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.24.2 คุณสมบัติทางด้านเอาต์พุตของโซลิดสเตทรีเลย์

1) กระแสโหลดสูงสุด (Maximum Load Current)

กระแสสูงสุดที่โซลิดสเตทรีเลย์สามารถทนได้ ขึ้นอยู่กับความสามารถในการระบายความร้อน เพราะว่ากระแสที่ทนได้อาจต่ำกว่าค่าที่ระบุก็ได้ ขึ้นอยู่กับแผ่นระบายความร้อนที่เลือกไว้ด้วย

2) กระแสโหลดต่ำสุด (Minimum Load Current)

กระแสโหลดต่ำสุดจะต้องไม่ต่ำกว่าที่กำหนดไว้โซลิดสเตทรีเลย์จึงจะทำงาน

2.25 การขยายออกทางลำโพง

หลักการกำเนิดเสียงโดยใช้ไมโครคอนโทรลเลอร์ ก็สามารถทำได้โดยการส่งเอาต์พุตให้มีการสลับไปมาอย่างรวดเร็วออกไปที่ลำโพง ก็คือ การส่ง “0” กับ “1” สลับกันจนเกิดเป็นความถี่เสียงขึ้นมา การกำเนิดเสียงสามารถแยกระดับของสัญญาณได้ถึง 255 ระดับ ซึ่งทำให้มีความละเอียดของเสียงมากขึ้น



รูปที่ 2.42 ระดับผลที่ได้จากลำโพง

สังเกตได้ว่า ถ้าเราสามารถกำหนดช่วงเวลาของการกำเนิดเสียง (T) ได้ ก็คือ เราสามารถกำหนดความถี่ของเสียงได้นั่นเอง โดย T = 1/F เราจึงสามารถทำให้ค่า T มีค่าน้อยมากๆ ได้ พอที่จะครอบคลุมช่วงความถี่ที่หูมนุษย์จะได้ยินอย่างสบาย (หูมนุษย์ได้ยินเสียงที่ช่วงความถี่ประมาณ 10 เฮิรตซ์ – 20 กิโลเฮิรตซ์) แต่อย่างไรก็ตาม เสียงตามตัวโน๊ตมีความถี่ที่กำหนดไว้แน่นอนตายตัวแล้ว

เสียงที่เกิดขึ้นจากการสังเคราะห์ทางโปรแกรมนี้ คงไม่สามารถทำให้นุ่มนวลเหมือนเสียงธรรมชาติได้ เพราะรูปคลื่นที่เกิดจากโปรแกรมทำขึ้นนั้น จะเป็นรูปคลื่นสี่เหลี่ยม (Square Wave) แต่อย่างไรก็ตาม การสังเคราะห์ก็มีประโยชน์หลายด้านด้วยกับ เช่น เป็นสัญญาณเตือน หรือเป็นสัญญาณที่สื่อความหมายใดๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การสร้างและการออกแบบ

3.1 กล่าวนำ

ชุดปฏิบัติการทดลองไมโครคอนโทรลเลอร์ PIC16F8XX ประกอบด้วย 2 ส่วน คือ ส่วนของฮาร์ดแวร์ (Hardware) และส่วนของซอฟต์แวร์ (Software) ซึ่งส่วนของซอฟต์แวร์ คือโปรแกรม PIC-ICSP Device สำหรับควบคุมการทำงานของโมดูลหลัก ที่ใช้โหลดข้อมูลโปรแกรมจากคอมพิวเตอร์ ลงในหน่วยความจำโปรแกรมของ PIC16F8XX และส่วนที่เป็นฮาร์ดแวร์นั้นประกอบไปด้วยวงจรต่างๆ ที่ใช้ในการทดลองเชื่อมต่อกับไมโครคอนโทรลเลอร์ PIC16F8XX โดยได้ออกแบบและสร้างอยู่ในรูปของ “โมดูล” (Module) แต่ละโมดูลจะเชื่อมต่อกับโมดูลหลักโดยใช้สายแพ เมื่อต้องการทดลองเรียนรู้ในเรื่องใดก็นำโมดูลนั้นมาเชื่อมต่อใช้งาน

3.2 การออกแบบทางด้านฮาร์ดแวร์

3.2.1 โมดูลหลัก PIC-ICSP (PIC- In-Circuit Serial Programming Module)

PIC-ICSP Module เป็นโมดูลตัวแม่สำหรับการทดลองใช้งานไมโครคอลโทรลเลอร์ตระกูล PIC16F8XX ซึ่งสามารถใช้งานได้หลายเบอร์ ทั้งแบบที่มีขาต่อใช้งาน 18 ขา แบบ 28 ขา และแบบ 40 ขา โดยเบอร์ไอซีที่ใช้งานกับ PIC-ICSP Module มีดังนี้

แบบ 18 ขา เช่น เบอร์ PIC16F83 PIC16F84 และ PIC16F84A

แบบ 28 ขา เช่น เบอร์ PIC16F873 และPIC16F876

แบบ 40 ขา เช่น เบอร์ PIC16F874 และPIC16F877

PIC-ICSP Module สามารถดาวน์โหลด (Down Load) ไฟล์เฮก (Hex File) จากคอมพิวเตอร์ลงบนตัวไมโครคอนโทรลเลอร์ได้ ในรูปแบบ In-Circuit Serial Programming จากพอร์ตขนานของคอมพิวเตอร์ และสั่งรัน (Run)โปรแกรมได้ทันทีโดยสั่งงานจากเครื่องคอมพิวเตอร์

ตัวโมดูลมีคอนเนคเตอร์ (Connector) ให้ต่อใช้งานโดยแยกเป็นพอร์ต จำนวน 5 พอร์ต โดยใช้งานตามคุณสมบัติของไมโครคอนโทรลเลอร์แต่ละเบอร์ และมีคอนเนคเตอร์ที่รวมพอร์ตทุกพอร์ตขนาด 40 ขา ไว้ใช้งานด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) คุณสมบัติของ PIC-ICSP Module

1.1) สามารถใช้ได้กับไมโครคอนโทรลเลอร์ตระกูล PIC16F8XX หลายเบอร์ เช่น เบอร์ PIC16F876 และเบอร์ PIC16F877

1.2) สามารถเลือกโหมดการใช้สัญญาณนาฬิกาได้ทั้ง 2 แบบ คือ แบบคริสตอล (Crystal) 4 เมกะเฮิรตซ์ และแบบ RC

1.3) โปรแกรมในรูปแบบดาวน์โหลดด้วยวิธี In-Circuit Serial Programming

1.4) สั่งทำงาน (Run) หยุดทำงาน (Stop) และรีเซต (Reset) ได้จากคอมพิวเตอร์

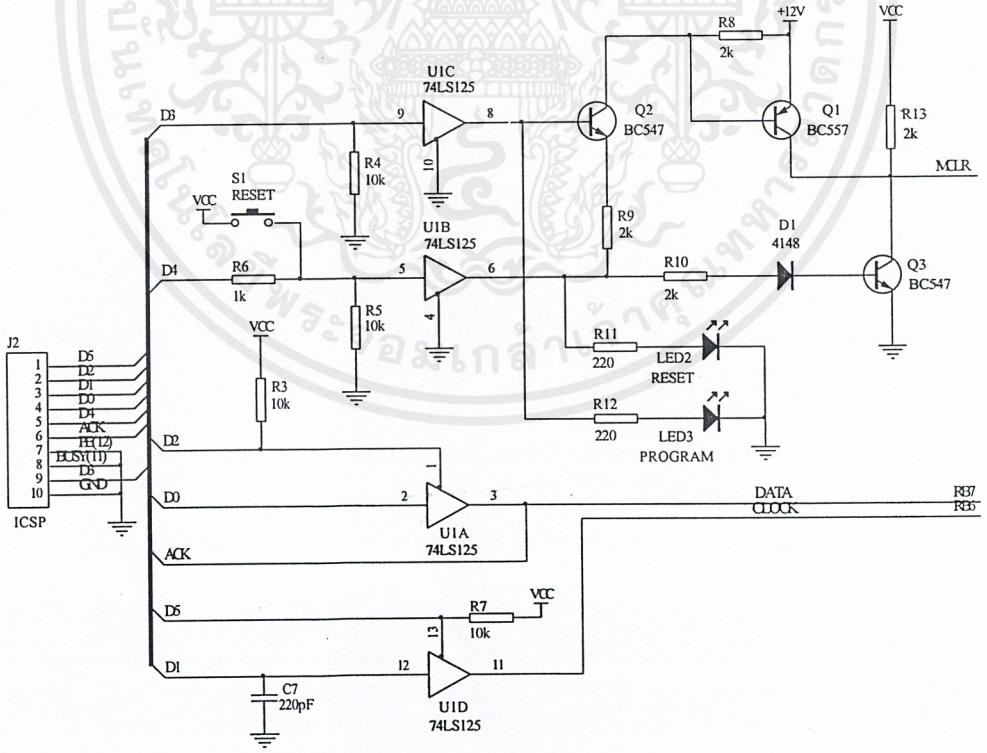
1.5) เชื่อมต่อกับคอมพิวเตอร์ทางพอร์ตขนาน (Parallel Port)

1.6) คอนเนคเตอร์ต่อใช้งานแยกเป็น 5 พอร์ต เป็นแบบ 10 ขา มีแรงดัน 5 โวลต์ สำหรับจ่ายให้โมดูลที่มาต่อพ่วง

1.7) คอนเนคเตอร์รวมพอร์ตต่อใช้งานทุกพอร์ตแบบ 40 ขา

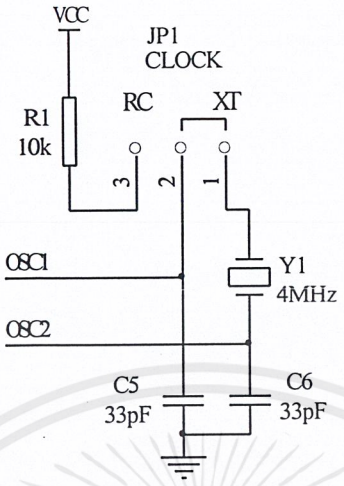
1.8) ใช้แหล่งจ่ายแรงดัน 5 โวลต์ และ 12 โวลต์

2) วงจรของ PIC-ICSP Module



รูปที่ 3.1 วงจรภาค In-Circuit Serial Programming ของ PIC-ICSP Module

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



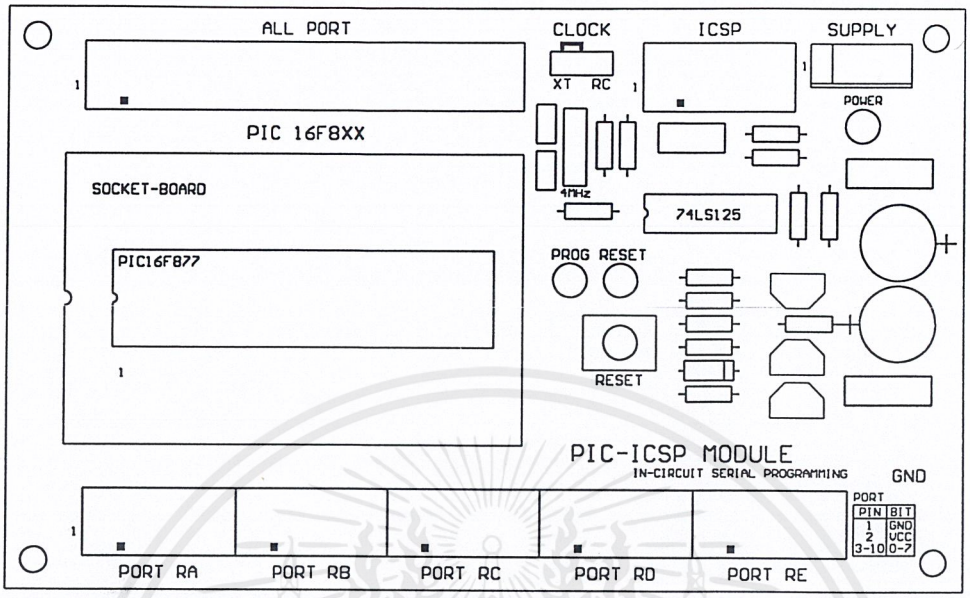
รูปที่ 3.2 วงจรภาคกำเนิดสัญญาณนาฬิกา

คุณสมบัติการโปรแกรมของไมโครคอนโทรลเลอร์ตระกูล PIC16F8XX จะใช้ขาสัญญาณจำนวน 3 เส้น คือ Data (RB7) Clock (RB6) และ MCLR ในการโปรแกรม ส่งข้อมูลในแบบอนุกรม ขา MCLR จะรับแรงดันไฟ 12 โวลต์ จากวงจรรูปที่ 3.6 เป็นวงจรสำหรับดาวน์โหลดข้อมูลผ่านทางพอร์ตขนานของคอมพิวเตอร์ โดยจะผ่านไอซีเบอร์ 74LS125 ทำหน้าที่เป็นบัฟเฟอร์ (Buffer) ซึ่งเป็นแบบเอาต์พุต 3 สถานะ (State) ใช้แยกสัญญาณที่พอร์ต RB6 และ RB7 ในขณะรันและโปรแกรม

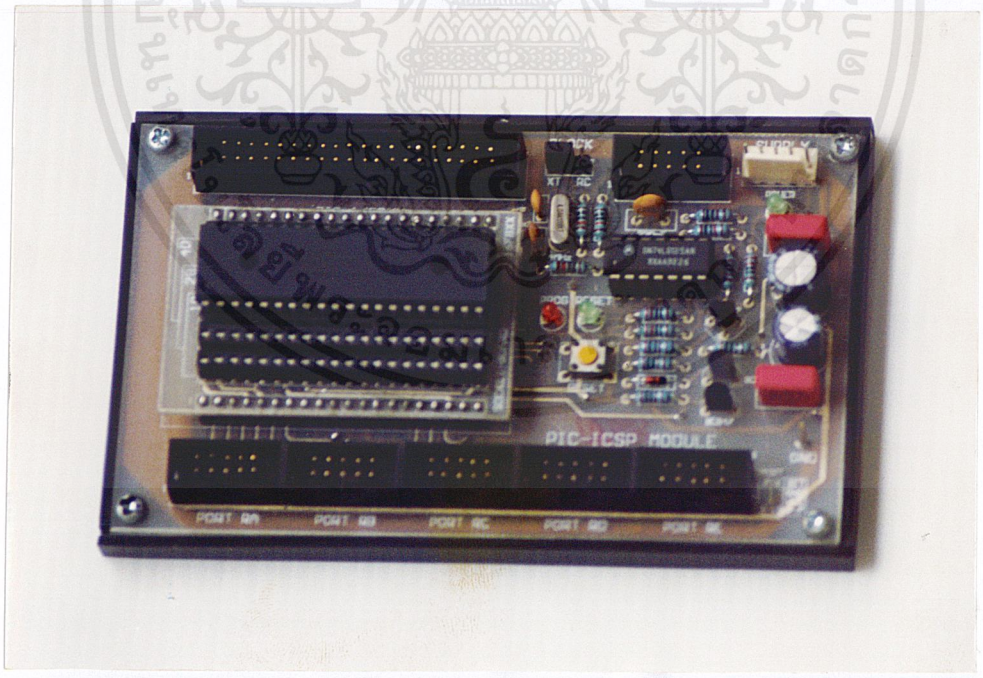
ที่ขา MCLR จะมีทรานซิสเตอร์สำหรับป้อนแรงดันให้ CPU โดยจะมีแรงดันอยู่ 3 ระดับ คือ แรงดัน 12 โวลต์ สำหรับการโปรแกรม แรงดัน 5 โวลต์ สำหรับขณะรัน และแรงดัน 0 โวลต์ เมื่อรีเซต (Reset) วงจรมี LED ใช้แสดงการโปรแกรมและการรีเซตของซีพียู สำหรับการรีเซตนั้นสามารถสั่งได้ทั้งจากคอมพิวเตอร์หรือโดยการกดสวิตช์บนตัวโมดูล

สัญญาณนาฬิกาที่ป้อนให้กับซีพียูสามารถเลือกได้ทั้งแบบคริสตอลความถี่ 4 เมกะเฮิรตซ์ หรือ แบบ RC โดยจะเลือกจากจั๊มเปอร์ (Jumper) ชื่อ CLOCK ซึ่งการที่สามารถเลือกสัญญาณนาฬิกาได้หลายแบบนี้ เป็นคุณสมบัติพิเศษของไมโครคอนโทรลเลอร์ตระกูล PIC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 แผนผังอุปกรณ์ของโมดูลหลัก PIC-ICSP

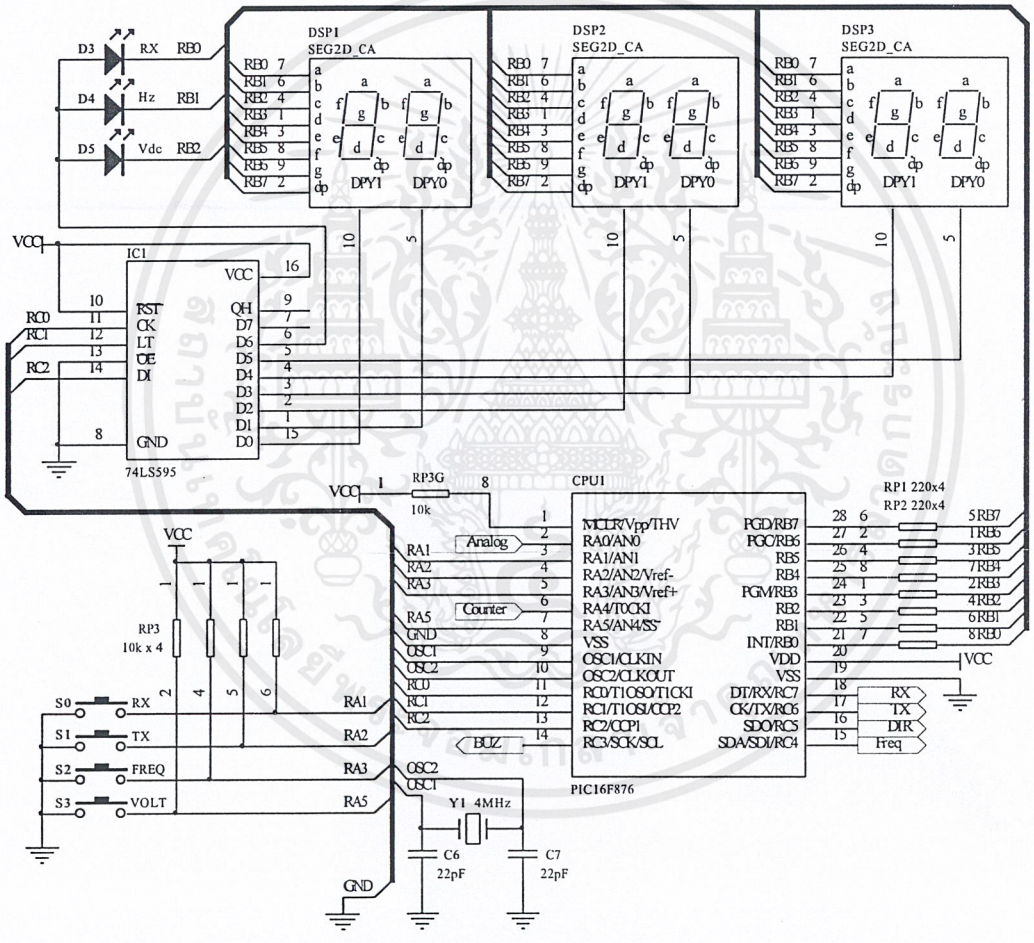


รูปที่ 3.4 โมดูลหลัก PIC-ICSP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 โมดูลเครื่องมือวัด (Measurement Module)

โมดูลประกอบไปด้วยเครื่องมือวัด สำหรับใช้ประกอบการทดลองใบงาน โดยโมดูลนี้จะใช้ไมโครคอนโทรลเลอร์ PIC16F876 เป็นตัวประมวลผลและควบคุมการทำงาน การแสดงผลใช้ LED ตัวเลข 7 ส่วนขนาด 6 หลัก ซึ่งจะใช้ไอซีเปลี่ยนสัญญาณอนุกรมเป็นขนาน เบอร์ 74LS595 ขยายพอร์ตของไมโครคอนโทรลเลอร์ในการขับ LED ตัวเลข 7 ส่วน วงจรส่วนของภาคแสดงผลแสดงดังรูปที่ 3.5



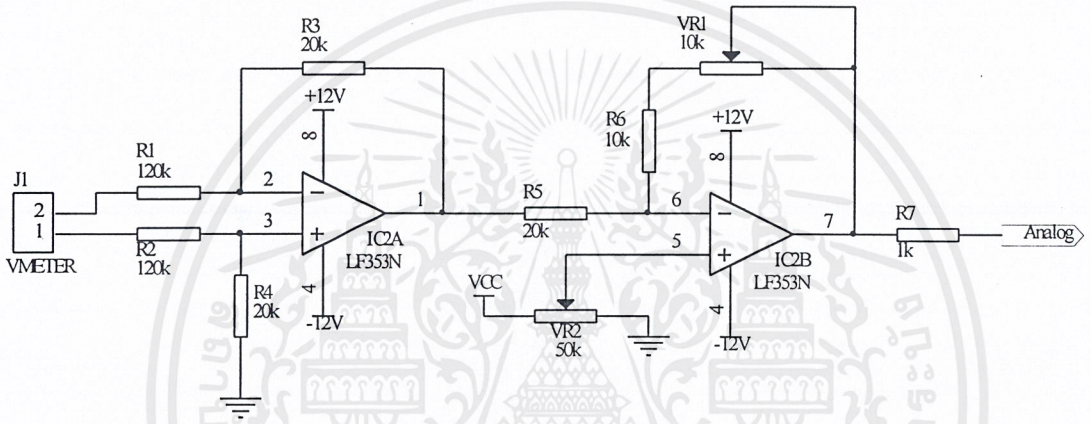
รูปที่ 3.5 วงจรหลักและวงจรภาคแสดงผลของโมดูลเครื่องมือวัด

จากวงจรรูปที่ 3.5 ประกอบไปด้วยวงจรภาคคีย์สวิตช์ (Switch Key) ที่ใช้เลือกชนิดของเครื่องมือวัด ตัวประมวลผล และภาคแสดงผล และ LED สำหรับแสดงสถานะของเครื่องมือวัดว่าเป็นเครื่องมือวัดชนิดใด ซึ่งจะบอกเป็นหน่วยของการวัดของเครื่องมือวัดชนิดนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) เครื่องวัดแรงดันไฟฟ้ากระแสตรง (DC Volt Meter)

เครื่องวัดแรงดันไฟฟ้ากระแสตรงสามารถวัดได้ $\pm 25\ V_{DC}$ ใช้ตัวแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอล ที่มีอยู่แล้วในไมโครคอนโทรลเลอร์ PIC16F8XX ซึ่งเป็น ADC ขนาด 10 บิต สามารถแสดงผลการวัดด้วยความละเอียดเป็นจุดทศนิยม 1½ หลัก ด้านอินพุตของเครื่องวัดนี้ใช้ออปแอมป์ทำงานเป็นวงจรขยายความแตกต่าง และเพื่อลดระดับแรงดันที่วัดได้ให้อยู่ในช่วง 0-5 V_{DC} ก่อนจะป้อนให้ ADC ในไมโครคอนโทรลเลอร์ วงจรแสดงดังรูปที่ 3.6



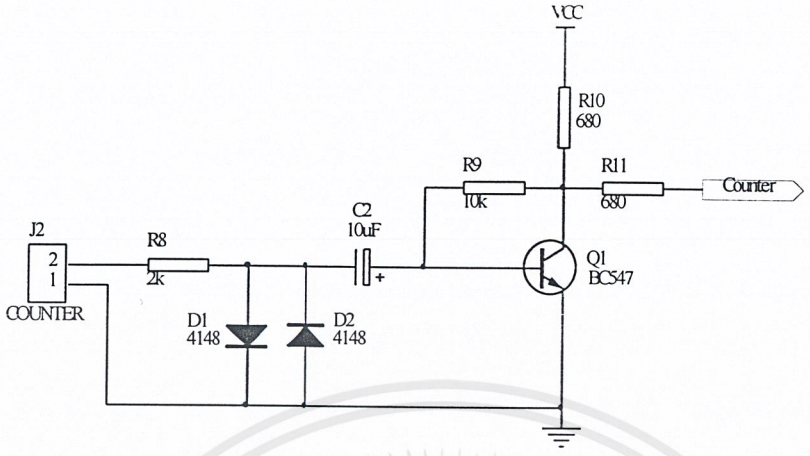
รูปที่ 3.6 วงจรของภาคโวลต์มิเตอร์

วงจรด้านอินพุต จะทำหน้าที่ปรับระดับแรงดันให้เหมาะสมกับที่ตัว ADC รับได้ และยังทำหน้าที่ปรับแรงดันอ้างอิงให้ ADC ด้วย คือถ้ายังไม่ทำการวัด ออปแอมป์จะจ่ายแรงดัน 2.5 โวลต์ให้แก่ ADC เพื่อแสดงผลออกมาเป็น 0.00 V แต่เมื่อทำการวัด แรงดันนี้ก็จะสวิงจาก 0 ถึง 5 โวลต์ และแสดงผลออกมาเป็น –25.50 และ 25.50 โวลต์ ตามลำดับ

2) เครื่องนับความถี่ (Frequency Meter)

เครื่องนับความถี่สามารถวัดความถี่ได้ตั้งแต่ 2 เฮิรตซ์ จนถึง 999,999 เฮิรตซ์ โดยจะใช้ Timer / Counter ในตัวไมโครคอนโทรลเลอร์เป็นตัวนับความถี่ของสัญญาณ วงจรด้านอินพุตจะเป็นตัวที่ทำหน้าที่ปรับระดับแรงดันของสัญญาณที่วัดได้ให้เหมาะสม ก่อนที่จะป้อนแรงดันเข้าไปสู่ไมโครคอนโทรลเลอร์ ใช้ทรานซิสเตอร์เป็นตัวขยายความถี่ในวงจร เพราะสามารถทำงานที่ความถี่ได้สูงกว่าใช้ออปแอมป์ วงจรแสดงดังรูปที่ 3.7

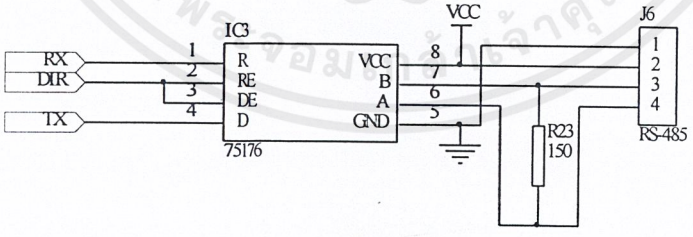
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 วงจรภาคเครื่องนับความถี่

3) ภาคส่งสัญญาณข้อมูลแบบอนุกรม

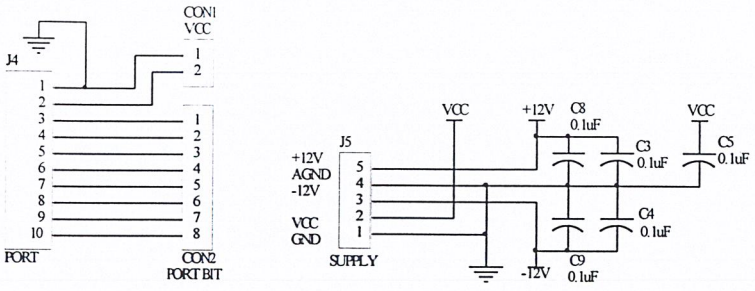
ภาคส่งสัญญาณข้อมูลแบบอนุกรมจะใช้สำหรับทดลองการส่งข้อมูลแบบ RS-485 ระหว่างโมดูลสื่อสารข้อมูลแบบอนุกรม การส่งข้อมูลจะใช้ไอซี SN75176 เป็นตัวส่งโดยวงจรจะเหมือนกับภาค RS-485 ของโมดูลสื่อสารแบบอนุกรม สัญญาณข้อมูลที่จะส่งออกมานั้น เป็นข้อมูลที่แสดงผลบน Seven Segment ทั้ง 6 หลัก ที่แสดงผลอยู่ขณะนั้น โดยเมื่อต้องการส่งข้อมูลก็กดปุ่ม TX ส่วนการรับข้อมูล เมื่อ กดปุ่ม RX จะเข้าสู่โหมดการรับข้อมูล ให้ส่งข้อมูลจาก 0x00 ถึง 0x09 เข้ามา ก็จะนำตัวเลขนั้นมาแสดงผล เลื่อนตัวเลขนั้นไปทางซ้ายเรื่อยๆ วงจรของภาคนี้แสดงดังรูปที่ 3.8



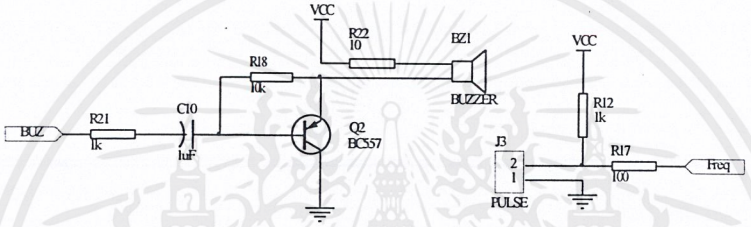
รูปที่ 3.8 วงจรภาครับส่งข้อมูลอนุกรม RS-485

วงจรส่วนอื่นๆ ของโมดูลเครื่องมือวัด ประกอบไปด้วย ภาคขับบัซเซอร์สำหรับแสดงเสียง ภาคจ่ายแรงดัน คอนเนคเตอร์สำหรับวัดสัญญาณจากพอร์ต และภาคกำเนิดสัญญาณพัลซ์ขนาด 1 กิโลเฮิรตซ์ สำหรับทดลอง วงจรของภาคนี้แสดงดังรูปที่ 3.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

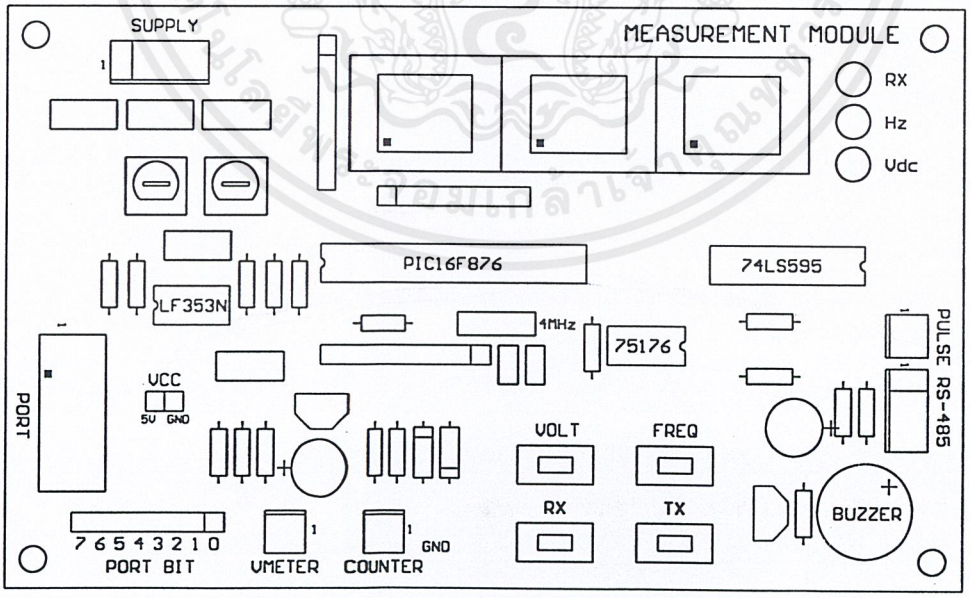


(ก) คอนเนคเตอร์วัดสัญญาณจากพอร์ต (ข) ภาคจ่ายแรงดัน



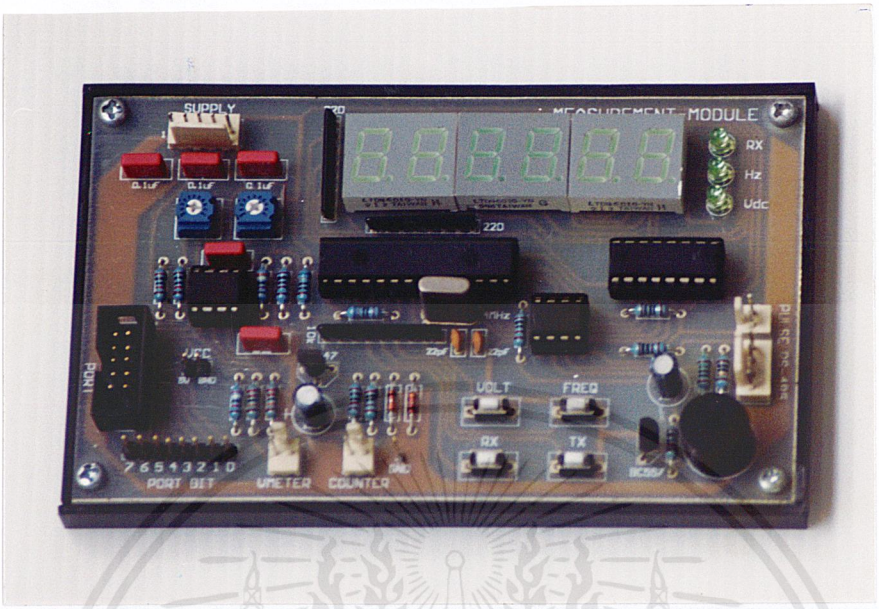
(ค) ภาคขับบัซเซอร์ (ง) ภาคกำเนิดสัญญาณพัลซ์

รูปที่ 3.9 วงจรในส่วนอื่นของโมดูลเครื่องมือวัด



รูปที่ 3.10 แผนผังอุปกรณ์ของโมดูลเครื่องมือวัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 โมดูลเครื่องมือวัด

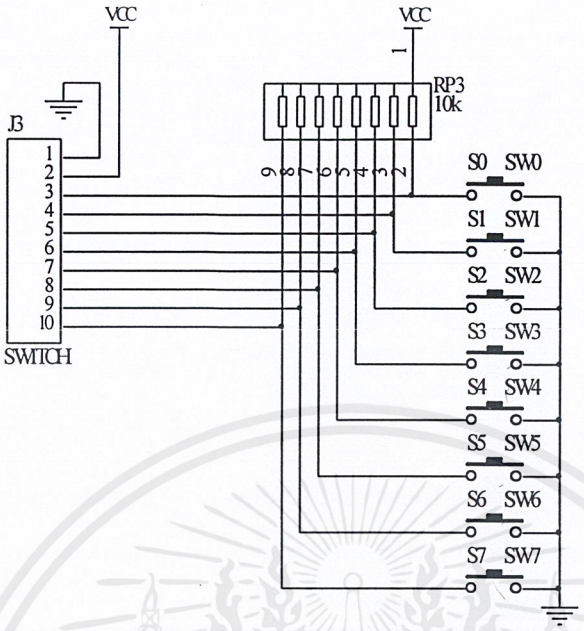
3.2.3 โมดูลเบสิกอินพุตเอาต์พุตพื้นฐาน (Basic Input Output Module)

โมดูลเบสิกอินพุตเอาต์พุตพื้นฐาน จะประกอบไปด้วยส่วนของสวิตช์สำหรับสร้างสัญญาณอินพุตให้ไมโครคอนโทรลเลอร์ PIC16F8XX และส่วนของ LED ที่ใช้สำหรับแสดงผลข้อมูลจากไมโครคอนโทรลเลอร์ โดยมีรายละเอียดดังนี้

1) เบสิกสวิตช์พื้นฐาน

เบสิกสวิตช์พื้นฐานจะเป็นตัวสร้างสัญญาณอินพุตพื้นฐาน เพื่อให้ไมโครคอนโทรลเลอร์รับข้อมูลเข้าไปประมวลผล ในโมดูลนี้จะมีสวิตช์อยู่จำนวน 8 ตัว แต่ละตัวจะใช้ต่อเข้ากับแต่ละบิตของพอร์ต วงจรจะมีลักษณะตามรูปที่ 3.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 วงจรของส่วนเบสิกสวิตช์พื้นฐาน

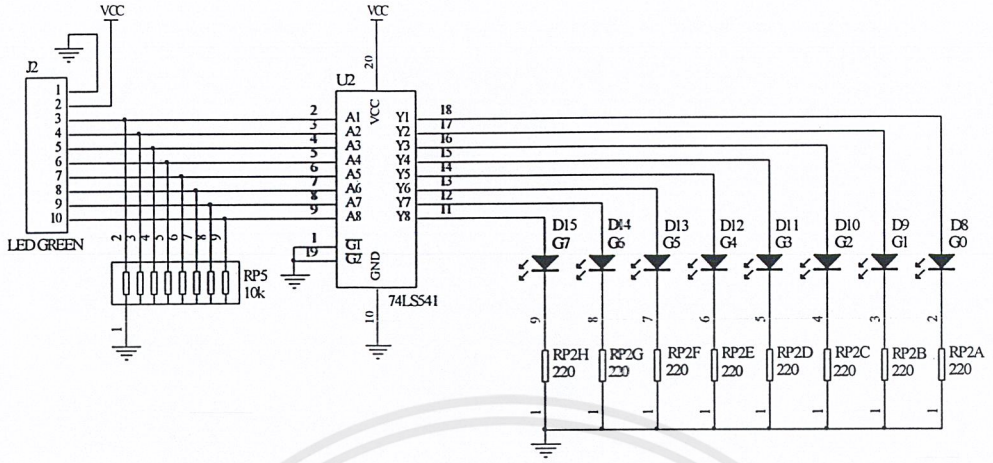
สวิตช์แต่ละตัวจะมีการต่อตัวต้านทานพูลอัป (Pull Up) ไว้ หากไม่กดสวิตช์ จะอ่านค่าได้เป็น “1” เพราะตัวต้านทานต่ออยู่กับแรงดัน VCC เมื่อไม่กดสวิตช์แรงดัน VCC ก็จะสามารถไหลผ่านตัวต้านทานพูลอัปเข้าสู่พอร์ต ไมโครคอนโทรลเลอร์จึงอ่านค่าได้เป็น “1” และเมื่อกดสวิตช์แรงดันที่ไหลผ่านตัวต้านทานจะลงสู่ GND ผ่านสวิตช์ที่ปิดวงจร ทำให้อ่านค่าจากสวิตช์ได้เป็น “0”

2) LED แสดงผลพื้นฐาน

ใช้สำหรับแสดงผลข้อมูลจากพอร์ตแบบพื้นฐานที่สุด โดยใช้แสดงระดับลอจิกจากพอร์ตว่าเป็น “0” หรือ “1” โมดูลนี้จะมีชุด LED แสดงผลอยู่ 2 ชุด แบ่งเป็น 2 สี คือ สีแดงและสีเขียว ซึ่งแต่ละชุดมี LED อยู่ 8 ดวง แต่ละดวง หมายถึง แต่ละบิตของพอร์ต จึงสามารถแสดงผลได้ทั้งหมด 16 บิต โดยวงจรมีลักษณะดังรูปที่ 3.13

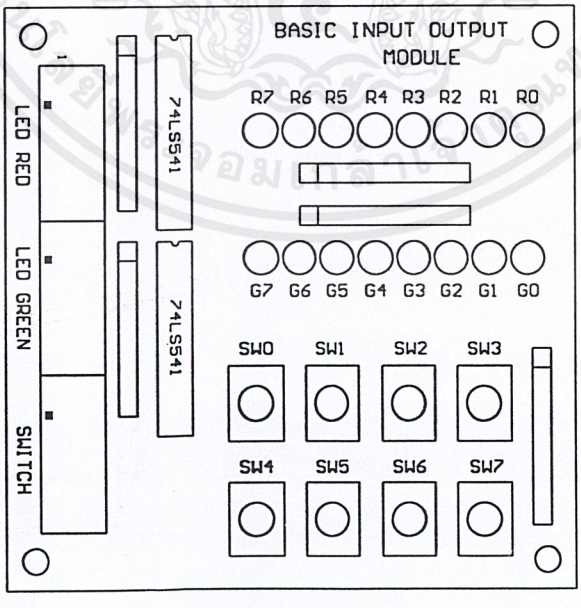
ในวงจรรูปที่ 3.13 เป็นวงจรแสดง LED 1 ชุด โดยวงจรอีก 1 ชุดจะมีลักษณะเหมือนกัน ภายในวงจรจะมีพอร์ตสำหรับต่อกับไมโครคอนโทรลเลอร์รับข้อมูลเข้ามาแสดงผล โดยจะผ่านไอซีเบอร์ 74LS541 ทำหน้าที่เป็นบัฟเฟอร์ ที่ด้านขาเข้าของไอซีจะมีตัวต้านทานต่อเป็นพูลดาวน์ (Pull Down) ให้แก่ไอซีเพื่อต้องการให้ระดับลอจิกเป็น “0” ขณะไม่ป้อนสัญญาณใดๆ เข้ามาหรือหากป้อนลอจิก “0” เข้ามา หลอด LED ก็จะไม่ติด เมื่อมีสัญญาณเข้ามาเป็น “1” ผ่านไอซีบัฟเฟอร์จะทำให้มีแรงดัน VCC ไหลเข้าขาอาโนด (Anode) ของหลอด LED ผ่านตัวต้านทานจำกัดกระแสแล้วไหลลงกราวน์ครบวงจร ทำให้หลอด LED ติด แสดงสภาวะลอจิก “1” ของบิตนั้นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



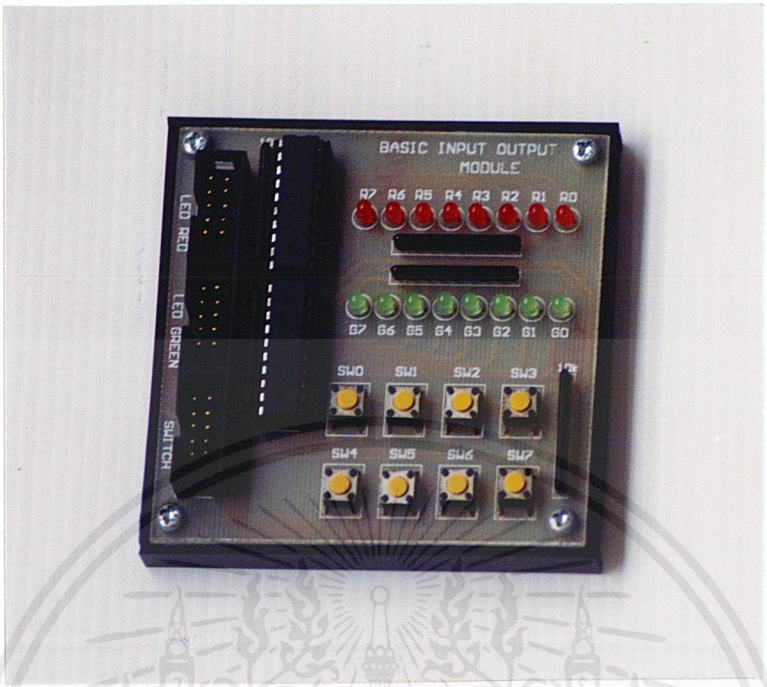
รูปที่ 3.13 วงจรของส่วน LED แสดงผลพื้นฐาน

ในตัวไมโครคอนโทรลเลอร์ PIC16F8XX นั้น สามารถที่จะกำหนดให้เป็นอินพุตหรือเป็นเอาต์พุตได้ทุกพอร์ต คอนเนคเตอร์ที่ต่อจะแบ่งเป็น 3 ตัว สำหรับต่อกับชุดเบสิกสวิตช์ 1 ตัว และต่อกับชุด LED แสดงผล 2 ตัว ชุดเบสิกสวิตช์ต่อเข้ากับคอนเนคเตอร์ชื่อ SWITCH มีสวิตช์อยู่ 8 ตัว ชื่อ SW0 ถึง SW7 ส่วนชุด LED แสดงผลจะมีคอนเนคเตอร์ชื่อ LED RED ต่อกับ LED สีแดง ชื่อ R0 ถึง R7 และคอนเนคเตอร์ชื่อ LED GREEN ต่อกับ LED ชื่อ G0 ถึง G7



รูปที่ 3.14 แผนผังอุปกรณ์ของโมดูลเบสิกอินพุตเอาต์พุตพื้นฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.15 โมดูลเบสิกอินพุตเอาต์พุตพื้นฐาน

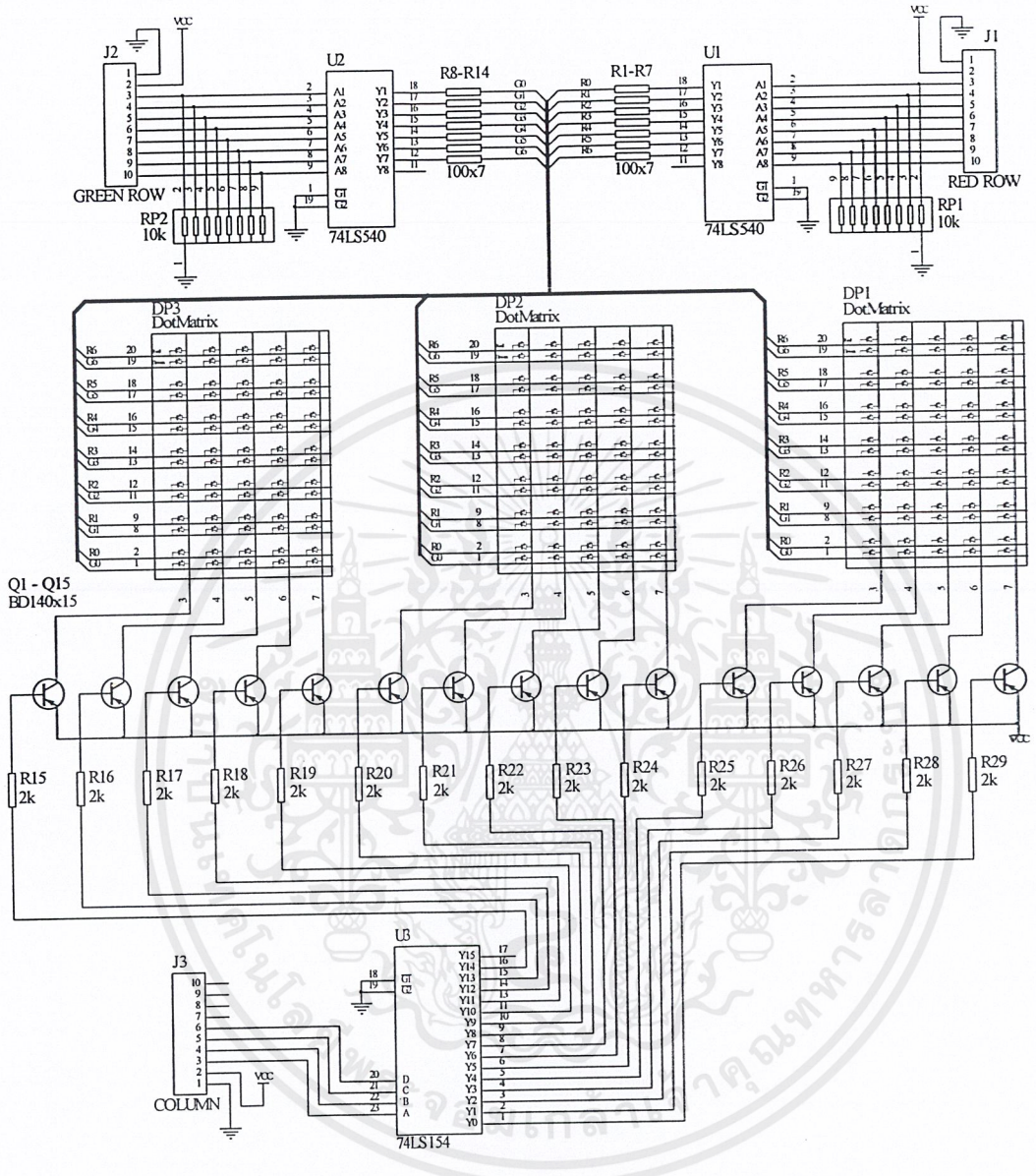
3.2.4 โมดูลแสดงผลแอลอีดีดอตเมตริกซ์ (LED Dot Matrix Display Module)

โมดูลแสดงผลแอลอีดีดอตเมตริกซ์ขนาด 7x15 จุด (7 แถว 15 คอลัมน์) แสดงผลได้ 2 สี คือสีแดงและสีเขียว โดยแต่ละสีจะแยกพอร์ตสำหรับส่งข้อมูลคอลัมน์ สามารถสั่งให้ติดพร้อมกันได้ทั้งสองสี ซึ่งจะมองเห็นเป็นอีกหนึ่งสีคือสีเหลือง จึงทำให้สามารถแสดงผลได้ 3 สี

วงจรรูปที่ 3.16 วงจรแอลอีดีดอตเมตริกซ์ที่ใช้เป็นชนิดคอมมอนอาโนด (Common Anode) การส่งข้อมูลทางด้านคอลัมน์จะผ่านไอซีบัฟเฟอร์ซึ่งทำงานเป็น NOT Gate ดังนั้นข้อมูลที่ทำให้หลอดแอลอีดีติด คือ ต้องส่งสภาวะลอจิก “1” ให้ด้านคอลัมน์

ทางด้านการเลือกหลักจะส่งข้อมูลผ่านไอซีดีโค้ดเดอร์เบอร์ 74LS154 ขนาดเข้า 4 ออก 16 จะใช้เลือกหลักของแอลอีดีที่ต้องการให้ติด ทรานซิสเตอร์ชนิด เอ็นพีเอ็น (NPN) ทำหน้าที่ขยายกระแสให้แอลอีดี 1 ตัวต่อ 1 คอลัมน์ และยังทำหน้าที่กลับสภาวะของสัญญาณด้วยเพราะแอลอีดีเป็นชนิดคอมมอนอาโนด แต่เอาต์พุตของไอซีดีโค้ดเดอร์แอคตีฟที่ลอจิก “0” ดังนั้นกระแสที่ไหลผ่านทรานซิสเตอร์สูงสุดนั้นจะเท่ากับ กระแสที่ให้แอลอีดีจำนวน 14 หลอดติดพร้อมกัน เนื่องจากแอลอีดีในหนึ่งคอลัมน์ประกอบไปด้วยแอลอีดีสีแดงจำนวน 7 หลอด และ สีเขียวอีก 7 หลอด ที่สามารถสั่งให้ติดได้พร้อมกัน จึงใช้ทรานซิสเตอร์เบอร์ BD140 ให้สามารถจ่ายกระแสได้สูงอย่างปลอดภัย

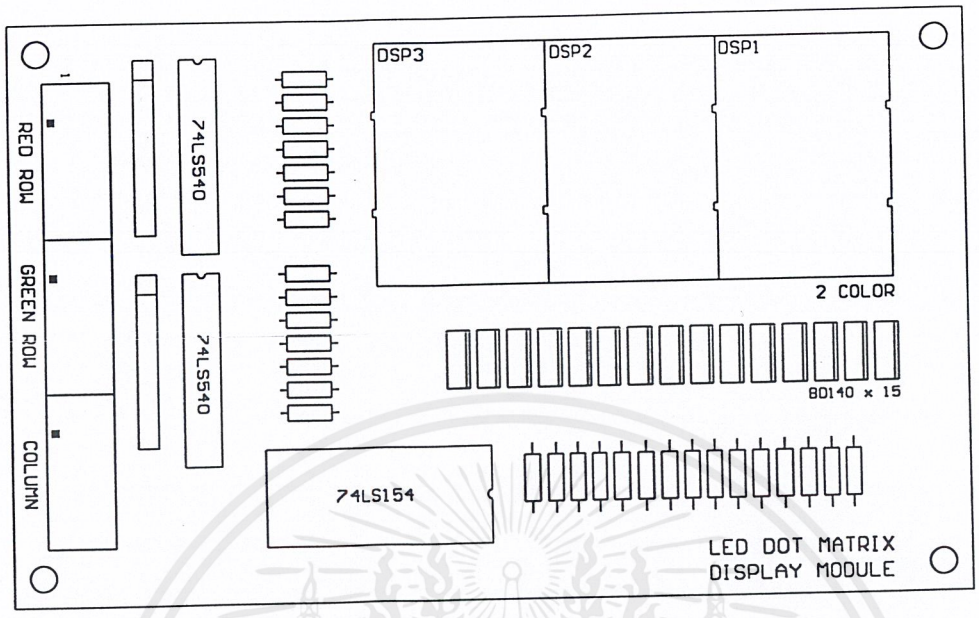
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



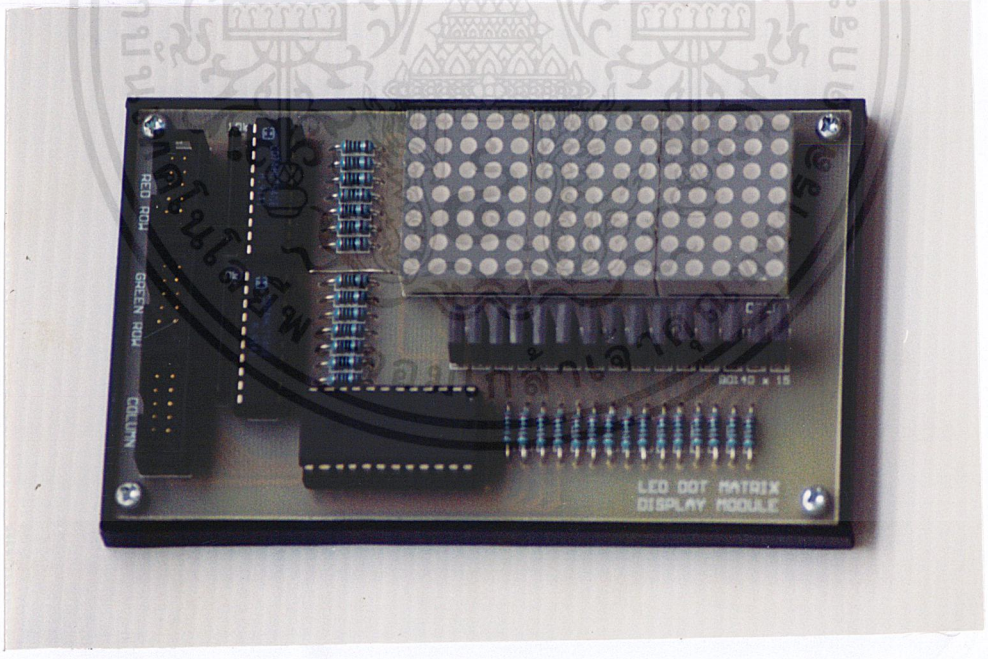
รูปที่ 3.16 วงจรของโมดูลแสดงผลแอลอีดีดอตเมตริกซ์

การใช้งานโมดูลแสดงผลแอลอีดีดอตเมตริกซ์จะต้องใช้สัญญาณข้อมูลอย่างน้อย 12 บิต เพื่อให้แอลอีดีติด 1 สีและ 20 บิต สำหรับให้แอลอีดีติด 2 สี หรือ 3 สี (สองสีติดพร้อมกัน) ตัวโมดูลมีลักษณะดังรูปที่ 3.17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.17 แผนผังอุปกรณ์ของโมดูลแสดงผลแอลอีดีดอตเมตริกซ์



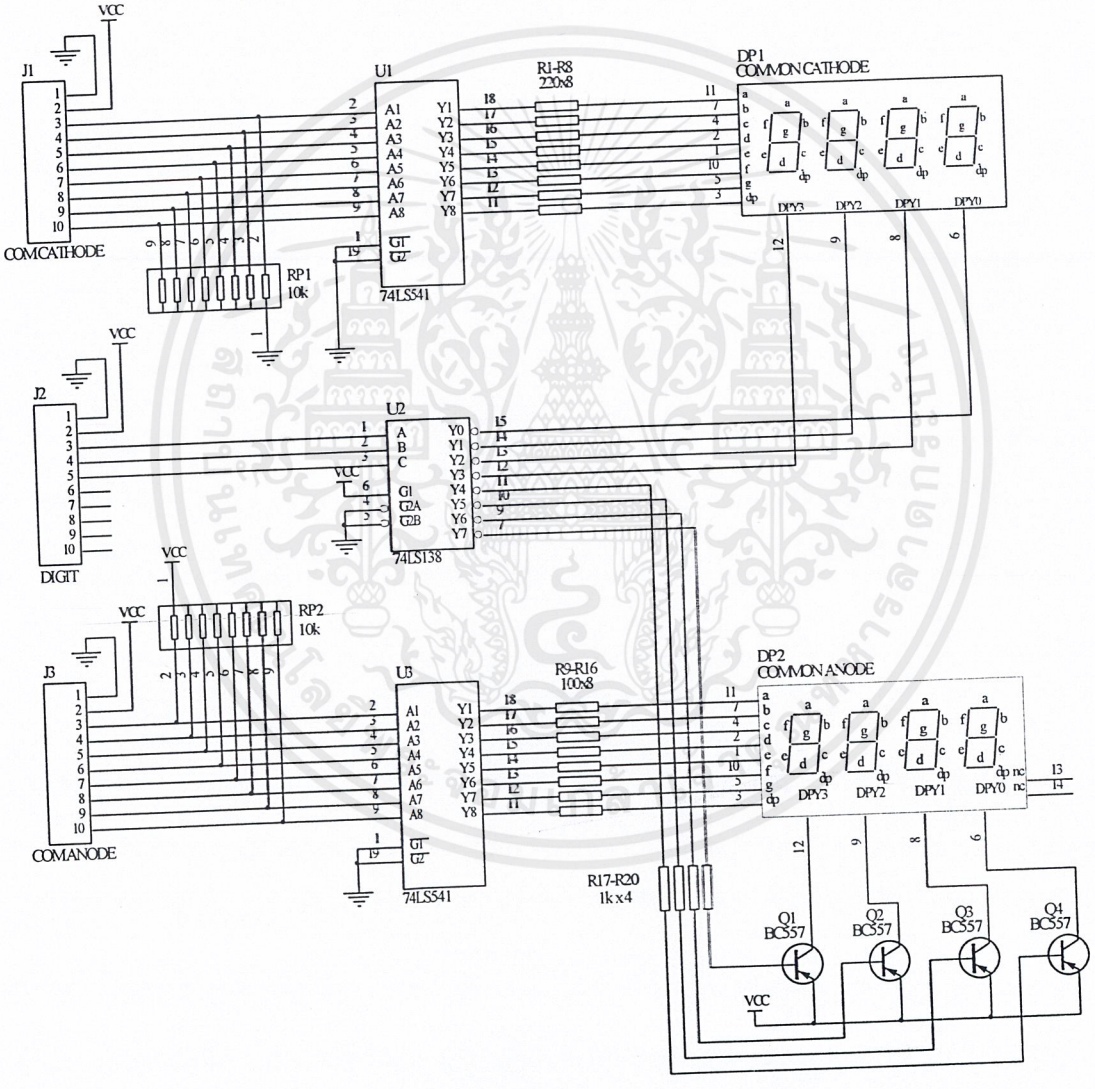
รูปที่ 3.18 โมดูลแสดงผลแอลอีดีดอตเมตริกซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5 โมดูลแสดงผลตัวเลข 7 ส่วน (Seven Segment Display Module)

โมดูลสำหรับแสดงผลแบบตัวเลข 7 ส่วน (7 Segment) มีทั้งแบบคอมมอนคาโถดและแบบคอมมอนอาโนดให้ใช้งาน แต่ละแบบจะมีตัวเลข 7 ส่วนอยู่ 4 หลัก (4 Digit) มีวงจรดังรูปที่ 3.19

ส่วนของคอมมอนอาโนดและคอมมอนคาโถดจะแยกกันเพื่อให้เลือกใช้งานในแต่ละแบบ โดยจะมีส่วนที่ใช้ร่วมกันคือตัวเลือกหลัก



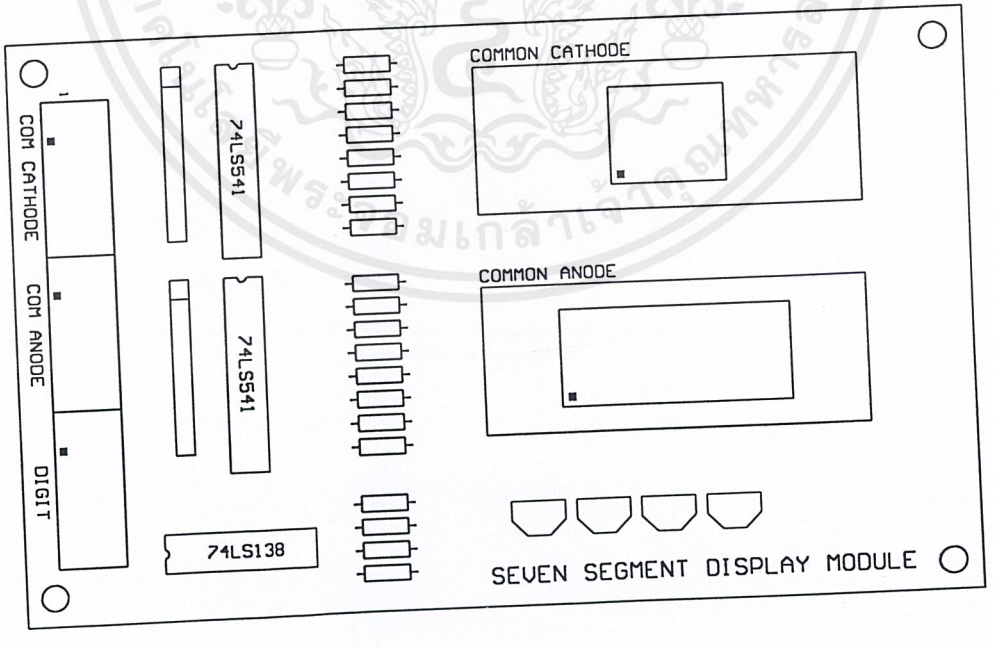
รูปที่ 3.19 วงจรของโมดูลแสดงผลตัวเลข 7 ส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของคอมมอนคาโถดจะมีพอร์ตสำหรับส่งสัญญาณข้อมูลเลือกเซกเมนต์ที่จะติดเรียกว่า Data มีจำนวน 8 บิต คือ บิต 0 ถึง 7 เพื่อขับเซกเมนต์ที่ชื่อ a, b, c, d, e, f, g, dp ตามลำดับ สัญญาณจะมาผ่านไอซีบัฟเฟอร์เบอร์ 74LS541 เพื่อขับกระแสให้ LED มีตัวต้านทาน 220 โอห์มจำกัดกระแส ที่ด้านขาเข้าของไอซีมีตัวต้านทานต่อพูลดาวน์อยู่เพื่อข้อมูลปกติเป็นลอจิก “0” เพื่อไม่ให้ LED ติดเมื่อไม่ต่อใช้งาน และเมื่อต้องการให้เซกเมนต์ใดติดก็ป้อนลอจิก “1” เข้ามาตามลักษณะของคอมมอนคาโถด

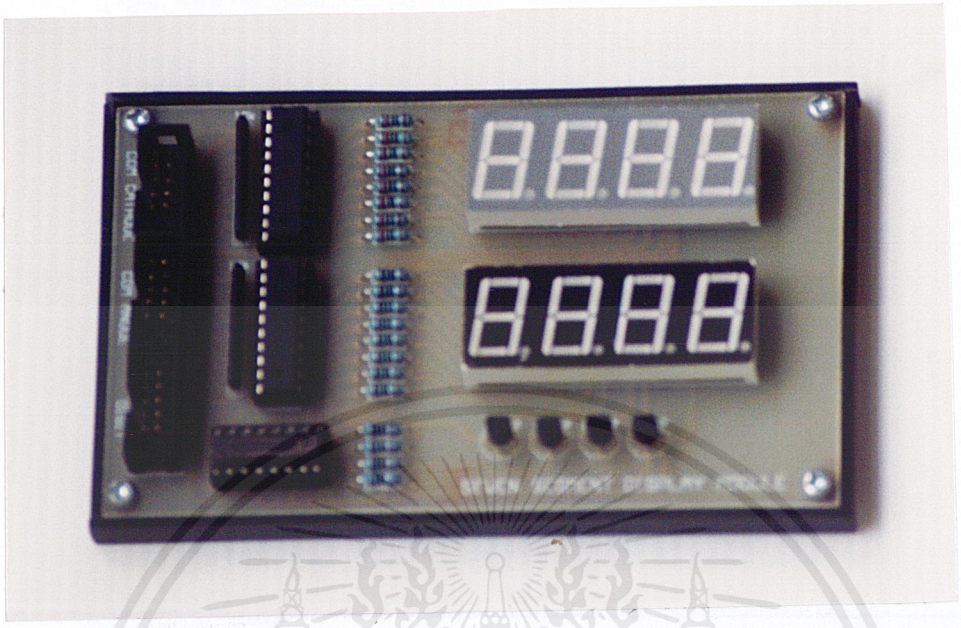
ตัวเลือกหลักจะใช้ไอซีดีโค้ดเดอร์เบอร์ 74LS138 ใช้งานร่วมกันทั้งสองคอมมอน โดยที่เอาต์พุต Y0 ถึง Y3 จะเป็นตัวเลือกหลักของคอมมอนคาโถด และ Y4 ถึง Y7 ใช้ในการเลือกหลักของคอมมอนอาโนด ในส่วนของคอมมอนอาโนดจะมีลักษณะวงจรคล้ายกัน แต่จะต่างกันตรงสภาวะที่ป้อนให้แก่เซกเมนต์จะใช้ “0” เพื่อให้เซกเมนต์ติดตามลักษณะของคอมมอนอาโนด ส่วนตัวเลือกหลักที่ใช้ร่วมกันนั้น จะมีทรานซิสเตอร์เพื่อกลับสภาวะแรงดันจากแรงดัน “0” เอาต์พุตของไอซีดีโค้ดเดอร์ ให้เป็น “1” ตามลักษณะของคอมมอนอาโนด

การใช้งานโมดูลแสดงผลตัวเลขเจ็ดส่วน สามารถเลือกใช้งานแบบคอมมอนคาโถดหรือแบบคอมมอนอาโนด หรือใช้พร้อมกันก็ได้ โดยแบบคอมมอนคาโถดจะมี LED ติดเป็นสีแดง ส่วนคอมมอนอาโนดจะมี LED ติดเป็นสีเขียว ตัวโมดูลมีลักษณะดังรูปที่ 3.20



รูปที่ 3.20 แผนผังอุปกรณ์ของโมดูลแสดงผลตัวเลขเจ็ดส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.21 โมดูลแสดงผลตัวเลข 7 ส่วน

3.2.6 โมดูลดิพสวิตช์และเมตริกซ์สวิตช์ (Dip Switch and Matrix Switch Module)

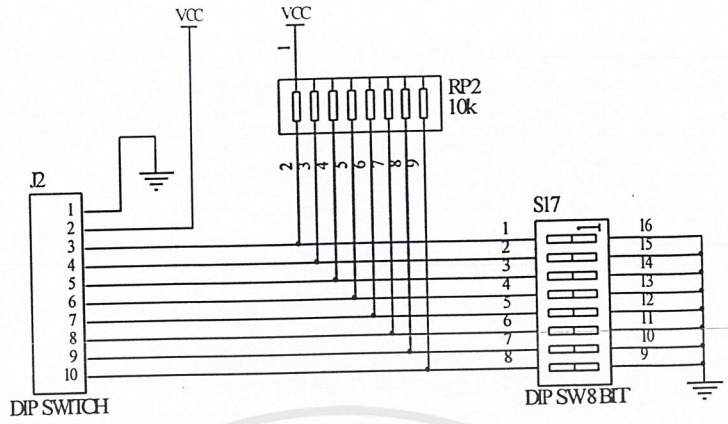
โมดูลดิพสวิตช์และเมตริกซ์สวิตช์ประกอบด้วย ชุดสร้างสัญญาณอินพุตที่เป็นสวิตช์ 2 แบบ คือ ดิพสวิตช์จำนวน 8 บิต และ เมตริกซ์สวิตช์ขนาด 4 x 4 โดยแต่ละแบบมีรายละเอียดดังนี้

1) ดิพสวิตช์

ใช้สำหรับสร้างสัญญาณลอจิกให้กับไมโครคอนโทรเลอร์ โดยสวิตช์จะทำงานในลักษณะ "กดติดกดดับ" เพื่อสร้างระดับลอจิกที่ค้างสภาวะเดิมไว้จนกว่าจะมีการเปลี่ยนสภาวะใหม่อีกครั้ง ประกอบไปด้วยดิพสวิตช์จำนวน 8 บิต มีลักษณะวงจรดังรูปที่ 3.22

วงจรดิพสวิตช์จะมีลักษณะคล้ายวงจรชุดสวิตช์พื้นฐาน โดยจะมีตัวต้านทานต่อพูลอัพต่อเพื่อให้ลอจิกปกติเป็น "1" เมื่อกดดิพสวิตช์เปลี่ยนสภาวะ ก็จะอ่านลอจิกได้เป็น "0" และจะค้างสภาวะนั้นอยู่จนกว่าจะกดเปลี่ยนสภาวะใหม่อีกครั้ง ซึ่งเป็นคุณสมบัติของตัวอุปกรณ์ดิพสวิตช์

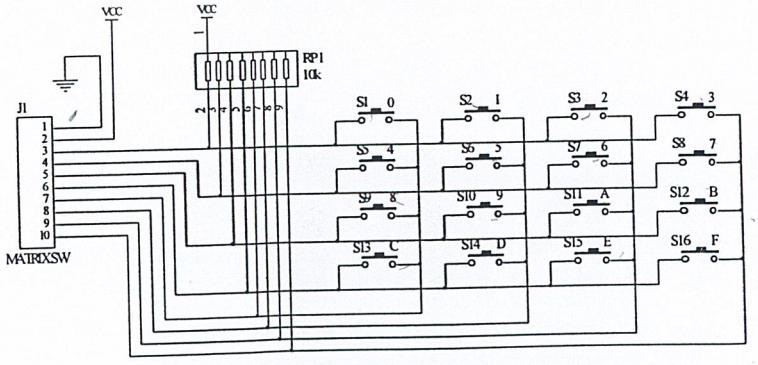
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.22 วงจรดิพสวิตช์ขนาด 8 บิต

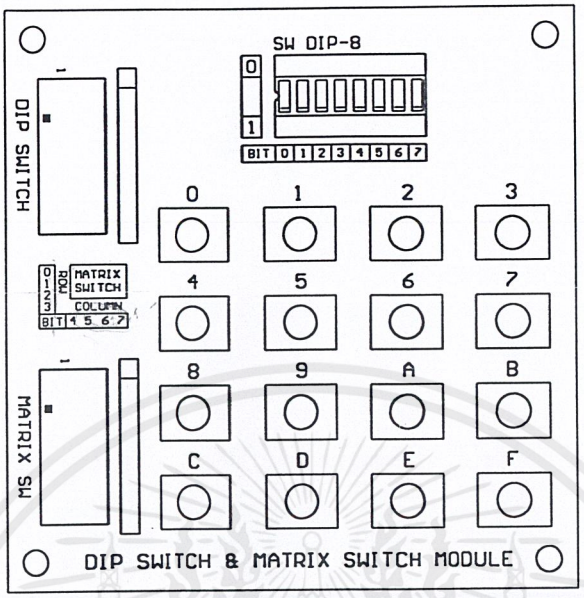
2) เมตริกซ์สวิตช์

ใช้เมื่อต้องการต่อสวิตช์จำนวนมากๆ เข้ากับไมโครคอนโทรลเลอร์ ซึ่งเป็นเมตริกซ์สวิตช์ขนาด 4 แถวคูณ 4 หลัก จึงมีสวิตช์จำนวน 16 ตัว ใช้พอร์ตขนาด 8 บิต ในการติดต่อเพื่ออ่านข้อมูลลอจิก โดยจะเป็นพอร์ตเอาต์พุตจำนวน 4 บิต เพื่อส่งข้อมูลเลือกหลักของเมตริกซ์สวิตช์ที่จะอ่านค่า และเป็นพอร์ตอินพุตจำนวน 4 บิต เพื่ออ่านระดับลอจิกจากสวิตช์ในแต่ละหลักจำนวน 4 ตัว เรียกว่า 4 แถว ทุกบิตจะมีตัวต้านทานต่อพูลอัปอยู่ การใช้งานจะแบ่งบิตออกเป็น 4 บิต 2 ชุด สามารถที่จะให้ชุดใดเป็นตัวเลือกสวิตช์และอีกชุดเป็นตัวอ่านสภาวะก็ได้ เวลาใช้งานจะต่อเข้ากับพอร์ต ขนาด 8 บิต ของไมโครคอนโทรลเลอร์และต้องกำหนดให้ 4 บิตบน หรือ 4 บิตล่าง ชุดใดชุดหนึ่งเป็นอินพุตและอีกชุดที่เหลือเป็นเอาต์พุต เมื่อต้องการเลือกหลักใดก็ให้ส่งลอจิก “0” มายังหลักนั้น และอ่านค่าจาก 4 บิตที่เหลือ บิตที่อ่านได้ “0” แสดงว่ามีสวิตช์ในแถวนั้นถูกกดอยู่ วงจรเมตริกซ์สวิตช์แสดงดังรูปที่ 3.23

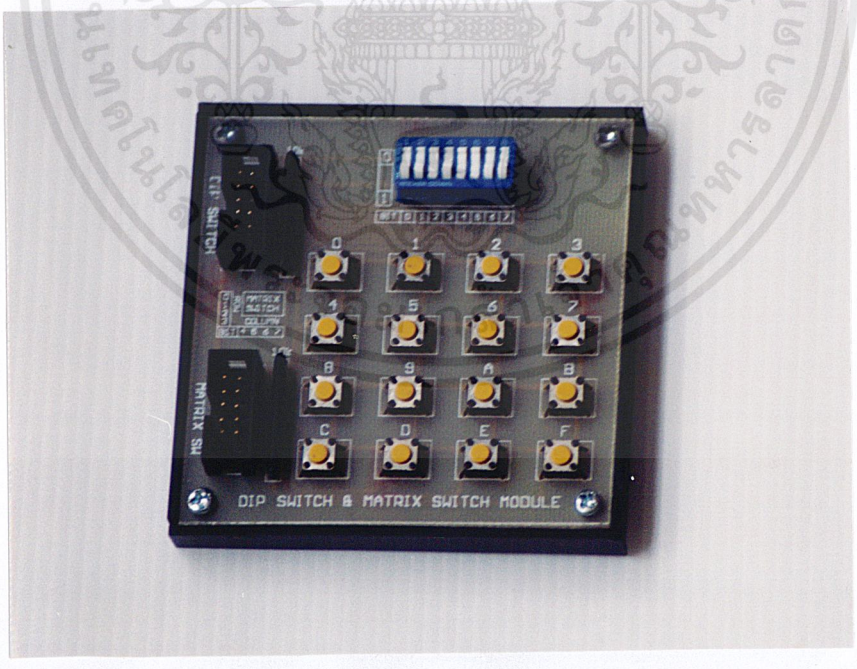


รูปที่ 3.23 วงจรเมตริกซ์สวิตช์ขนาด 4x4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.24 แผนผังอุปกรณ์โมดูลดิพสวิตช์และเมตริกซ์สวิตช์

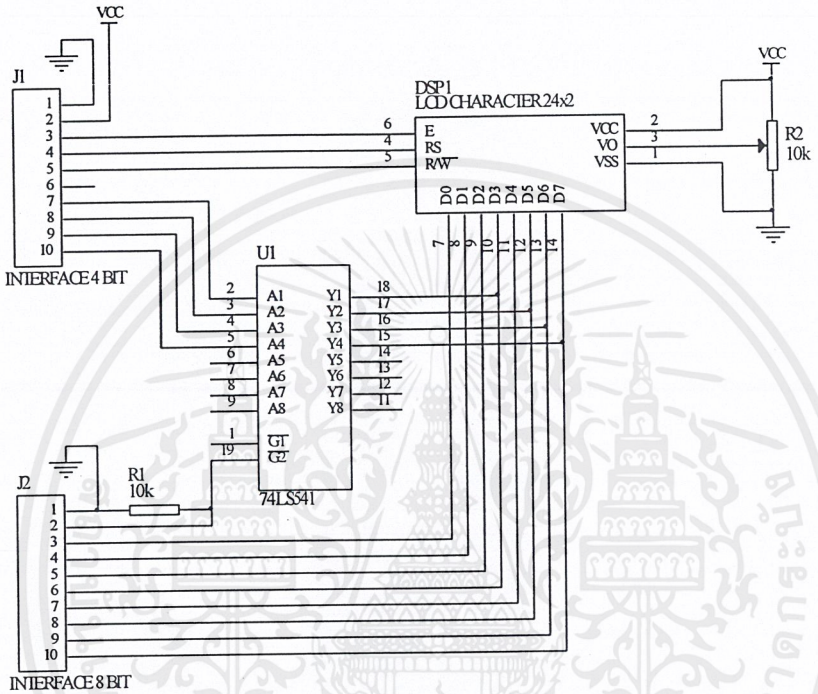


รูปที่ 3.25 โมดูลเมตริกซ์สวิตช์ขนาด 4x4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.7 โมดูลแสดงผลแบบผลึกเหลว (LCD Character Display Module)

โมดูลแสดงผลแอลซีดีแบบตัวอักษรขนาด 24 x 2 (24 ตัวอักษร 2 บรรทัด) สามารถต่อใช้งานได้ทั้งแบบ 4 บิต และแบบ 8 บิต วงจรมีลักษณะดังรูปที่ 3.26



รูปที่ 3.26 วงจรของโมดูลแสดงผลแบบผลึกเหลว

การใช้งานจอแสดงผลแอลซีดี มีมาตรฐานการต่อใช้งาน คือ มีขาสำหรับต่อใช้งานทั้งหมดจำนวน 14 เส้น แบ่งเป็นขาสัญญาณข้อมูล 8 เส้น ขาสัญญาณควบคุม 4 เส้น และเป็นขาสำหรับป้อนแรงดันอีก 2 เส้น มาตรฐานการใช้งานของแอลซีดีสามารถใช้งานได้ทั้งแบบ 4 บิต และแบบ 8 บิต ซึ่งจำนวนบิตนี้คือบิตของสัญญาณข้อมูล เมื่อต้องการใช้งาน 8 บิตก็จะใช้สัญญาณข้อมูลทั้ง 8 เส้น แต่เมื่อต้องการต่อใช้งานแบบ 4 บิตก็ใช้สัญญาณข้อมูล 4 เส้น คือ D4-D7 ส่วนขาที่ไม่ใช้งานก็ต่อลงกราวน์

จากวงจรรูปที่ 3.26 เมื่อต้องการใช้งานจอแสดงผลแอลซีดีแบบ 8 บิต สามารถต่อได้โดยใช้คอนเนคเตอร์จำนวน 2 พอร์ต ใช้เป็นพอร์ตควบคุม 1 พอร์ต ขนาด 3 บิต คือ บิตควบคุมขาอีนาเบิล (E) ขาอินพุตที่ใช้ในการแยกชนิดของข้อมูลว่าเป็นคำสั่ง หรือเป็นข้อมูล ถ้าเป็น “0” ข้อมูลที่เข้ามาจะเป็นคำสั่ง ถ้าเป็น “1” ข้อมูลที่ส่งมาเป็นข้อมูลสำหรับการแสดงผล (RS) และขาที่ใช้เลือกในการอ่านหรือเขียนข้อมูล คือถ้าเป็น “0” เป็นการกำหนดให้เขียน แต่ถ้าเป็น “1” จะเป็นการอ่านข้อมูล

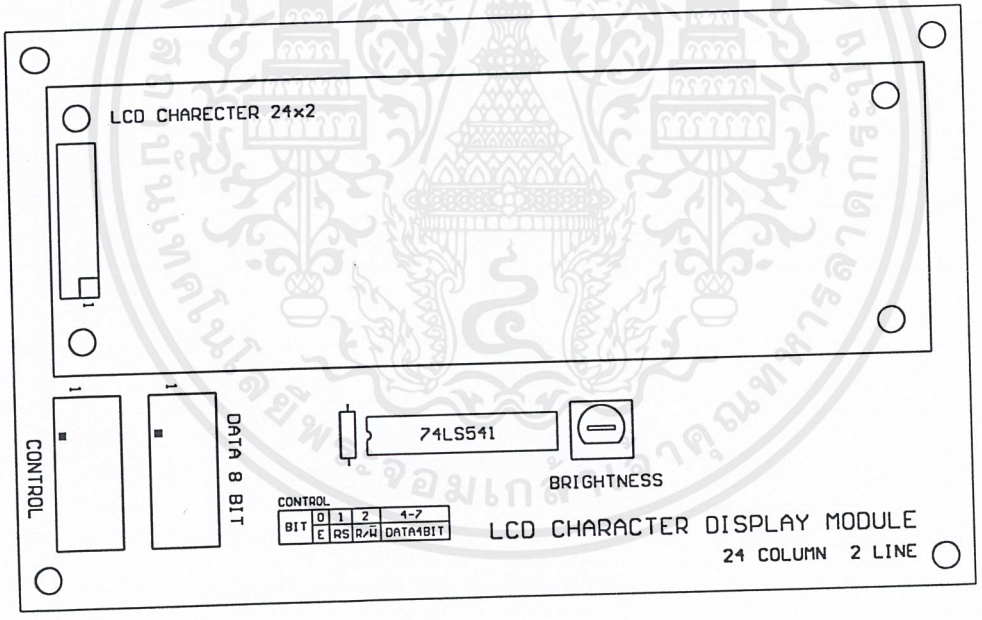
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(R/W) ของแอลซีดี และใช้เป็นพอร์ตสัญญาณข้อมูลอีก 1 พอร์ต ขนาด 8 บิต และเมื่อต้องการต่อใช้งานแบบ 4 บิต ทำโดยการถอดคอนเนคเตอร์ที่ใช้ส่งสัญญาณข้อมูล 8 บิตนั้นออก แต่จะใช้สัญญาณข้อมูลผ่านพอร์ตเดียวกับพอร์ตควบคุมรวมทั้งหมด 7 บิต

การทำงานของวงจรตัวเลือกว่าจะติดต่อแบบ 4 บิต หรือ 8 บิต จะใช้ไอซีเบอร์ 74LS541 ทำหน้าที่อีนาเบิลหรือดิสเอเบิลสัญญาณข้อมูลขนาด 4 บิต ที่อยู่ในพอร์ตควบคุม

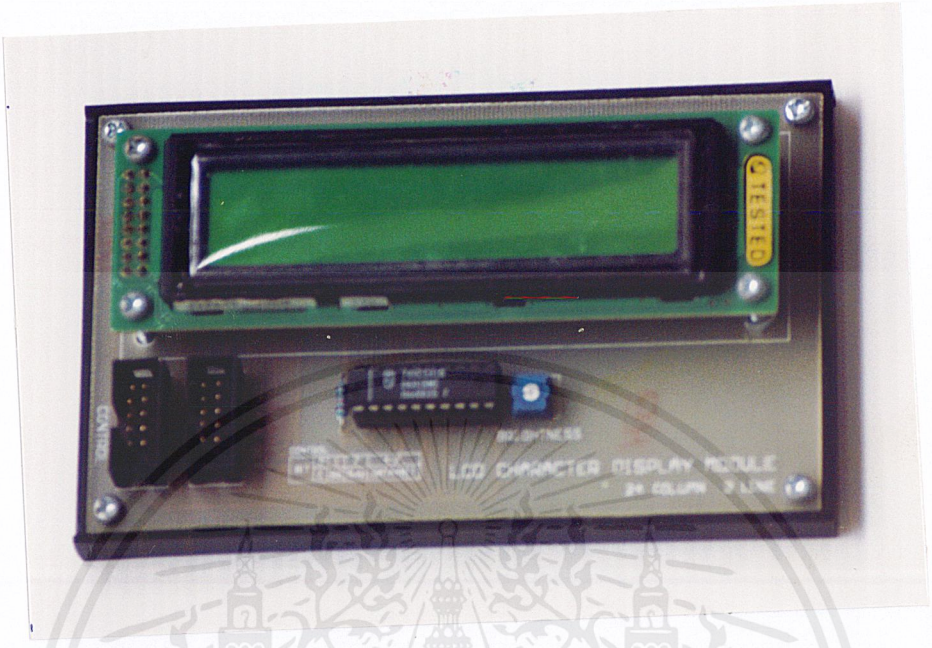
เมื่อต่อสัญญาณข้อมูล 8 บิต เข้าไป จะมีแรงดัน VCC จากคอนเนคเตอร์เข้าไปดิสเอเบิลตัวไอซี 74LS541 ที่ขา G1 และ G2 ทำให้ไอซีตัดสัญญาณจากพอร์ต 4 บิตออก แล้วนำสัญญาณข้อมูลจากพอร์ต 8 บิตไปใช้แทน จากนั้นก็สั่งงานการเชื่อมต่อจากไมโครคอนโทรลเลอร์

ตัวต้านทานปรับค่า R2 สำหรับปรับความสว่าง (Brightness) ให้กับแอลซีดี โดยปรับเมื่อปรับค่า R2 มาทางซ้ายสุดจะทำให้จอแอลซีดีสว่างที่สุด และเมื่อปรับค่า R2 มาทางขวามือสุดก็จะทำให้จอแอลซีดีสว่างน้อยที่สุด



รูปที่ 3.27 แผนผังอุปกรณ์ของโมดูลแสดงผลแบบผลึกเหลว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.28 วงจรของโมดูลแสดงผลแบบผลึกเหลว

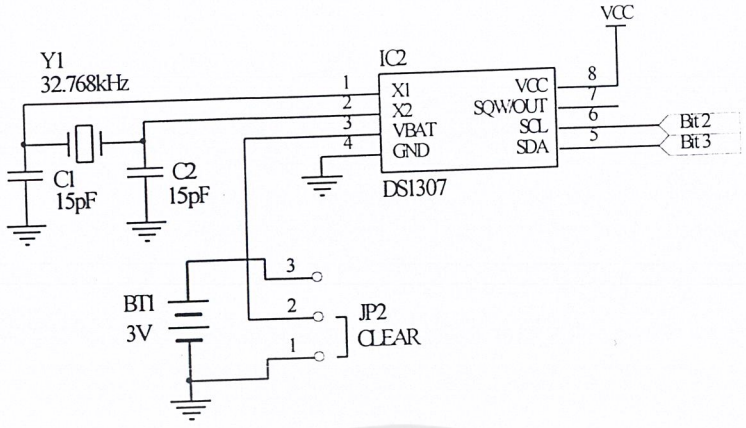
3.2.8 โมดูลเชื่อมต่ออุปกรณ์ RTC / EEPROM / SPEAKER (RTC / EEPROM / Speaker Interfacing Module)

โมดูลเชื่อมต่ออุปกรณ์เพื่อใช้ทำงานร่วมกับไมโครคอนโทรลเลอร์ ซึ่งโมดูลนี้ประกอบไปด้วยวงจร RTC วงจร EEPROM และวงจร SPEAKER มีรายละเอียดของแต่ละวงจรดังนี้

1) RTC : Real Time Clock

ไอซีเบอร์ DS1307 เป็นไอซีสำเร็จรูปที่ใช้บรรจุฐานเวลาจริงเพื่อป้อนฐานเวลาจริง ให้กับไมโครคอนโทรลเลอร์ นำไปใช้งาน ซึ่งไอซีเบอร์นี้ จะมีการติดต่อผ่านทางระบบ I^2C Bus ทำให้ประหยัดพอร์ตสำหรับต่อใช้งานลงได้ มีวงจรดังรูปที่ 3.29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

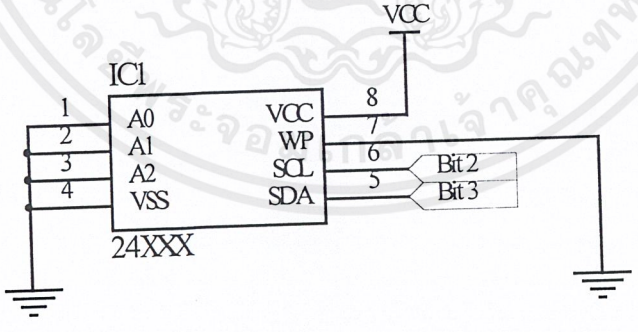


รูปที่ 3.29 วงจรติดต่อกับไอซีฐานเวลา DS1307

ไอซีเบอร์ DS1307 จะติดต่อกับไมโครคอนโทรลเลอร์ผ่านบิต 2 และบิต 3 ของพอร์ตที่นำมาต่อ ซึ่งเป็นพอร์ตใดก็ได้ JP2 มีไว้เพื่อเคลียร์ข้อมูลในไอซี เมื่อเวลาต้องการลบล้างฐานเวลาไปเริ่มต้นใหม่ โดยจะทำการปลดถ่านแบคอัพ (Battery Back Up) ออก

2) EEPROM 24XXX

เป็นการเชื่อมต่อกับหน่วยความจำชนิด EEPROM (ในตัวไมโครคอนโทรลเลอร์จะมี EEPROM อยู่ภายใน) เมื่อต้องการเพิ่มหน่วยความจำภายนอก โดยจะใช้หน่วยความจำตระกูล 24XXX ซึ่งจะเชื่อมต่อกับไมโครคอนโทรลเลอร์ผ่าน I^2C Bus เช่นกัน มีวงจรดังรูปที่ 3.30

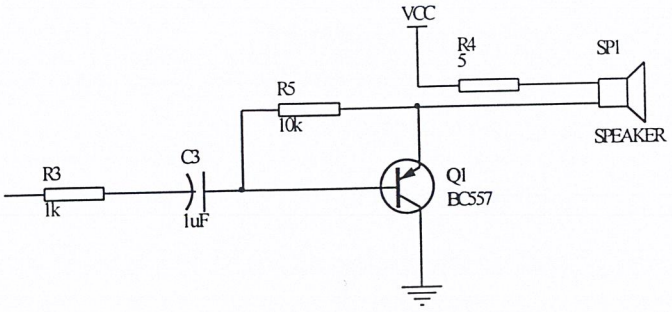


รูปที่ 3.30 วงจรติดต่อกับ EEROM ตระกูล 24XXX

3) SPEAKER

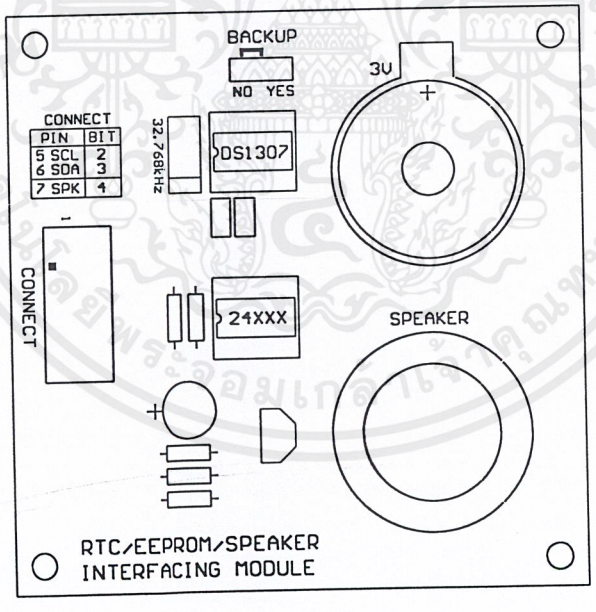
เป็นการเชื่อมต่อไมโครคอนโทรลเลอร์เข้ากับลำโพงเปล่งเสียง โดยเสียงที่จะได้จะเกิดจากพัลซ์ที่ส่งออกมาจากไมโครคอนโทรลเลอร์ วงจรเป็นดังรูปที่ 3.31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.31 วงจรเชื่อมต่อลำโพง

สัญญาณที่ได้จากไมโครคอนโทรลเลอร์จะไปควบคุมทรานซิสเตอร์ Q1 เป็นตัวเปิดปิดสัญญาณความถี่ที่ลำโพง โดยจั๊มเปอร์ JP1 เป็นตัวอีนาเบิลการใช้งานลำโพงนี้ ตัวโมดูลเชื่อมต่ออุปกรณ์ดังกล่าวมีจุดต่อใช้งานดังรูปที่ 3.32



รูปที่ 3.32 แผนผังอุปกรณ์ของโมดูลเชื่อมต่ออุปกรณ์ RTC/EEPROM/SPEAKER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.33 โมดูลเชื่อมต่ออุปกรณ์ RTC / EEPROM / SPEAKER

3.2.9 โมดูลแปลงระหว่างสัญญาณแอนะลอกและสัญญาณดิจิตอล (Analog and Digital Converter Module)

เป็นโมดูลสำหรับใช้ในการทดลองเรื่องการแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอล และการแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะลอก โดยใช้ไมโครคอนโทรเลอร์ PIC16F8XX เป็นตัวควบคุมการทำงาน

1) การแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอล

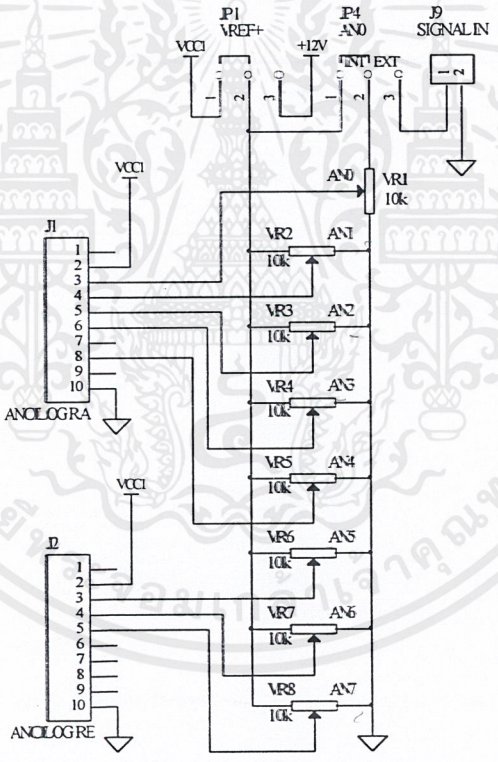
จากคุณสมบัติพิเศษของไมโครคอนโทรลเลอร์ตระกูล PIC16F8XX ที่มีอุปกรณ์แปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอลในตัวไมโครคอนโทรลเลอร์ โมดูลนี้จึงมีวงจรที่ใช้สร้างสัญญาณแอนะลอก ที่เป็นแรงดันปรับค่าได้จาก 0 ถึง 5 โวลต์ ต่อเข้าพอร์ต A จำนวน 8 ช่องสัญญาณ ซึ่งมีวงจรดังรูปที่ 3.34

จากรูปที่ 3.34 วงจรภายในของวงจรสร้างสัญญาณแอนะลอก จะใช้ตัวต้านทานปรับค่าได้ต่อเป็นวงจรแบ่งแรงดัน ขนาด 10 กิโลโอห์ม จะได้แรงดันเอาต์พุตที่เป็นสัญญาณแอนะลอกจากค่าแรงดัน 0 ถึงแรงดันอ้างอิงที่สามารถเลือกได้ว่าจะใช้แรงดันจาก VCC หรือใช้แรงดันจากภายนอก (12 โวลต์) โดยเลือกที่จั๊มเปอร์ JP1 โดยปกติจะเลือกไว้ที่ VCC (5 โวลต์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้ยังสามารถป้อนสัญญาณแอนะลอกจากภายนอก เช่น สัญญาณไซน์เวฟ (Sine Wave) หรือสัญญาณเสียง เข้ามายังวงจรเพื่อป้อนให้กับไมโครคอนโทรเลอร์นำไปแปลงเป็นสัญญาณดิจิตอลได้ โดยจะป้อนสัญญาณแอนะลอกเข้ามาทางคอนเนคเตอร์ J9 แล้วใช้จั๊มเปอร์ JP4 เลือกไปที่ EXT โดยสัญญาณแอนะลอกนี้จะถูกส่งให้ไมโครคอนโทรลเลอร์ผ่านทางพอร์ต RA0 ซึ่งสัญญาณแอนะลอกนี้จะยังคงผ่านตัวต้านทานปรับค่าได้ 10 กิโลโอห์ม อยู่เพื่อใช้ปรับขนาดความแรงของสัญญาณจากภายนอกหากมีความแรงมากเกินไป

ไมโครคอนโทรลเลอร์ตระกูล PIC16F8XX มีพอร์ตต่อสัญญาณแอนะลอกอยู่ที่พอร์ต RA (ตระกูล 28 และ 40 ขา) และพอร์ต RE (ตระกูล 40 ขา อย่างเดียว) ในตัวโมดูลนี้จะมีคอนเนคเตอร์สำหรับใช้ต่อกับพอร์ตดังกล่าวโดยเฉพาะ ชื่อว่า ANALOG RA และ ANALOG RE



รูปที่ 3.34 วงจรสร้างสัญญาณแอนะลอก

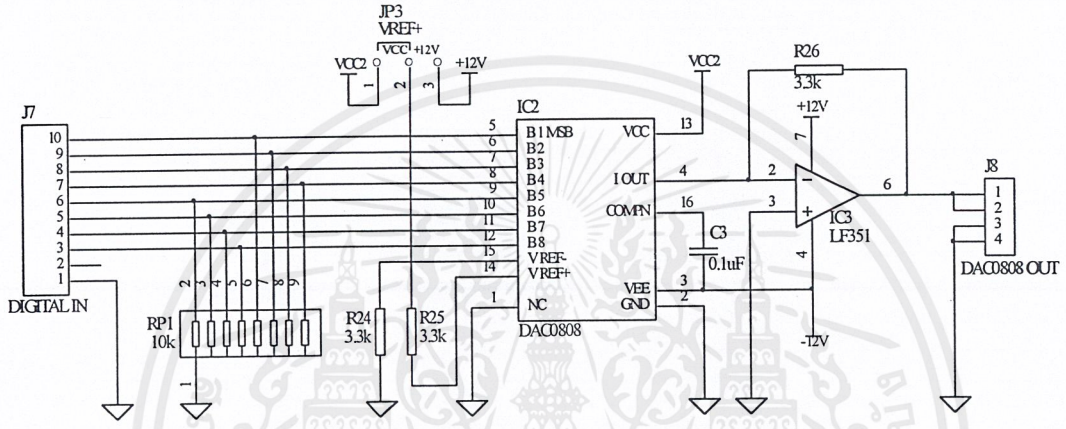
2) การแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะลอก

ในตัวโมดูลจะมีวงจรเพื่อแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะลอกอยู่ 2 แบบ คือ แบบใช้ไอซี DAC0808 ขนาด 8 บิต และ แบบ R-2R Ladder ขนาด 10 บิต ซึ่งการแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะลอกแต่ละแบบมีรายละเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1) แบบใช้ไอซี DAC0808 ขนาด 8 บิต

จากวงจรรูปที่ 3.35 แบบใช้ไอซี DAC0808 สามารถเลือกแรงดันอ้างอิงแอนะลอกได้ว่าจะใช้แรงดันจาก VCC หรือแรงดันจากภายนอก (12 โวลต์) โดยเลือกที่จั๊มเปอร์ JP3 สัญญาณดิจิตอลขนาด 8 บิต จะส่งเข้ามาทางคอนเนคเตอร์ชื่อ DIGITAL IN และมีตัวต้านทานพูลดาวน์ต่ออยู่ก่อนจะเข้าไปยังไอซี



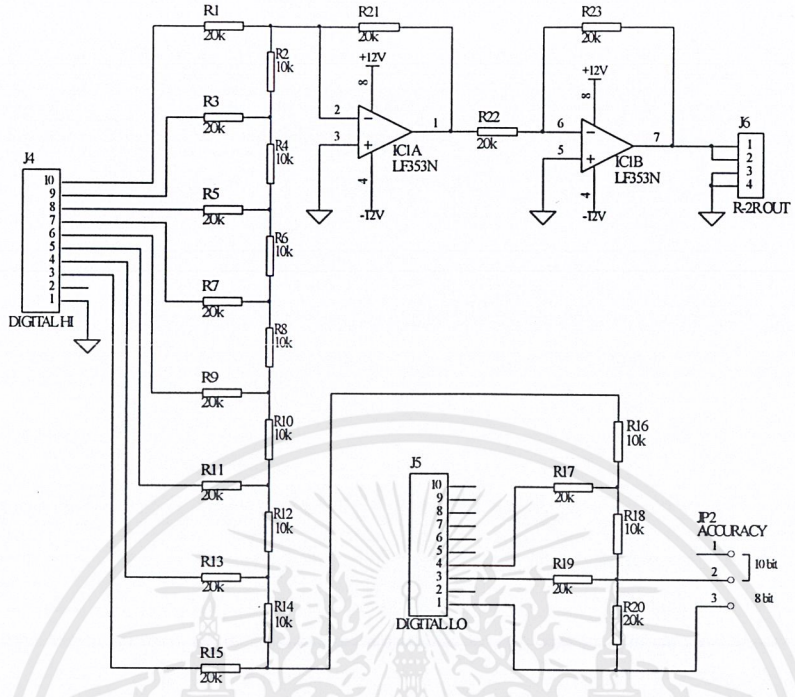
รูปที่ 3.35 วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะลอกแบบใช้ไอซี DAC0808

ด้านเอาต์พุตของไอซีจะมีออปแอมป์ต่อเป็นวงจรอินเวอร์ติ้ง แอมปลิไฟล์เออร์ (Inverting Amplifier) ทำหน้าที่กลับเฟสของสัญญาณและชดเชยเอาต์พุตอิมพิแดนซ์ของไอซี DAC0808 ทำให้วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะลอกมีเอาต์พุตอิมพิแดนซ์ของไอซี DAC0808 ต่ำลง และสัญญาณเอาต์พุตจะออกทางคอนเนคเตอร์ที่มีชื่อว่า DAC0808 OUT

2.2) แบบ R-2R Ladder ขนาด 10 บิต

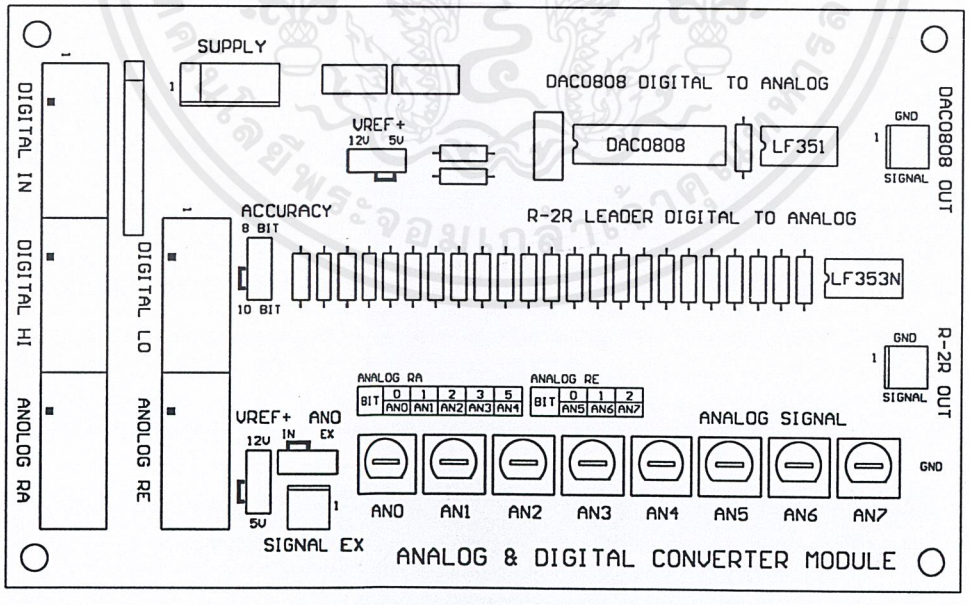
ในวงจรสามารถเลือกที่จะแปลงสัญญาณดิจิตอลขนาด 10 บิต หรือ 8 บิต ให้เป็นสัญญาณแอนะลอกได้ โดยเลือกที่จั๊มเปอร์ JP2 การต่อสัญญาณดิจิตอลเข้ามาผ่านคอนเนคเตอร์ 2 ตัว หากใช้ 8 บิต จะใช้เฉพาะคอนเนคเตอร์ชื่อ DIGITAL HI แต่ถ้าใช้แบบ 10 บิต จะใช้คอนเนคเตอร์ชื่อ DIGITAL LO ด้วย ดังวงจรในรูปที่ 3.36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



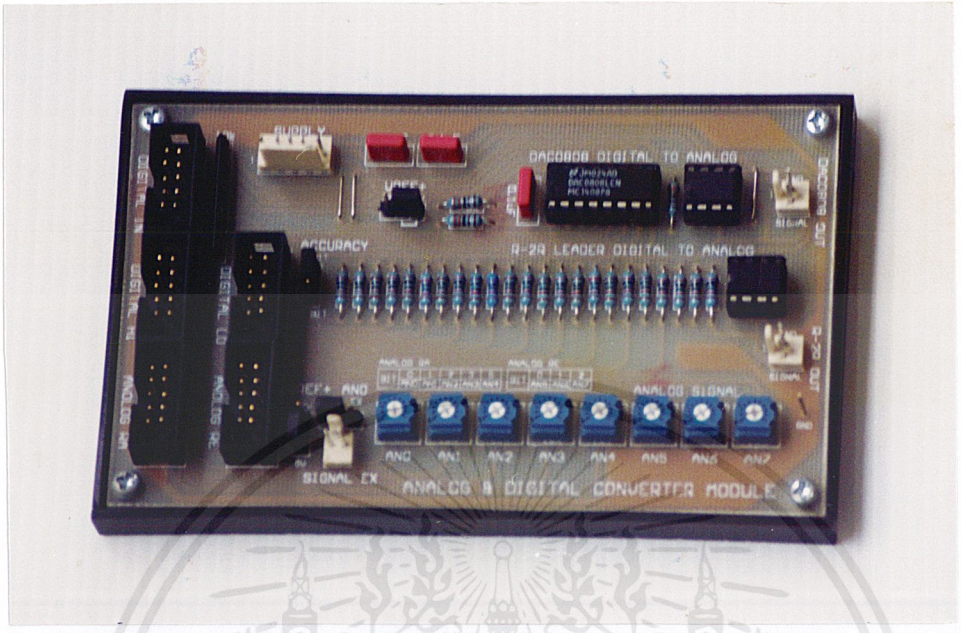
รูปที่ 3.36 วงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอกแบบ R-2R แลดเดอร์

โมดูลแปลงระหว่างสัญญาณแอนะลอกและสัญญาณดิจิตอล มีลักษณะดังรูปที่ 3.37



รูปที่ 3.37 แผนผังอุปกรณ์ของโมดูลแปลงระหว่างสัญญาณแอนะลอกและสัญญาณดิจิตอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.38 โมดูลแปลงระหว่างสัญญาณแอนะลอกและสัญญาณดิจิตอล

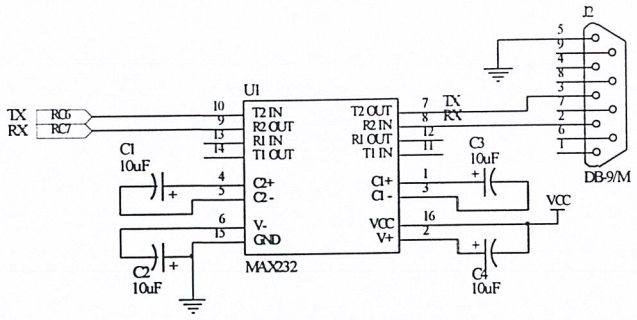
3.2.10 โมดูลสื่อสารข้อมูลอนุกรม (Serial Interfacing Module)

จากคุณสมบัติพิเศษของไมโครคอนโทรลเลอร์ตระกูล PIC16F8XX สามารถเชื่อมต่ออุปกรณ์สื่อสารในรูปแบบสัญญาณอนุกรมได้โดยผ่าน USART (Universal Synchronous Asynchronous Receiver Transmitter) ในตัวไมโครคอนโทรลเลอร์ โมดูลนี้จะมีชุดอุปกรณ์สื่อสารแบบอนุกรมให้ใช้ในมาตรฐาน RS-232 RS-485 และ RS-422 มีรายละเอียดมาตรฐานต่างๆ ดังนี้

1) RS-232 Interface

ใช้ไอซี MAX232 ในการแปลงสัญญาณจากไมโครคอนโทรลเลอร์ไปสู่มาตรฐาน RS-232 จะติดต่อกับพอร์ต RC6 (TX) และ RC7 (RX) ของไมโครคอนโทรลเลอร์ ด้านจุดเชื่อมต่อจะผ่านคอนเนคเตอร์ DB-9 ชนิด Male มีลักษณะเหมือนกับจุดเชื่อมต่อกับคอมพิวเตอร์ จึงสามารถใช้สายสัญญาณจากคอมพิวเตอร์ในการเชื่อมต่อได้ การใช้งาน RS-232 จะต้องเลือกจั๊มเปอร์ชื่อ RS-232 ไปที่ ON ก่อน โดยจั๊มเปอร์ตัวนี้จะใช้ตัดต่อสัญญาณระหว่างตัวไอซี MAX232 กับพอร์ต RC7 ของไมโครคอนโทรลเลอร์ มีวงจรใช้งานดังรูปที่ 3.39

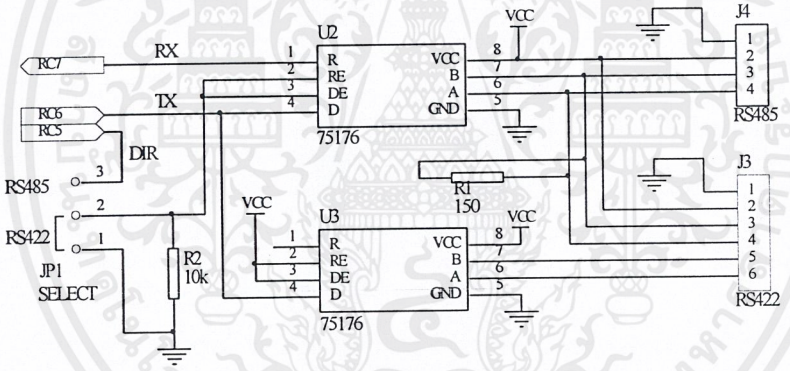
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.39 วงจรสื่อสารข้อมูลอนุกรมมาตรฐาน RS-232

2) RS-485 / RS-422 Interface

ในมาตรฐานนี้จะใช้ไอซีเบอร์ SN75176 ในการเชื่อมต่อ โดยในมาตรฐาน RS-485 นั้นจะเป็นการเชื่อมต่อแบบ Half Duplex ส่วน RS-422 จะเป็นแบบ Full Duplex มีวงจรดังรูปที่ 3.40

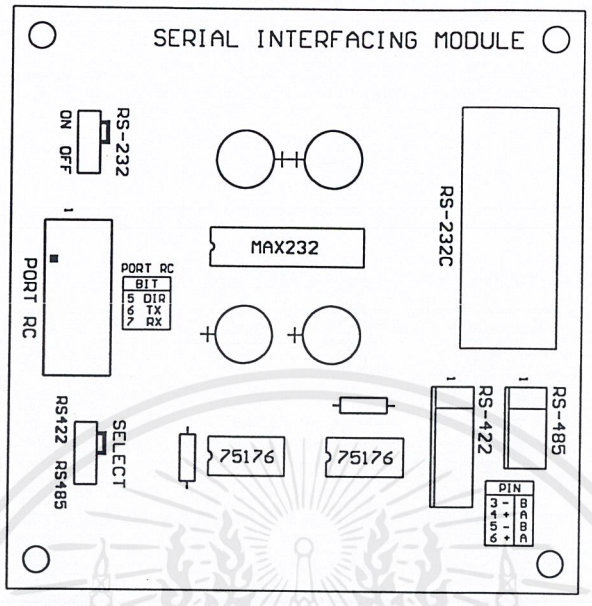


รูปที่ 3.40 วงจรสื่อสารข้อมูลอนุกรมมาตรฐาน RS-485 และ RS-422

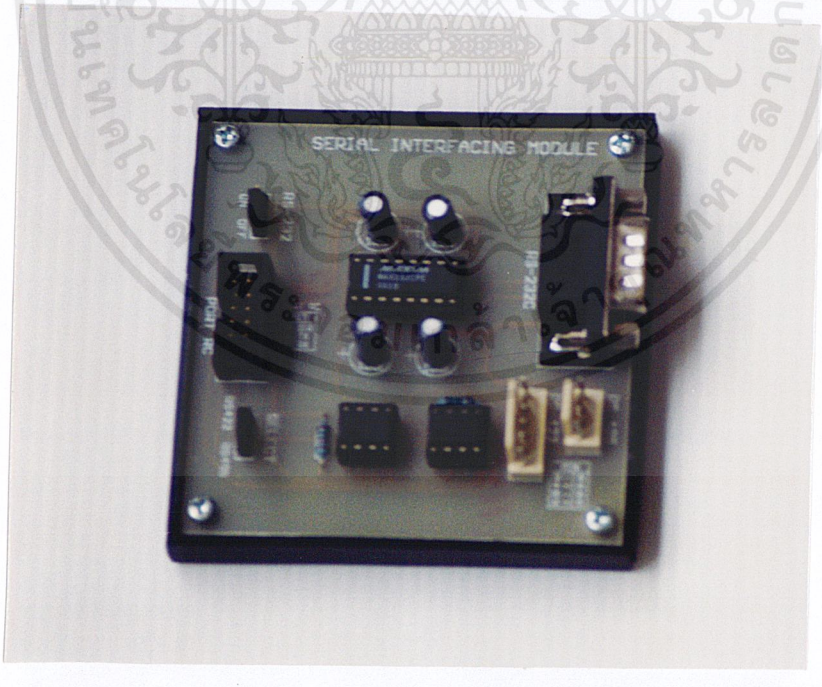
เมื่อใช้งานมาตรฐาน RS-485 หรือ RS-422 จะต้องทำการ OFF ที่จั๊มเปอร์ RS-232 ก่อน จากนั้นก็ทำการเลือกที่จั๊มเปอร์ SELECT ว่าจะใช้มาตรฐานใดระหว่าง RS-485 หรือ RS-422

มาตรฐาน RS-485 ใช้ไอซี U2 เป็น Line Driver ในการรับส่งข้อมูลอนุกรมที่คอนเนคเตอร์ J3 ขา 2 และ 3 ทำงานเป็น Half Duplex สามารถเลือกว่าจะส่งหรือจะรับโดยเลือกจากพอร์ต RC5 ของไมโครคอนโทรลเลอร์ ถ้าจะส่งข้อมูลให้เซตพอร์ต RC5 ถ้าจะรับก็ให้เคลียร์พอร์ต RC5 ส่วนมาตรฐาน RS-422 จะใช้ทั้ง U2 และ U3 ซึ่งทำงานเป็น Half Duplex ทั้งสองตัว U3 เป็นตัวส่ง และ U2 จะเป็นตัวรับ จะเป็น Full Duplex ไปจุดต่อใช้งานของโมดูลจะอยู่ในตำแหน่งต่างๆ ดังรูปที่ 3.41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.41 แผนผังอุปกรณ์ของโมดูลสื่อสารข้อมูลอนุกรม



รูปที่ 3.42 โมดูลสื่อสารข้อมูลอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

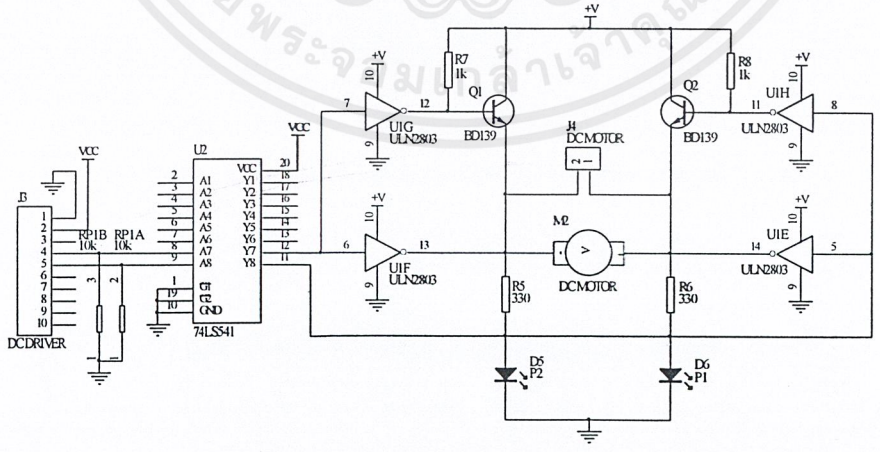
3.2.11 โมดูลขับดีซีมอเตอร์และสเตปปิ้งมอเตอร์ (DC Motor and Stepping Motor Driver Module)

เป็นโมดูลขับมอเตอร์ที่มีทั้งวงจรขับดีซีมอเตอร์และวงจรขับสเตปปิ้งมอเตอร์ โดยโมดูลนี้จะมีรายละเอียดดังนี้

1) วงจรขับดีซีมอเตอร์

ใช้ขับดีซีมอเตอร์ขนาดเล็ก สามารถเลือกแรงดันได้ทั้งจากVCC (5 โวลต์) หรือแรงดันจากภายนอก (12 โวลต์) มีคอนเนคเตอร์สำหรับต่อใช้งานที่บิต 1 และ บิต 2 ของ PORT RC เนื่องจากรองรับการใช้ PWM (Pulse Width Modulation) ในการควบคุมความเร็วมอเตอร์ ซึ่งเป็นคุณสมบัติพิเศษของไมโครคอนโทรลเลอร์ตระกูล PIC16F8XX ก็ได้ มีวงจรดังรูปที่ 3.43

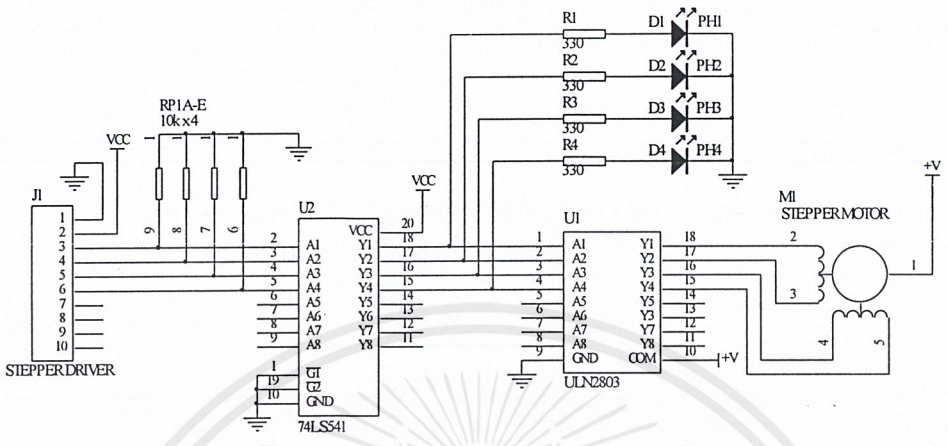
วงจรจะใช้ไอซีขับกระแสสูงเบอร์ ULN2803 ต่อใช้งานในรูปวงจรพุช-พูล (Push Pull) สามารถเปลี่ยนทิศทางการหมุนของมอเตอร์ได้ เช่น เมื่อต้องการให้มอเตอร์หมุนทวนเข็มนาฬิกาต้องส่งลอจิก “10” ให้กับ Port RC1 และ RC2 ตามลำดับ และถ้าจะให้มอเตอร์หมุนกลับทิศทางส่งลอจิก “01” ส่วนการส่งลอจิก “00” และ “11” จะเป็นการสั่งให้มอเตอร์หยุดหมุน ทำงานร่วมกับทรานซิสเตอร์เบอร์ BD139 ทำให้เป็นการประหยัดในการใช้อุปกรณ์ขับกระแสลงได้ เพราะตัวขับกระแสอีก 4 ตัวของไอซีจะถูกใช้ในการขับสเตปปิ้งมอเตอร์ในโมดูลเดียวกันนี้ ส่วนวงจรตัวขับกระแสสูงจะถูกด้านเอาต์พุตของวงจรจะต่อกับดีซีมอเตอร์ สามารถต่อได้จากคอนเนคเตอร์ J4 หรือ M2 ก็ได้ นอกจากนี้วงจรยังมี LED ชื่อ P1 และ P2 ที่ใช้บอกสถานะของแรงดันที่ส่งไปขับมอเตอร์ด้วยโดยถ้ามีแรงดันที่สายเส้นไฟใดก็จะทำให้ LED ที่เส้นนั้นติด



รูปที่ 3.43 วงจรขับดีซีมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

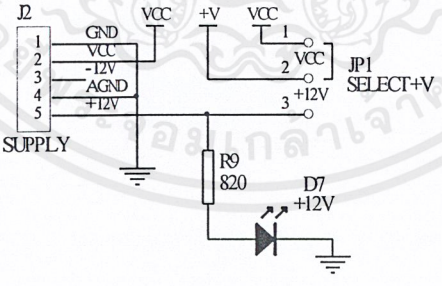
2) วงจรขับสเตปปิ้งมอเตอร์



รูปที่ 3.44 วงจรขับสเตปปิ้งมอเตอร์

จากวงจรดังรูปที่ 3.44 ใช้ขับสเตปปิ้งมอเตอร์แบบยูนิโพลาร์ จะใช้ไอซีเบอร์ ULN2803 ในการขับกระแส ซึ่งไอซีนี้จะใช้ร่วมกับวงจรขับดีซีมอเตอร์และในวงจรยังมี LED ที่ใช้บอกสถานะของสายไฟที่ใช้ขับมอเตอร์ทั้งสี่เฟสด้วย

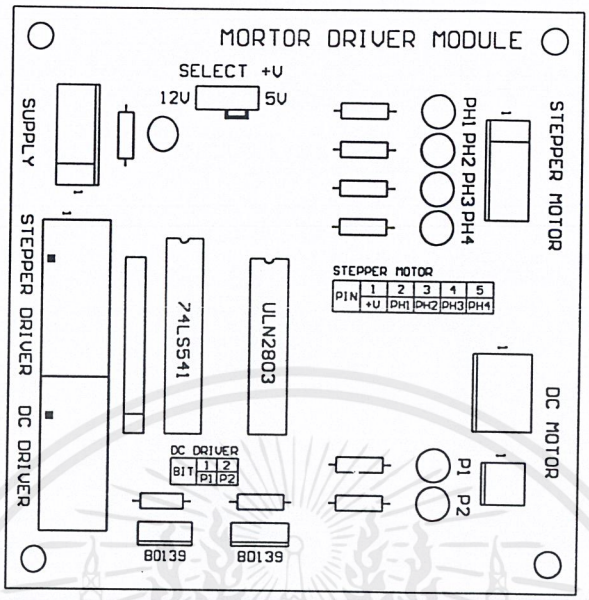
การเลือกแรงดันที่จะใช้ขับมอเตอร์ทั้งสองแบบสามารถเลือกได้ที่จั๊มเปอร์ชื่อ SELECT +V มีวงจรดังรูปที่ 3.45



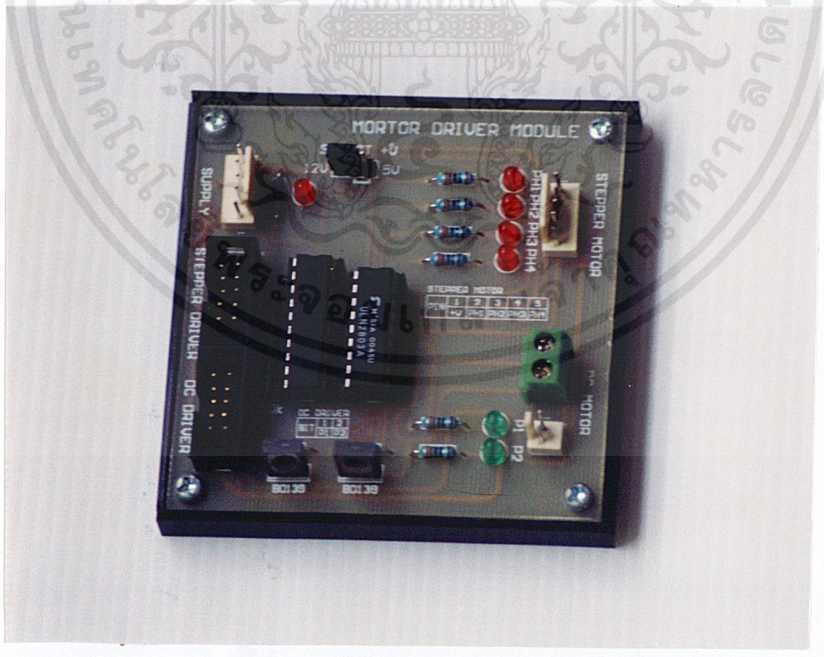
รูปที่ 3.45 จั๊มเปอร์ที่ใช้เลือกแรงดันในการขับมอเตอร์

คอนเนคเตอร์ชื่อ SUPPLY ใช้สำหรับป้อนแรงดัน +12 โวลต์ จากภายนอกเข้ามาเพื่อใช้ขับมอเตอร์ทั้งสองแบบ LED D7 ใช้แสดงแรงดันที่ป้อนเข้ามา เมื่อเลือกจั๊มเปอร์ที่แรงดัน 12 โวลต์ ทั้งดีซีมอเตอร์และสเตปปิ้งมอเตอร์ก็จะถูกขับด้วยแรงดัน 12 โวลต์ ทั้งคู่ โมดูลขับดีซีมอเตอร์และสเตปปิ้งมอเตอร์มีจุดต่อใช้งานต่างๆ ดังรูปที่ 3.46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.46 แผนผังอุปกรณ์ของโมดูลขับดีซีมอเตอร์และสเตปปิ้งมอเตอร์



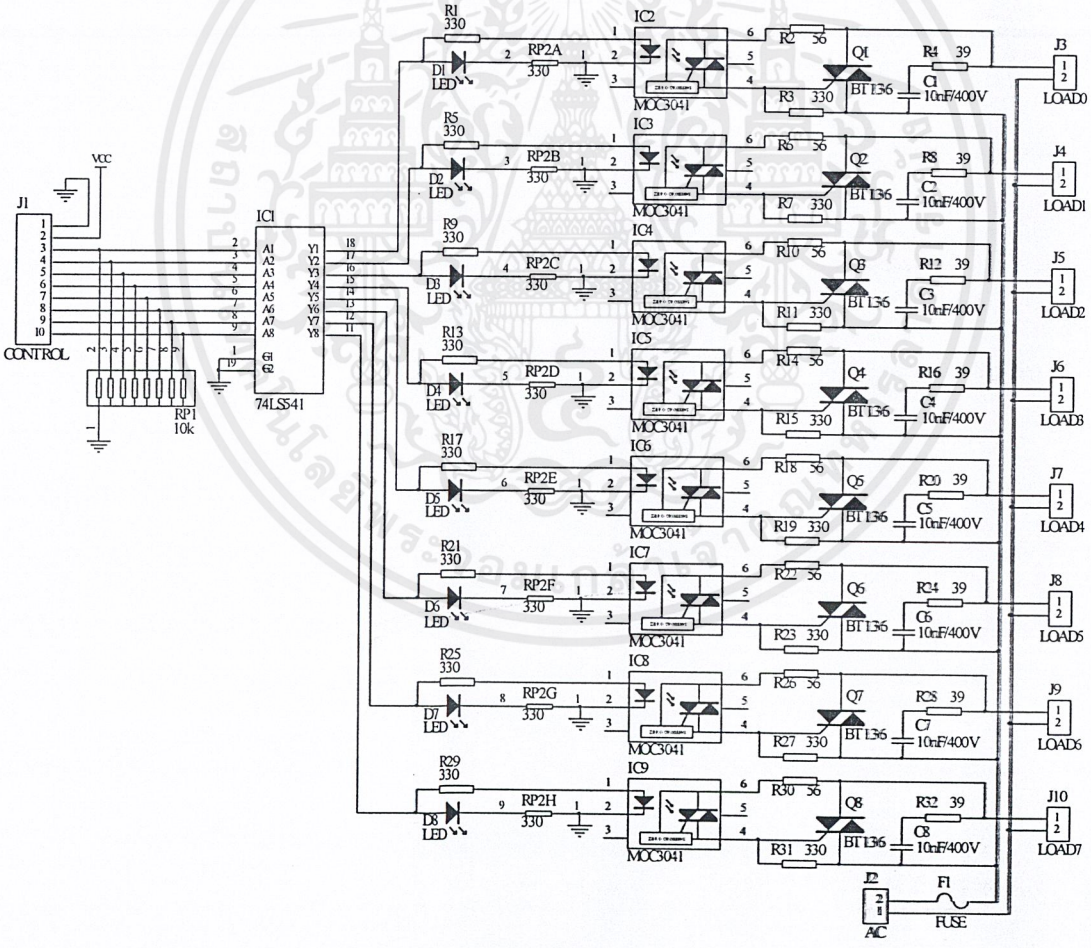
รูปที่ 3.47 โมดูลขับดีซีมอเตอร์และสเตปปิ้งมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.12 โมดูลโซลิดสเตทรีเลย์ (Solid State Relay Module)

โซลิดสเตทรีเลย์เป็นอุปกรณ์อิเล็กทรอนิกส์สำหรับเปิด-ปิด แรงดันไฟฟ้า 220 VAC ที่ปราศจากกลไกหน้าสัมผัส ใช้ในการควบคุมเครื่องใช้ไฟฟ้าต่างๆ ซึ่งในโมดูลนี้มีให้ใช้งาน 8 ช่อง มีวงจรดังรูปที่ 3.48

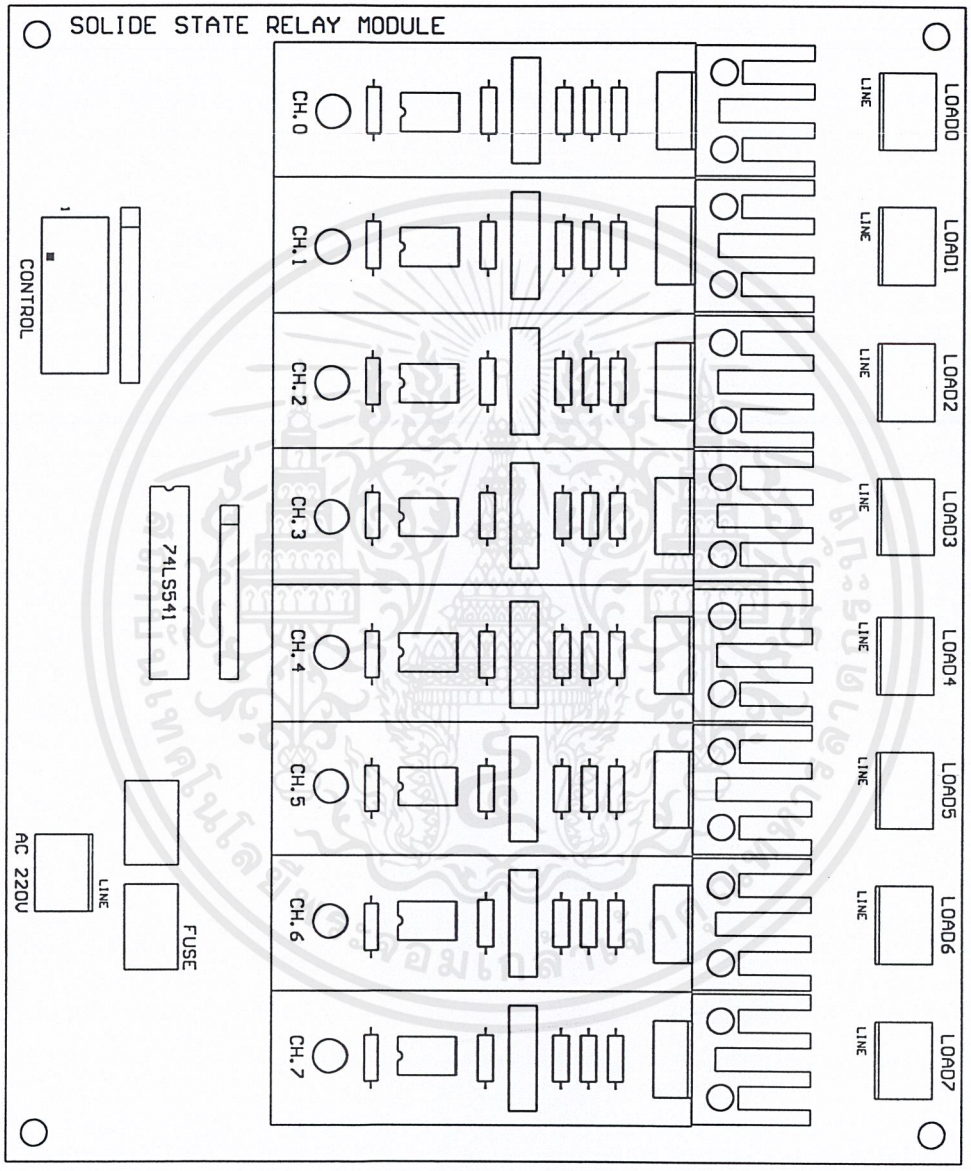
การควบคุมด้วยไมโครคอนโทรลเลอร์จะผ่านไอซีบัฟเฟอร์เบอร์ 74LS541 เพื่อขับอุปกรณ์เชื่อมโยงทางแสง (Op-to Coupler) เบอร์ MOC-3041 ทำให้สามารถแยกระบบกราวน์ทางไฟฟ้าของไมโครคอนโทรลเลอร์ออกจากโหลดที่มีแรงดันสูง เอาต์พุตเป็นโฟโต้ไตรแอก และมีวงจรตรวจจับแรงดันศูนย์อยู่ภายใน นำไปขับไตรแอกขนาดใหญ่อีกที เพื่อขับโหลดให้ได้กระแสมากขึ้น ลักษณะของโมดูลโซชิดสเตทรีเลย์ดังรูปที่ 3.49



รูปที่ 3.48 วงจรโซลิดสเตทรีเลย์ขนาด 8 ช่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมจะเชื่อมต่อผ่านคอนเนคเตอร์ชื่อ CONTROL การป้อนแรงดันไฟสูง 220 VAC จะป้อนเข้าที่ LINE AC จากนั้นต่อโหลดใช้งานได้ทั้ง 8 ช่อง มี LED แสดงสภาวะการเปิด-ปิด



รูปที่ 3.49 แผนผังอุปกรณ์ของโมดูลโซลิดสเตทรีเลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

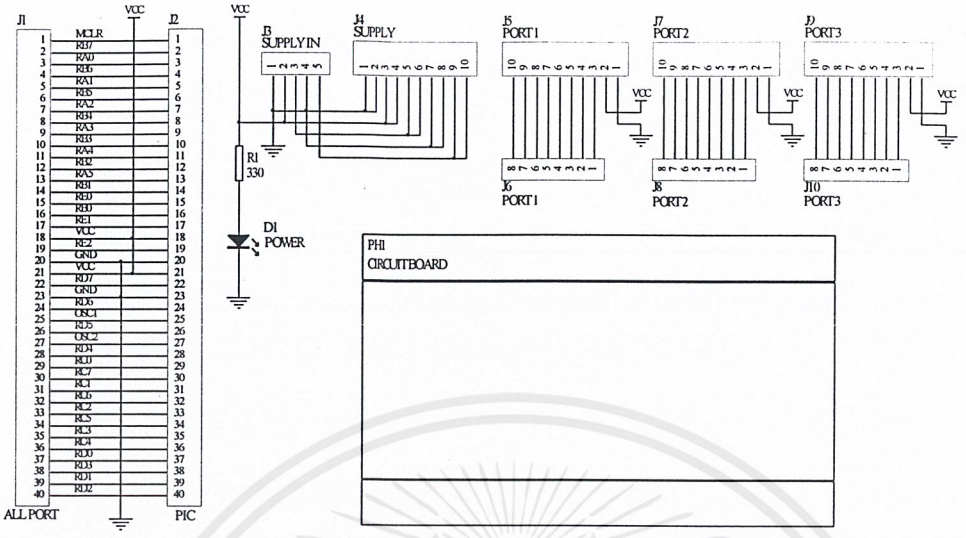


รูปที่ 3.50 โมดูลโซลิดสเตทรีเลย์

3.2.13 โมดูลทดลองวงจร (Circuit Experimental Module)

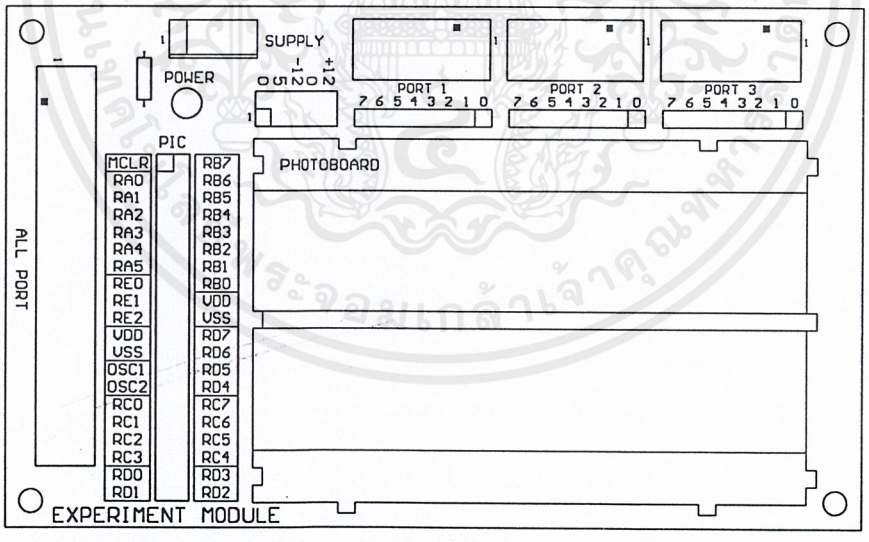
โมดูลทดลองวงจรเป็นโมดูลสำหรับใช้ทดลองต่อวงจรขนาดเล็กต่างๆ ในแผงทดลองที่มีขนาด 2" x 3" สามารถต่อใช้งานกับไมโครคอนโทรลเลอร์ PIC16F8XX ได้ทุกขาผ่านคอนเนคเตอร์ขนาด 40 ขา หรือใช้เชื่อมต่อกับโมดูลอื่นๆ ผ่านคอนเนคเตอร์ 10 ขา ก็ได้ มีวงจรดังรูปที่ 3.51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



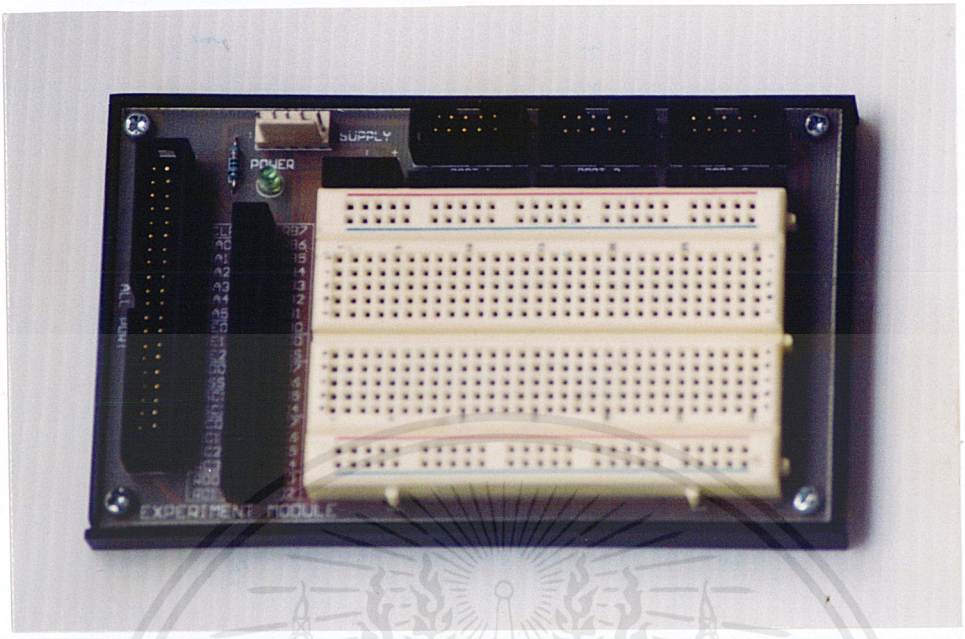
รูปที่ 3.51 วงจรในโมดูลทดลองวงจร

โมดูลทดลองวงจรสำหรับใช้ทดลองต่อวงจรขนาดเล็ก มีจุดต่อใช้งานต่างๆ ดังรูปที่ 3.52



รูปที่ 3.52 แผนผังอุปกรณ์ของโมดูลทดลองวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



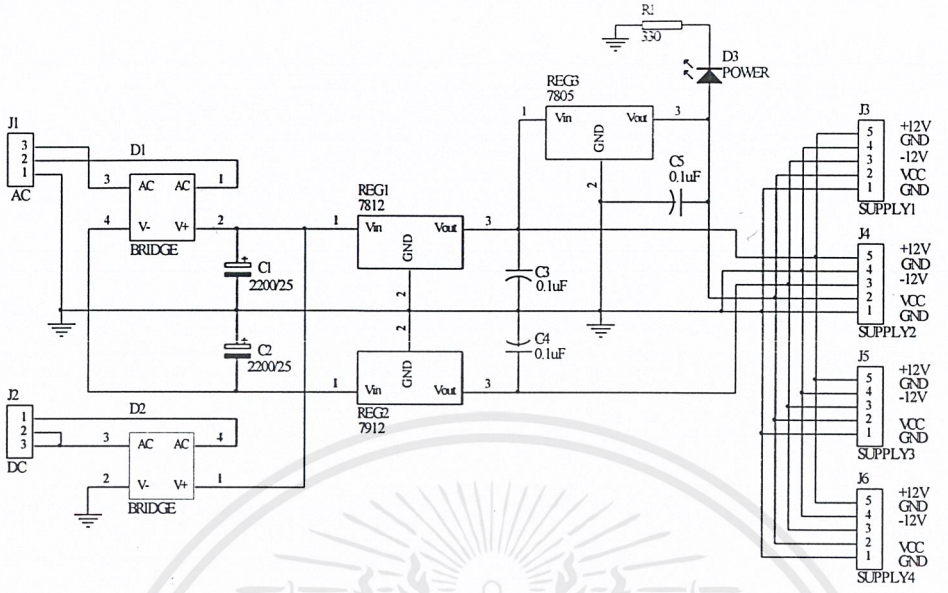
รูปที่ 3.53 โมดูลทดลองวงจร

3.2.14 โมดูลภาคจ่ายไฟ (Power Supply Module)

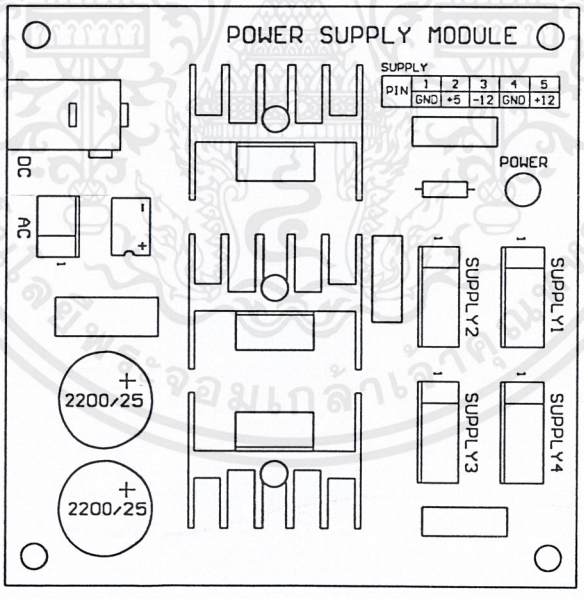
โมดูลภาคจ่ายไฟเป็นตัวจ่ายแรงดันในกับโมดูลทุกๆ โมดูล มีแรงดันให้ใช้งานคือ 5 โวลต์ สำหรับจ่ายให้อุปกรณ์ดิจิตอลที่เป็น TTL แรงดัน ±12 โวลต์ สำหรับอุปกรณ์ที่ต้องใช้แรงดันสูง เช่น ออปแอมป์ และมอเตอร์ ภาคจ่ายไฟมีวงจรดังรูปที่ 3.54

วงจรจะใช้หม้อแปลงชนิด 3 เฟสจากภายนอก ในการจ่ายไฟให้โมดูล ซึ่งแรงดันประมาณ 15-0-15 VAC จากนั้นก็จะนำมาผ่านการเรียงกระแสเป็นกระแสตรง และใช้เร็กกูเรเตอร์ (Regulator) เป็นตัวควบคุมระดับแรงดันคงที่ จากนั้นจะถูกต่อไปใช้งานด้วยคอนเนคเตอร์ขนาด 5 ขา มีแรงดันแต่ละขาคือ 0, 5, -12, 0, +12 โวลต์ และในวงจรภาคจ่ายไฟก็ยังสามารถใช้อะแดปเตอร์ (Adapter) ขนาด 12 โวลต์ ขึ้นไป เป็นตัวจ่ายแทนหม้อแปลงก็ได้ แต่แรงดันที่ได้จะไม่มีแรงดัน –12 โวลต์ ซึ่งจะทำให้โมดูลแปลงระหว่างสัญญาณแอนะลอกและสัญญาณดิจิตอลทำงานไม่ได้ ส่วนโมดูลอื่นๆ ก็ยังทำงานได้ตามปกติ โมดูลภาคจ่ายไฟมีลักษณะดังรูปที่ 3.55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

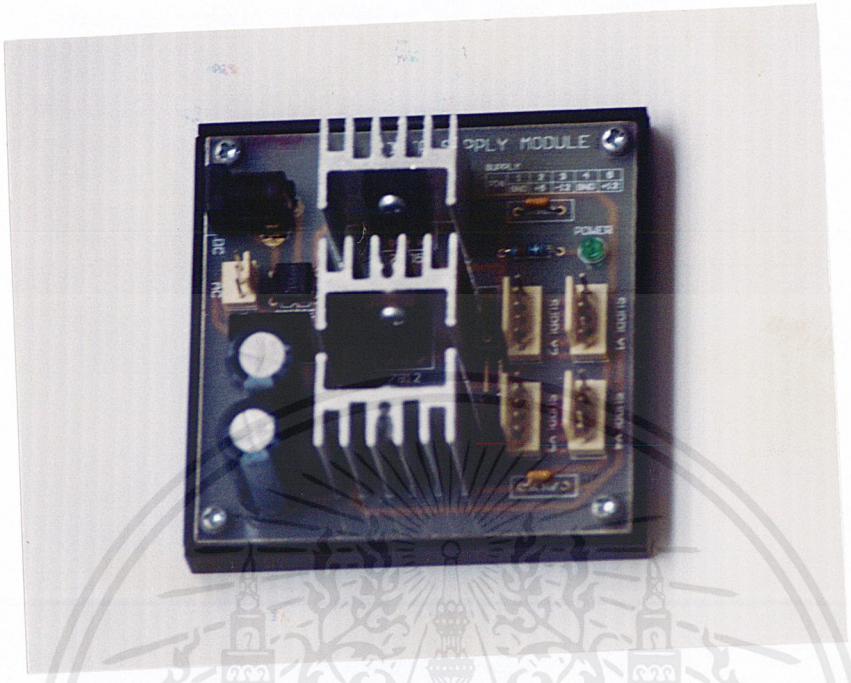


รูปที่ 3.54 วงจรภาคจ่ายไฟ



รูปที่ 3.55 แผนผังอุปกรณ์ของโมดูลภาคจ่ายไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.56 โมดูลภาคจ่ายไฟ

3.3 การออกแบบทางด้านซอฟต์แวร์

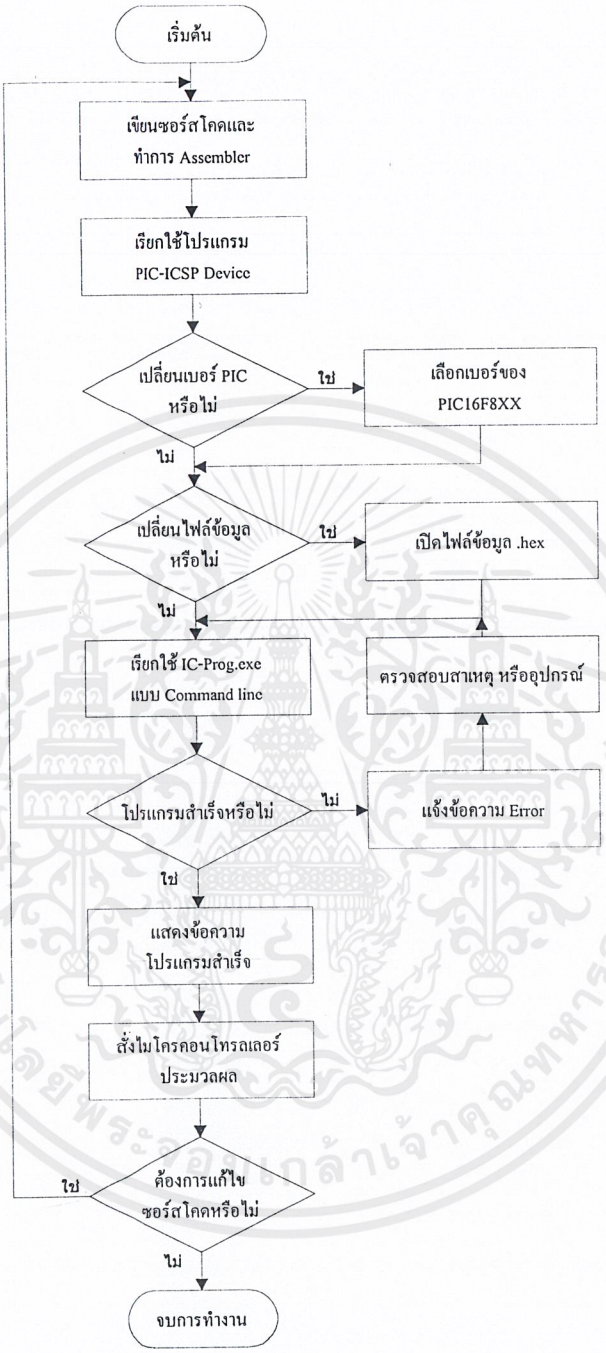
3.3.1 การออกแบบโปรแกรม PIC – ICSP Device

โปรแกรม PIC-ICSP Device เป็นโปรแกรมที่ทำงานบนระบบปฏิบัติการ Windows 9X สร้างขึ้นจาก Delphi 4 ใช้สำหรับโหลดข้อมูลโปรแกรมจากคอมพิวเตอร์ลงสู่หน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ PIC16F8XX ผ่านพอร์ตขนาน (Parallel Port) นอกจากนี้ยังสามารถใช้เป็นตัวควบคุมการประมวลผล การรีเซต และการหยุดการประมวลผลได้จากโปรแกรม

3.3.2 การส่งข้อมูลโปรแกรมลงหน่วยความจำโปรแกรมของ PIC16F8XX

โปรแกรม PIC-ICSP Device จะเรียกใช้งานโปรแกรมชื่อ IC-Prog.Exe ในลักษณะ Command Line เพื่อโหลดข้อมูลไปยัง PIC16F8XX โดยส่ง Part ของไฟล์นามสกุล . Hex ที่ต้องการและเบอร์ของไมโครคอนโทรลเลอร์ PIC16F8XX ที่ใช้ทดลองให้โปรแกรม IC-Prog.Exe เป็นตัวจัดการโหลดข้อมูลลงบนไมโครคอนโทรลเลอร์ PIC16F8XX ดังแสดงขั้นตอนตามแผนผังการทำงานในรูปที่ 3.57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.57 ขั้นตอนการโหลดข้อมูลโปรแกรมลงบนหน่วยความจำโปรแกรม

จากรูปที่ 3.56 แสดงถึงขั้นตอนในการปฏิบัติการทดลองไมโครคอนโทรลเลอร์ PIC16F8XX โดยทำการเชื่อมต่อโมดูลต่างๆ ตามที่ต้องการทดลองแล้ว ก็ทำการเขียนซอสโค้ด (Source Code) โปรแกรมการทำงานให้แก่ PIC16F8XX โดยจะใช้โปรแกรม MPLAB ในการเขียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาษาแอสเซมบลี (Assembly) จากนั้นก็ทำการแอสเซมเบลอร์ (Assembler) เป็นไฟล์ .Hex แล้วก็ทำการโหลดข้อมูลจาก Hex ไฟล์ลงสู่ PIC16F8XX ด้วยการป้อนชื่อของไฟล์ .Hex พร้อมเส้นทางของชื่อ ลงในโปรแกรม PIC-ICSP Device ซึ่งโปรแกรมจะนำข้อมูลในไฟล์นี้โหลดลงสู่ PIC16F8XX โดยผ่านโปรแกรม IC-Prog อีกที ก็เสร็จกระบวนการโปรแกรม ต่อไปเมื่อต้องการแก้ไขคำสั่งในไฟล์แอสเซมบลีสามารถแก้ไขแล้วทำการแอสเซมเบลอร์เป็นไฟล์ Hex ชื่อเดิม จากนั้นก็สั่งโหลดข้อมูลลงสู่ PIC16F8XX ได้ทันที โดยไม่ต้องใส่เส้นทางของชื่อไฟล์ใหม่

3.3.3 การเรียกใช้โปรแกรม IC-Prog แบบ Command Line

ICSP-PIC Device จะเรียกใช้งานโปรแกรม IC-Prog โดยใช้ฟังก์ชั่น API ชื่อ ShellExecute ซึ่งสามารถใส่อาร์กิวเมนต์ (Argument) ไปพร้อมกับคำสั่ง โปรแกรม IC-Prog ก็จะทำงานตามที่เรียกใช้ โดยมีคำสั่งใช้งานดังนี้

IC-Prog.Exe [Option1] [Option2] [Option....]

Options:
- p สั่งโปรแกรม
- v เปรียบเทียบโค้ดโปรแกรม
- r อ่านโค้ด (Code)โปรแกรม
- i ซ่อน IC-Prog ขณะโปรแกรม
- t00 เบอร์ของ PIC (รหัสดูได้จากโปรแกรม IC-Prog)
- g00 ชนิดของเครื่องโปรแกรม (ต้องเลือกเป็น –g9)
- o0 เบอร์ของพอร์ตขนานหรือพอร์ตอนุกรม
- ft0 ชนิดของ Hex File
- lFILENAME ชื่อ Hex File ที่จะสั่งโปรแกรม
- sFILENAME ชื่อ Hex File ที่จะบันทึกเมื่ออ่านโค้ดโปรแกรม
- q ปิดโปรแกรม IC-Prog หลังโปรแกรมข้อมูลเสร็จ
- fxxxx รหัสป้องกันโค้ด

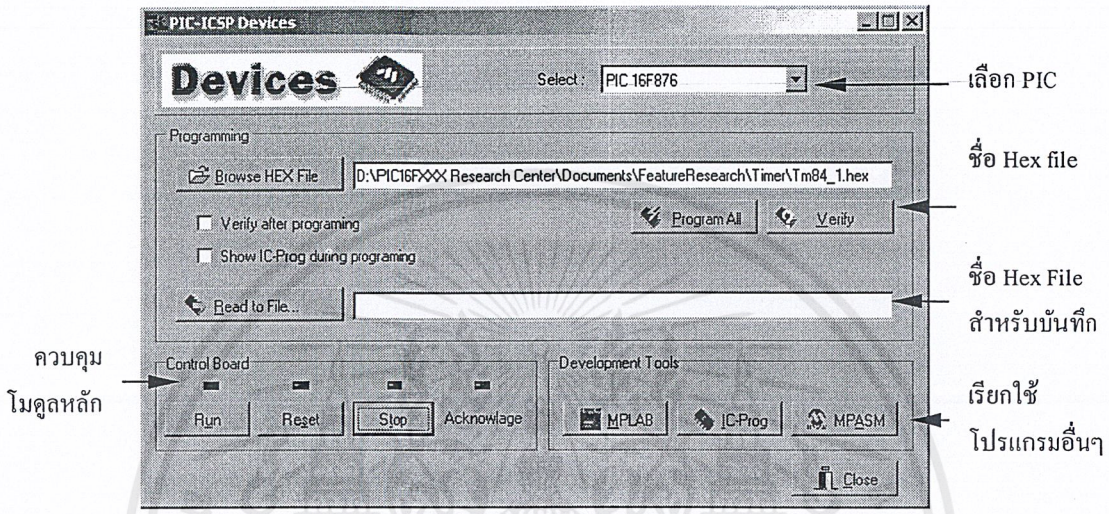
ตัวอย่างการเรียกใช้ IC-Prog.exe – p – v – i – t84 – g9 – o1 – lc : \ File.Hex - q

หมายถึง การเรียกโปรแกรมโค้ดพร้อมเปรียบเทียบโค้ดลงบน PIC16F8XX ผ่าน Parallel Port1 จาก C : \ File.Hex

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.4 การสร้างโปรแกรม PIC-ICSP Device และรายละเอียดการสร้าง

โปรแกรม PIC-ICSP Device สร้างขึ้นในฟอร์ม (Form) แบบ SDI (Single Document Interface) คือ การทำงานทุกอย่างภายในหน้าต่างเดียว



รูปที่ 3.58 ฟอร์มของโปรแกรม PIC-ICSP Device

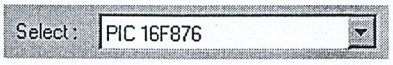
3.3.5 การใช้ฟังก์ชั่นเรียกใช้งานโปรแกรม IC-Prog.exe

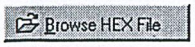
อาร์กิวเมนต์ที่จะส่งไปพร้อมกับคำสั่งเรียกใช้โปรแกรม IC-Prog.Exe นั้นจะนำมาจากค่าคุณสมบัติ (Properties) ในคอมโพแนนต์ (Components) ต่างๆ ที่อยู่บนฟอร์ม เช่น Combo Box ชื่อ Select สำหรับเลือกเบอร์ของ PIC ส่วนปุ่มชื่อ Browse Hex File... สำหรับใส่ชื่อไฟล์ .Hex พร้อม Part ซึ่งค่าต่างๆ เหล่านี้จะถูกนำมาใช้เป็นอาร์กิวเมนต์สำหรับเรียกใช้โปรแกรม IC-Prog.Exe

3.3.6 การควบคุมโมดูล PIC-ICSP

ใน Panel ชื่อ Control Board จะมีปุ่มควบคุมการประมวลผลไมโครคอนโทรลเลอร์ นั่นคือ ปุ่มชื่อ Run Reset และ Stop ซึ่งจะควบคุมผ่าน Parallel Port ไปสั่งสถานะลอจิกของไอซีบัฟเฟอร์เบอร์ 74LS125 ให้ควบคุมการประมวลผลไมโครคอนโทรลเลอร์ PIC16F8XX ที่ขา MCLR

3.3.7 การใช้งานโปรแกรม PIC-ICSP Device

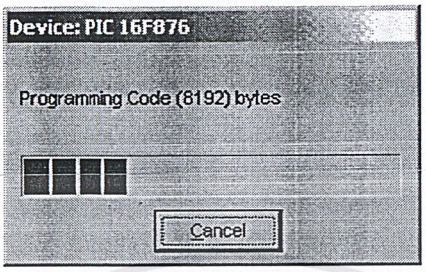
1) เลือกเบอร์ของไมโครคอนโทรลเลอร์ที่ใช้งานจาก 

2) เปิดไฟล์ Hex โดยคลิกปุ่ม 

จะปรากฏหน้าต่างเปิดไฟล์ ให้เลือกไฟล์ที่ต้องการโปรแกรม

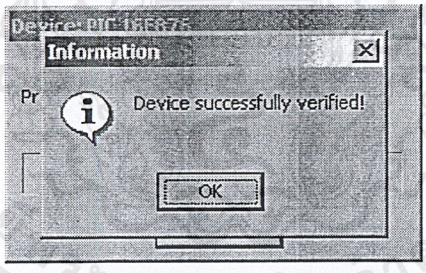
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) สามารถโปรแกรมได้ทันทีโดยคลิกปุ่ม Program All จะเริ่มทำการโปรแกรมโดยจะปรากฏหน้าต่างแสดงการโปรแกรมดังรูปที่ 3.59



รูปที่ 3.59 หน้าต่างแสดงขณะทำการโปรแกรม

ขณะโปรแกรมข้อมูลลงบนไมโครคอนโทรลเลอร์นี้ จะมี LED แสดงสถานะการโปรแกรมบนตัวโมดูล ซึ่งในขณะโปรแกรมข้อมูลนี้ห้ามกดปุ่ม RESET เป็นอันขาด เพราะจะทำให้เกิดความผิดพลาดของข้อมูลขึ้น

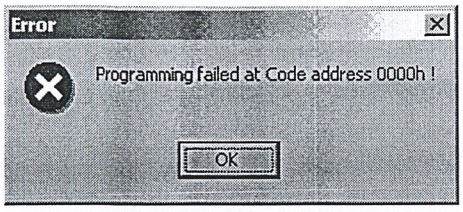


รูปที่ 3.60 เมื่อโปรแกรมเสร็จสมบูรณ์

เมื่อโปรแกรมเสร็จ จะแสดงหน้าต่างดังรูปที่ 3.60 ให้คลิกปุ่ม OK จากนั้นทำการรัน CPU โดยการคลิกที่ปุ่ม RUN ไมโครคอนโทรลเลอร์ก็จะเริ่มทำงาน

กรณีที่เกิดความผิดพลาดขึ้นในขณะดาวน์โหลดจะเกิดหน้าต่างดังรูปที่ 3.61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.61 เมื่อเกิดความผิดพลาดขณะทำการโปรแกรม

ในกรณีเกิดความผิดพลาดขึ้นซึ่งอาจเกิดจากสาเหตุดังต่อไปนี้

1.1) การเชื่อมต่อสายสัญญาณระหว่างคอมพิวเตอร์และตัวโมดูลไม่สมบูรณ์

1.2) ไม่จ่ายไฟให้ตัวโมดูล

1.3) มีโมดูลที่จ่ายสถานะอินพุตให้กับ PIC-ICSP Module ต่ออยู่ที่ Port RB ทำให้ข้อมูลที่ส่งไปโปรแกรมเกิดการผิดพลาด ให้ถอดอุปกรณ์นั้นออกก่อนทำการโปรแกรม

1.4) ในระหว่างโปรแกรมห้ามกดปุ่ม RESET ไม่ว่าจะเป็นบนตัวโมดูล หรือโปรแกรม PIC-ICSP Devices เพราะจะเกิดความผิดพลาดขึ้นทันที

4) เมื่อโปรแกรมเสร็จ สามารถที่จะตรวจสอบโค้ดโปรแกรมบนตัวไมโครคอนโทรลเลอร์ได้ว่าข้อมูลตรงกันกับที่โปรแกรมไปหรือไม่โดยคลิกปุ่ม 

ก็จะแสดงหน้าต่างการเปรียบเทียบโค้ด ถ้ามีโค้ดไม่ตรงกัน จะแสดงหน้าต่างบอกความผิดพลาดแสดงขึ้นมา ให้ตรวจเช็คความผิดพลาดที่เกิดขึ้นซึ่งอาจจะมาจากการติดต่อระหว่างตัวโมดูลกับคอมพิวเตอร์ จากนั้นทำการโปรแกรมใหม่อีกครั้ง

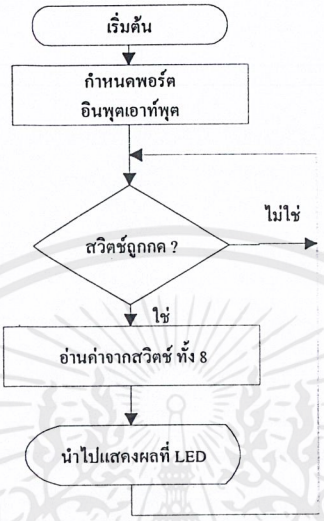
5) โปรแกรม PIC-ICSP Devices สามารถสั่งงานตัวโมดูลได้คือ เมื่อคลิกที่ปุ่ม RESET ใน Control Board ก็จะเป็นการ Reset ตัว CPU ให้เริ่มต้นทำงานใหม่ และเมื่อกดปุ่ม STOP ก็จะเป็นการหยุดการประมวลผลของ CPU โดยจะมีลักษณะเหมือนการกด Reset ค้างไว้โดยเมื่อต้องการให้ CPU ประมวลผลอีกครั้ง ก็คลิกที่ปุ่ม RUN ตัว CPU ก็จะเริ่มการทำงานใหม่อีกครั้ง

6) สามารถเรียกใช้โปรแกรมอื่นได้ คือ โปรแกรม MPLAB ซึ่งเป็นโปรแกรมสำหรับจัดการกับการเขียนโปรแกรมภาษาแอสเซมบลี สั่งตัวไมโครคอนโทรลเลอร์โปรแกรม IC-Prog เป็นโปรแกรมสำหรับจัดการการดาวน์โหลดไฟล์ Hex ลงบนตัวไมโครคอนโทรลเลอร์ โปรแกรม MPASM เป็นตัวแอสเซมเบลอร์ สร้างไฟล์ .Hex ซึ่งสำหรับการเรียกใช้ โปรแกรม MPLAB และ MPASM นั้น จะต้องมีโปรแกรมดังกล่าวติดตั้งอยู่บนเครื่องคอมพิวเตอร์แล้วเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

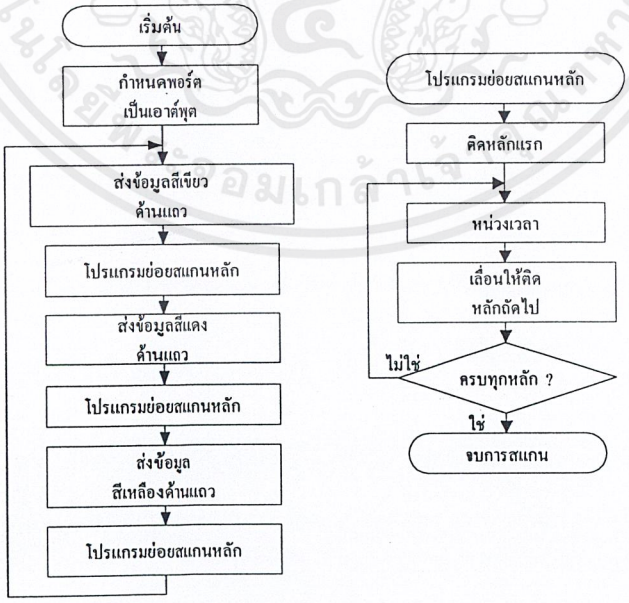
3.4 การออกแบบโปรแกรมทดลองการทำงานของโมดูล

3.4.1 ผังการทำงานของโปรแกรมทดลองโมดูลเบสิกอินพุตเอาต์พุตพื้นฐาน



รูปที่ 3.62 ผังการทำงานของโปรแกรมทดลองโมดูลเบสิกอินพุตเอาต์พุตพื้นฐาน

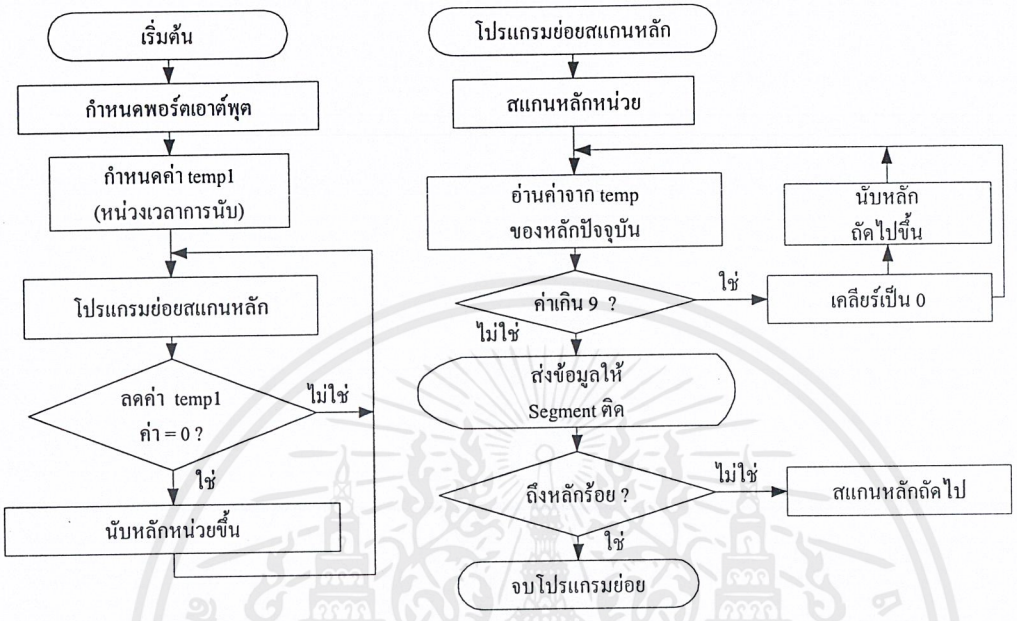
3.4.2 ผังการทำงานของโปรแกรมทดลองแอลอีดีดอตเมตริกซ์



รูปที่ 3.63 ผังการทำงานของโปรแกรมทดลองแอลอีดีดอตเมตริกซ์

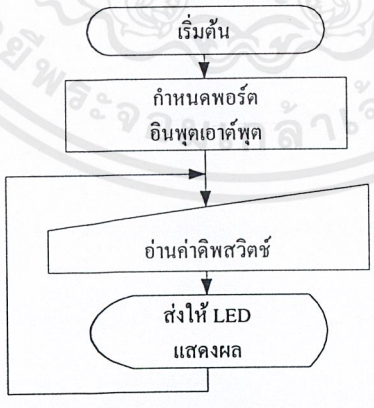
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.3 ผังการทำงานของโปรแกรมทดลองโมดูลแสดงผลตัวเลข 7 ส่วน นับเลข 4 หลัก



รูปที่ 3.64 ผังการทำงานของโปรแกรมทดลองโมดูลแสดงผลตัวเลข 7 ส่วน นับเลข 4 หลัก

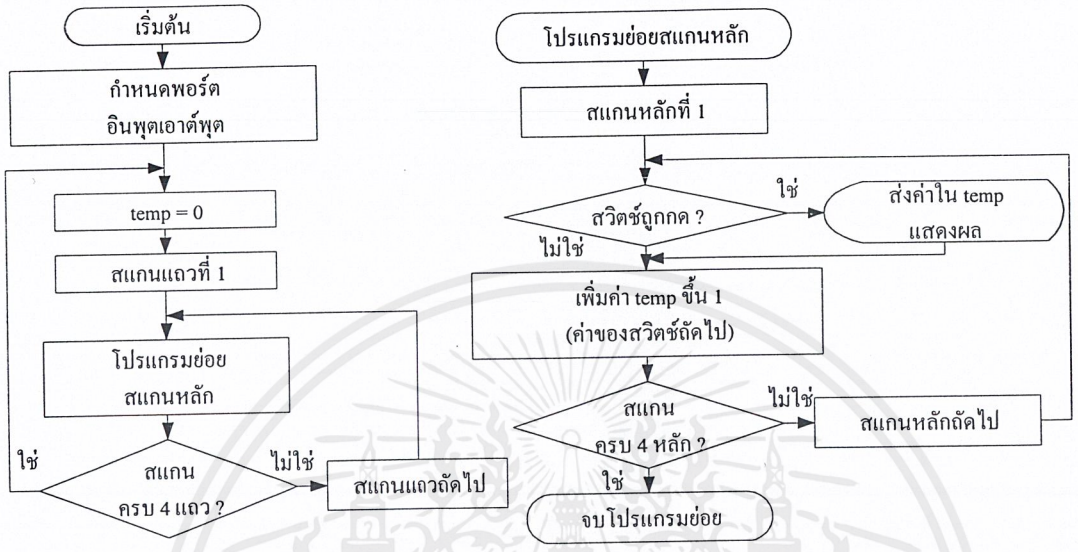
3.4.4 ผังการทำงานของโปรแกรมทดลองดิพสวิตช์



รูปที่ 3.65 ผังการทำงานของโปรแกรมทดลองดิพสวิตช์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

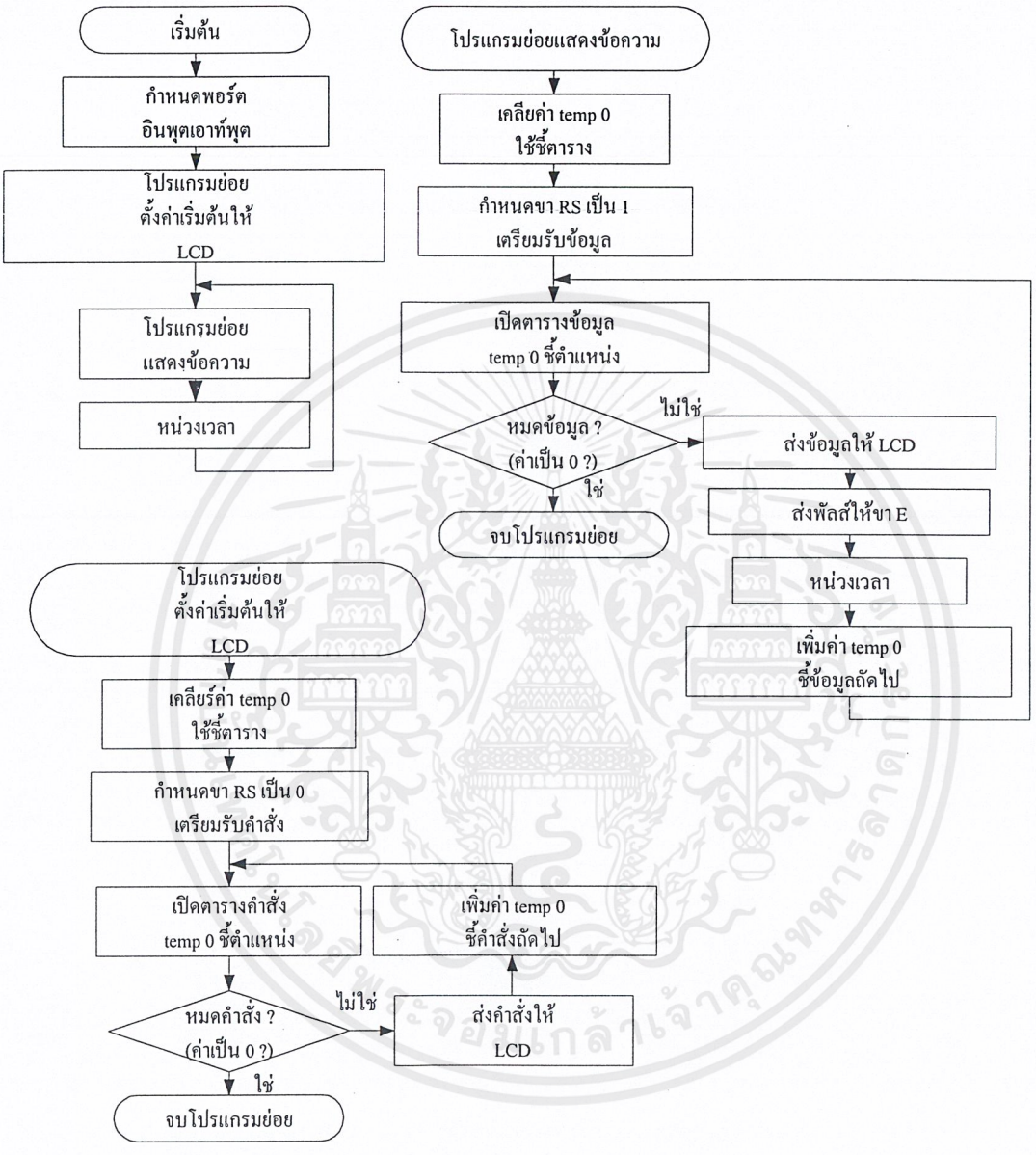
3.4.5 ผังการทำงานของโปรแกรมทดลองเมตริกสวิตช์



รูปที่ 3.66 ผังการทำงานของโปรแกรมทดลองเมตริกสวิตช์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

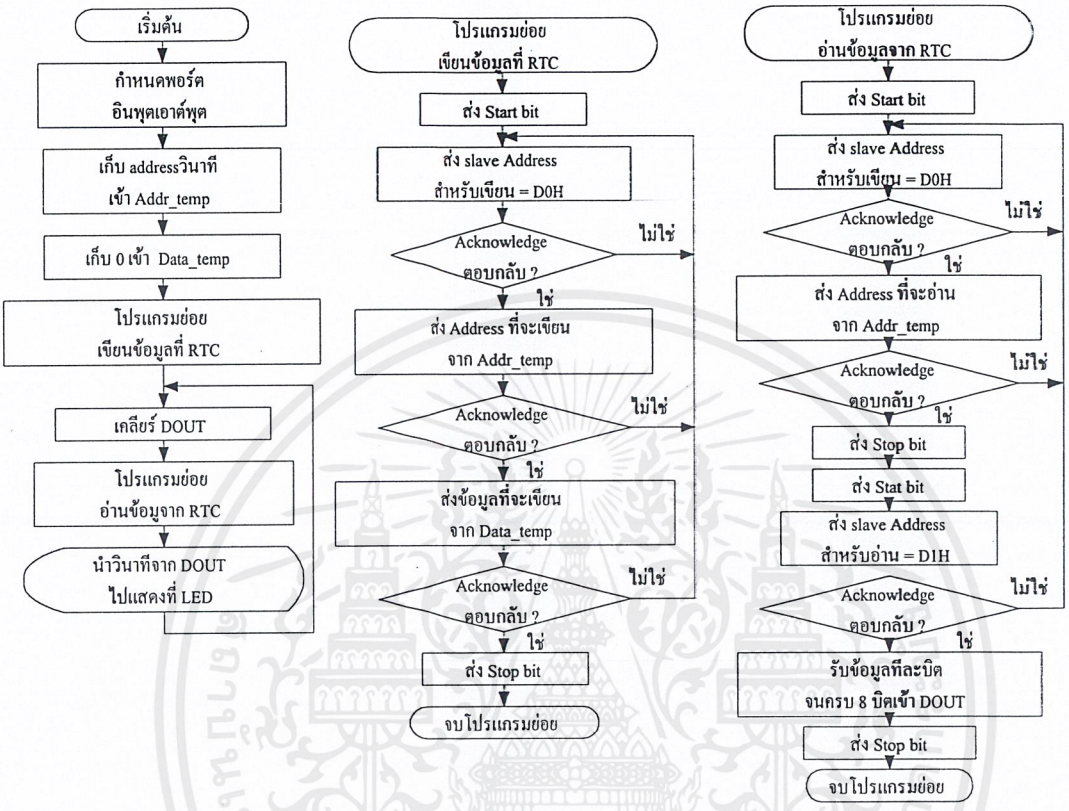
3.4.6 ผังการทำงานของโปรแกรมทดลองโมดูลแสดงผลแบบผลึกเหลว



รูปที่ 3.67 ผังการทำงานของโปรแกรมทดลองโมดูลแสดงผลแบบผลึกเหลว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

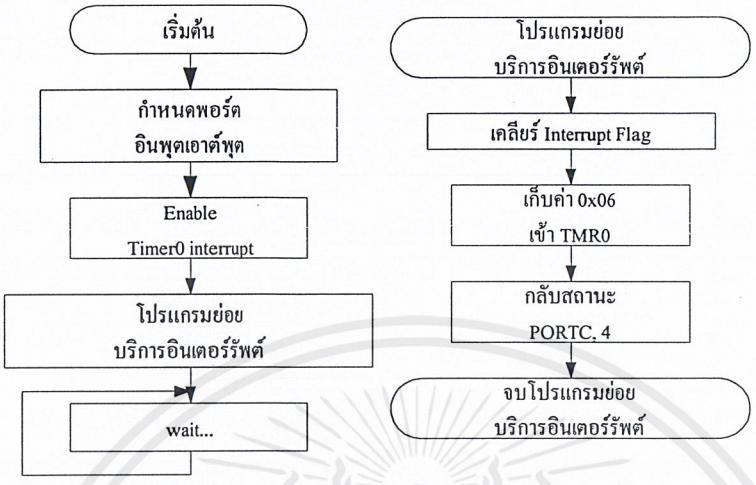
3.4.7 ผังการทำงานของโปรแกรมทดลองการอ่านข้อมูลวินาทีจาก RTC



รูปที่ 3.68 ผังการทำงานของโปรแกรมทดลองการอ่านข้อมูลวินาทีจาก RTC

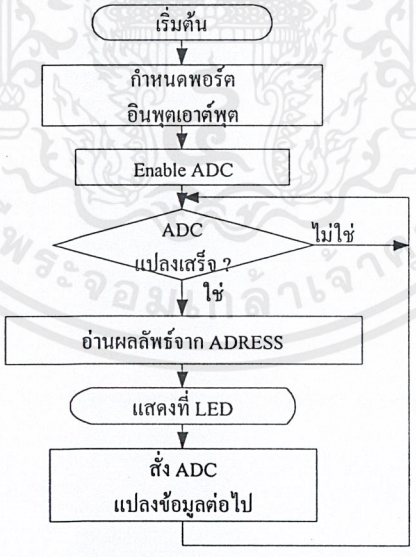
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.8 ผังการทำงานของโปรแกรมทดลองลำโพง ความถี่ 1 kHz



รูปที่ 3.69 ผังการทำงานของโปรแกรมทดลองลำโพง ความถี่ 1 kHz

3.4.9 ผังการทำงานของโปรแกรมทดลองการแปลงสัญญาณแอนะลอกเป็นดิจิตอล



รูปที่ 3.70 ผังการทำงานของโปรแกรมทดลองการแปลงสัญญาณแอนะลอกเป็นดิจิตอล

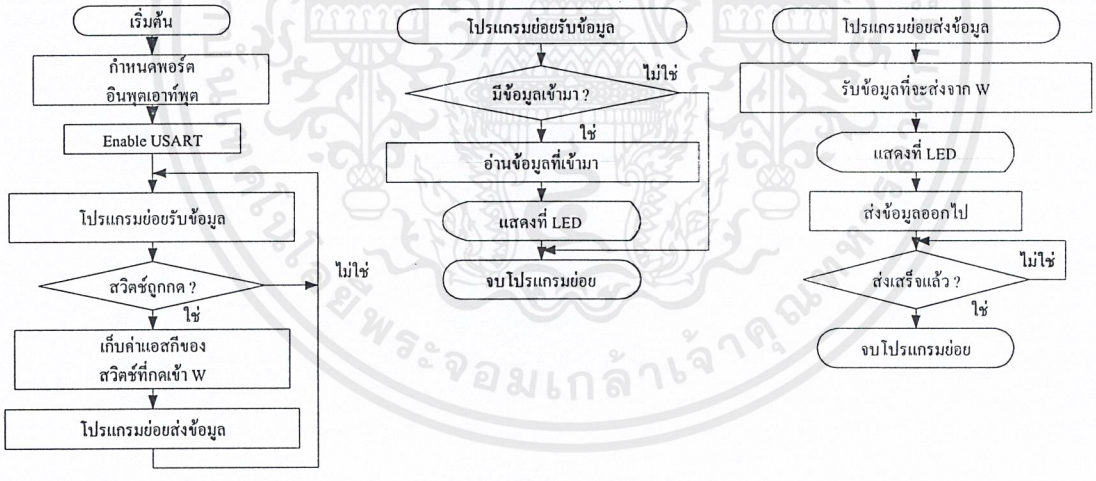
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.10 ผังการทำงานของโปรแกรมทดลองการรับข้อมูลดิจิตอลจากดิพสวิตช์



รูปที่ 3.71 ผังการทำงานของโปรแกรมทดลองการรับข้อมูลดิจิตอลจากดิพสวิตช์

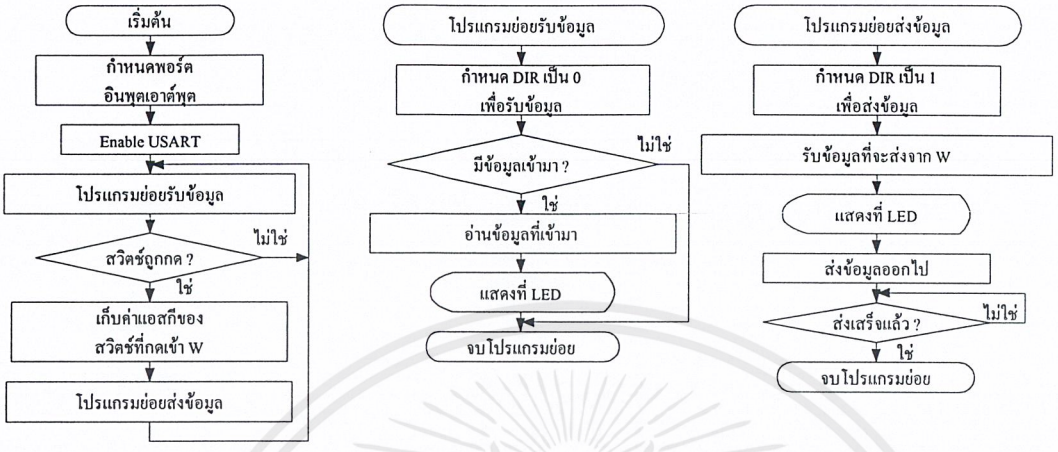
3.4.11 ผังการทำงานของโปรแกรมทดลองการรับส่งข้อมูลอนุกรมแบบ RS-232



รูปที่ 3.72 ผังการทำงานของโปรแกรมทดลองการรับส่งข้อมูลอนุกรมแบบ RS-232

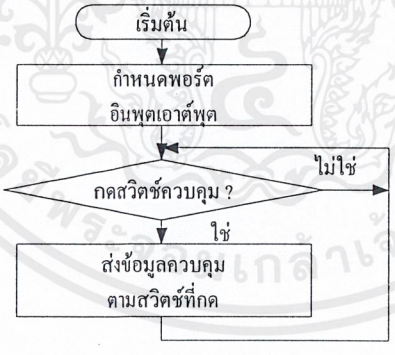
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.12 ผังการทำงานของโปรแกรมทดลองการรับส่งข้อมูลอนุกรมแบบ RS-485



รูปที่ 3.73 ผังการทำงานของโปรแกรมทดลองการรับส่งข้อมูลอนุกรมแบบ RS-485

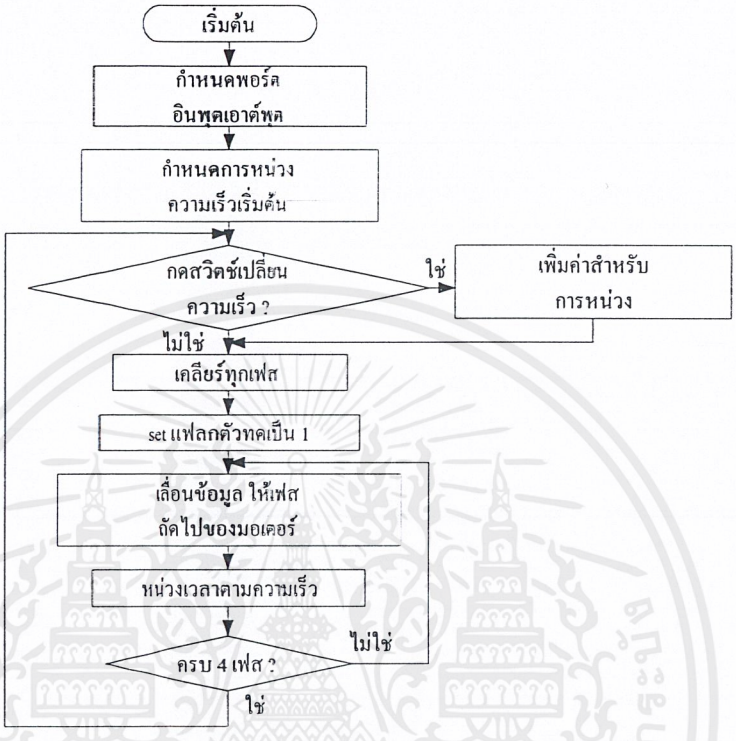
3.4.13 ผังการทำงานของโปรแกรมทดลองการควบคุมดีซีมอเตอร์



รูปที่ 3.74 ผังการทำงานของโปรแกรมทดลองการควบคุมดีซีมอเตอร์

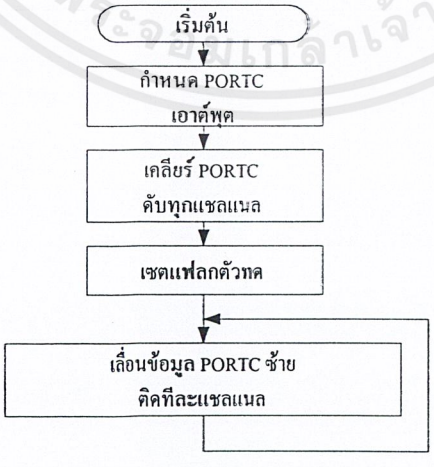
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.14 ผังการทำงานของโปรแกรมทดลองการควบคุมสเตปปิ้งมอเตอร์แบบเวฟ



รูปที่ 3.75 ผังการทำงานของโปรแกรมทดลองการควบคุมสเตปปิ้งมอเตอร์แบบเวฟ

3.4.15 ผังการทำงานของโปรแกรมทดลองการควบคุมโซลิดสเตทรีเลย์



รูปที่ 3.76 ผังการทำงานของโปรแกรมทดลองการควบคุมโซลิดสเตทรีเลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 กล่าวนำ

จากที่ได้กล่าวมาแล้วในบทที่ 3 เกี่ยวกับการสร้างและการออกแบบ ทั้งทางด้านฮาร์ดแวร์และด้านซอฟต์แวร์ ในบทนี้จะเป็นการทดลองและผลการทดลองตามที่ได้ทำการออกแบบไว้ว่าในแต่ละโมดูลที่ทำการทดลองนั้นมีผลการทดลองที่ถูกต้องหรือไม่ ดังนั้นในการทดลองจึงแยกย่อยออกเป็นการทดลองแต่ละโมดูล ดังนี้

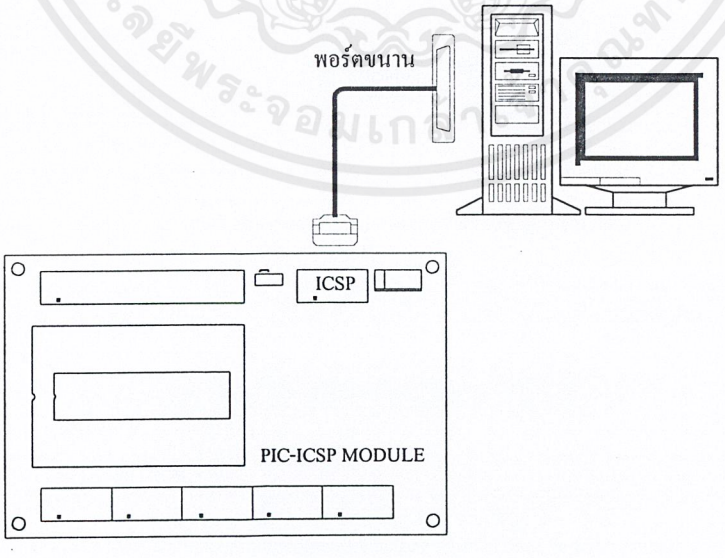
4.2 การทดลองและผลการทดลอง

4.2.1 การทดลองใช้งานโมดูล PIC-ICSP

โมดูล PIC-ICSP เป็นตัวควบคุมการทำงานของไมโครคอนโทรลเลอร์ PIC16F87X รวมทั้งการดาวน์โหลดโปรแกรมจากคอมพิวเตอร์ลงสู่ไมโครคอนโทรลเลอร์ การใช้งานโมดูลจะใช้ร่วมกับโปรแกรม PIC-ICSP Device ที่ได้เขียนขึ้น ซึ่งได้ทำการทดลองการทำงานดังนี้

ลำดับขั้นการทดลอง

1. ทำการเชื่อมต่อตัวโมดูลเข้ากับคอมพิวเตอร์ โดยต่อเข้ากับพอร์ตขนาน แสดงดังรูปที่ 4.1



รูปที่ 4.1 การเชื่อมต่อโมดูล PIC-ICSP เข้ากับคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. จ่ายแรงดันให้แก่โมดูล PIC-ICSP จากโมดูลภาคจ่ายไฟ

3. ทดลองการดาวน์โหลดข้อมูลโปรแกรมลงบนไมโครคอนโทรลเลอร์ โปรแกรมที่ใช้ในการทดลองเป็นโปรแกรม ตั้งค่าเริ่มต้นให้แก่พอร์ต เขียนขึ้นจาก MPLAB ดังนี้ แสดงดังรูปที่ 4.2

```
    List p=16f876
    Include "p16f876.inc"
__CONFIG _WDT_OFF & _PWRTE_ON & _LVP_OFF & _XT_OSC

            org     0x00
            bsf     STATUS,RP0      ; Bank 1
            movlw   0x06
            movwf   ADCON1
            movlw   0xff
            movwf   TRISA           ; PORTA = input
            clrf    TRISB           ; PORTB = output
            clrf    TRISC           ; PORTC = output
            bcf     STATUS,RP0      ; Bank 0
            clrf    PORTB
            clrf    PORTC

Start       movlw   0xff            ; set all bit
            Movwf   PORTB
            Goto    Start
            End
```

รูปที่ 4.2 โปรแกรมที่ใช้ทดลองดาวน์โหลดข้อมูลของโมดูล PIC-ICSP

ถ้าการดาวน์โหลดไม่มีข้อผิดพลาด โปรแกรม PIC-ICSP Device ก็จะไม่แสดงข้อความผิดพลาด และจะแสดงข้อความการดาวน์โหลดสมบูรณ์ ระหว่างดาวน์โหลด LED สีแดงชื่อ PROG ที่อยู่บนโมดูลก็จะติด

4. อ่านข้อมูลกลับจากไมโครคอนโทรลเลอร์ เพื่อตรวจสอบโค้ดโปรแกรมที่ได้โปรแกรมลงไป ว่าถูกต้องหรือไม่

5. ทดลองควบคุมการทำงานของไมโครคอนโทรลเลอร์ จากโปรแกรม PIC-ICSP Device โดยการกดปุ่ม RUN RESET และ STOP เพื่อดูการติดดับของ LED สีเขียวชื่อ RESET และวัดแรงดันตามจุดต่างๆ โดยเฉพาะที่ขา MCLR ของไมโครคอนโทรลเลอร์

ผลการทดลอง

การทดลองดาวน์โหลดข้อมูลลงบนไมโครคอนโทรลเลอร์ ให้ผลถูกต้อง LED แสดงการโปรแกรมติด เมื่อโปรแกรมเสร็จ LED ชื่อ RESET ก็ติดแทนเพื่อรอรับคำสั่งการทำงาน เมื่อการอ่านข้อมูลเปรียบเทียบกับข้อมูลเดิมถูกต้องตรงกัน ซึ่งแสดงว่า เครื่องโปรแกรมไมโครคอนโทรลเลอร์ PIC16F87X ทำงานแล้ว จากนั้นลองสั่งให้ไมโครคอนโทรลเลอร์ทำงาน LED ชื่อ RESET ก็ดับลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราสามารถรู้ว่าการควบคุมการทำงานถูกต้องหรือไม่ดูได้จากแรงดันที่ขา MCLR เมื่อสั่ง RUN วัดแรงดันได้ 5V เมื่อสั่ง RESET หรือ STOP วัดแรงดันได้ 0V แสดงว่า การควบคุมการทำงานถูต้อง

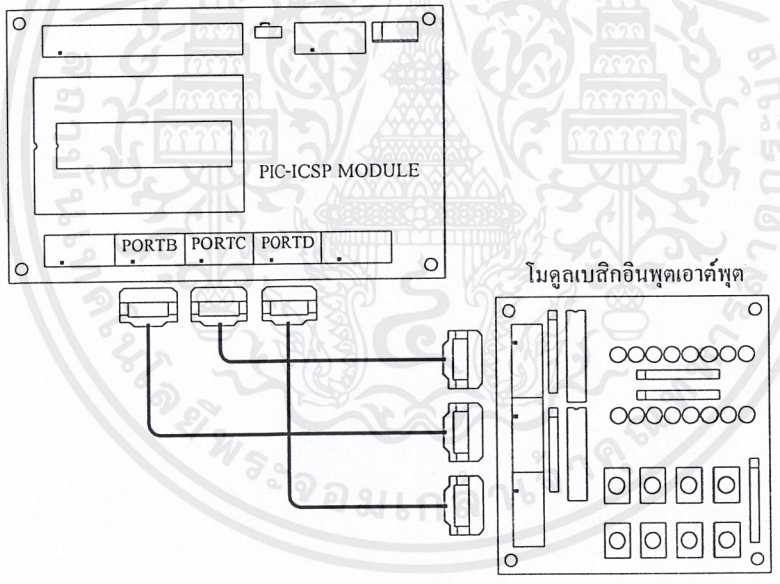
จากผลการทดลองสามารถบอกได้ว่าโมดูล PIC-ICSP ทำงานได้แล้วประมาณ 80 เปอร์เซ็นต์ เนื่องจากยังไม่ได้ต่อโมดูลอื่นๆ ให้แสดงผลการทำงานจริง ดังนั้นจึงทำการทดลองขั้นต่อไป คือ การเชื่อมต่อโมดูล PIC-ICSP นี้เข้ากับโมดูลแสดงผล แล้วทดลองใช้งาน

4.2.2 การทดลองใช้งานโมดูลเบสิกอินพุตเอาต์พุตพื้นฐาน

โมดูลนี้ประกอบไปด้วยสวิตช์สำหรับสร้างระดับลอจิกอินพุตพื้นฐาน และแอลอีดีสำหรับแสดงระดับลอจิกเอาต์พุตพื้นฐาน การทดลองการทำงานจะทดลองพร้อมกัน โดยจะเขียนโปรแกรมแสดงผลแอลอีดี เมื่อกดสวิตช์บิตใด จาก SW0 ถึง SW7 หลอดที่บิตนั้นก็จะติด

ลำดับขั้นการทดลอง

1. ทำการเชื่อมต่อโมดูลเบสิกอินพุตเอาต์พุตพื้นฐานกับโมดูลหลัก PIC-ICSP ดังรูปที่ 4.3



รูปที่ 4.3 การเชื่อมต่อใช้งานโมดูลเบสิกอินพุตเอาต์พุต

2. จากผังการทำงานของโปรแกรมที่ได้ออกแบบไว้ ทำการเขียนโปรแกรมที่ใช้ทดลองการทำงานของโมดูลเบสิกอินพุตเอาต์พุตพื้นฐาน ดังโปรแกรมรูปที่ 4.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
list p=16f877
  include "p16f877.inc"
__CONFIG _WDT_OFF & _PWRTE_ON & _LVP_OFF & _XT_OSC

              org     0x00
              bsf     STATUS,RP0      ; bank 1
              movlw   0x06
              movwf   ADCON1          ; digital input
              clrf    TRISB           ; PORTB => output
              clrf    TRISC           ; PORTC => output
              movlw   0xff
              movwf   TRISD           ; PORTD => input
              bcf     STATUS,RP0      ; Bank 0
              clrf    PORTB
              clrf    PORTC

Start         comf    PORTD,W
              btfsc   STATUS,Z        ; press switch ?
              goto    Start           ; no
              movwf   PORTB           ; yes
              comf    PORTB,W
              movwf   PORTC
              goto    Start

              end
```

รูปที่ 4.4 โปรแกรมทดลองโมดูลเบสิกอินพุตเอาต์พุต

3. สั่งให้ไมโครคอนโทรลเลอร์ทำงาน โดยทดลองกดสวิตช์ จาก SW0 ถึง SW7 แล้วสังเกตหลอด LED ที่ต่ออยู่กับพอร์ต B และ พอร์ต C โดยหลอด LED ที่พอร์ต B จะต้องติดทีละดวง ตรงกับตำแหน่งบิตที่กดสวิตช์ ส่วนแอลอีดีที่พอร์ต C จะติดกลับสถานะกับพอร์ต B คือ ติดทีละ 7 ดวง ดวงที่ไม่ติดจะตรงกับดวงที่ติดในพอร์ต B

ผลการทดลอง

เป็นไปตามที่กำหนดในโปรแกรม แสดงว่าโมดูลเบสิกอินพุตเอาต์พุตพื้นฐานใช้งานได้ ส่วนแสดงผล LED ติดครบทุกดวงทั้งสีแดงและสีเขียว และก็สามารถกดสวิตช์สร้างระดับลอจิกเพื่อป้อนให้แก่ไมโครคอนโทรลเลอร์ PIC16F87X ได้

4.2.3 การทดลองใช้งานโมดูลเครื่องมือวัด

โมดูลนี้ เป็นเครื่องมือวัดสำหรับใช้ในการทดลองใบงาน จะทดลองว่าโมดูลทำงานได้อย่างสมบูรณ์ โดยต้องนำมาวัดพร้อมกับทำการปรับแต่ง ให้ผลการวัดเป็นไปอย่างถูกต้อง โดยจะแยกการทดลองออกเป็นภาคต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) การทดลองเครื่องมือวัดแรงดัน

ลำดับขั้นการทดลอง

1. นำสายโพรบเสียบเข้าที่คอนเนคเตอร์ชื่อ VMETER

2. การปรับแต่งเครื่องมือวัด (Adjust) ให้ได้ผลการวัดที่ถูกต้อง โดยช็อตสายโพรบ แล้วปรับตัวต้านทานปรับค่าสองตัวที่อยู่ภาคปรับขนาดแรงดัน โดยด้านซ้ายมือจะเป็นตัวต้านทานปรับอัตราขยาย ด้านขวาเป็นตัวต้านทานปรับแรงดันอ้างอิง ซึ่งจะต้องปรับให้แสดงผลเป็นแรงดัน 0.00 VDC จากนั้นลองนำโพรบไปวัดแรงดันที่รู้ค่าแน่นอน แล้วทำการปรับตัวต้านทานทั้งสองตัวให้แสดงค่าแรงดันที่ถูกต้อง

3. ทดลองโดยนำไปวัดแรงดันจากแหล่งจ่ายไฟปรับค่าได้เพื่อดูการแสดงผล ให้ตรงกับค่าที่แหล่งจ่ายไฟจ่ายออกมา ทั้งค่าบวกและค่าลบโดยการกลับขั้วโพรบ

ผลการทดลอง

จากการปรับแต่งและทดลองวัดแรงดัน จาก +25.50 V ถึง –25.55 V ถ้าวัดแรงดันเกินจากนี้จะแสดงผลข้อความ Error พร้อมเสียงเตือน ทำให้ทราบว่าภาควัดแรงดันสามารถใช้งานได้

2) การทดลองเครื่องมือวัดความถี่

ลำดับขั้นการทดลอง

1. เสียบสายโพรบที่คอนเนคเตอร์ชื่อ FREQ สำหรับวัดความถี่ โดยต้องใช้สายโพรบสีแดงวัดสายสัญญาณเท่านั้น ห้ามนำสายสีดำไปวัด เนื่องจากสายสีดำนี้เป็นสายกราวน์ ซึ่งอาจจะทำให้สายสัญญาณลัดวงจรแล้วเกิดเสียหายได้ ส่วนสายสีดำให้ต่อเข้ากับกราวน์ของสัญญาณ

2. เครื่องมือวัดความถี่นี้จะไม่มีการปรับแต่งใดๆ เพราะความถูกต้องของการวัดจะขึ้นอยู่กับโปรแกรม สามารถนำไปวัดความถี่ได้ทันทีโดยทดลองนำไปวัดสัญญาณจากเครื่องกำเนิดความถี่แล้วทดลองปรับความถี่ไปเรื่อยๆ ทุกๆ ค่าตามที่แสดงผลได้

ผลการทดลอง

ความถี่ที่วัดได้แต่ละค่าใกล้เคียงกับที่เครื่องกำเนิดความถี่กำเนิดออกมา โดยสามารถวัดความถี่ได้ตั้งแต่ 2 Hz ถึง 999999 Hz เมื่อลองปรับขนาดของสัญญาณที่เครื่องกำเนิดความถี่ ได้ผลว่าถ้าขนาดของสัญญาณเล็กเกินไปจะทำให้ค่าความถี่ที่วัดได้ผิดพลาดมากขึ้น ขนาดของสัญญาณที่วัดควรจะมากกว่า 1 V จากการทดลองทำให้ทราบว่าภาควัดความถี่สามารถใช้งานได้

3) การทดลองรับ-ส่งสัญญาณแบบ RS-485

การทดลองจำเป็นต้องมีตัวรับส่งสัญญาณ RS-485 อีกตัวเพื่อทำงานร่วมกัน ซึ่งจะใช้โมดูลสื่อสารข้อมูลอนุกรมเป็นตัวทดลองร่วม โดยรายละเอียดของการทดลองได้แสดงรวมกันไว้ในหัวข้อ การทดลองใช้งานโมดูลสื่อสารข้อมูลอนุกรม

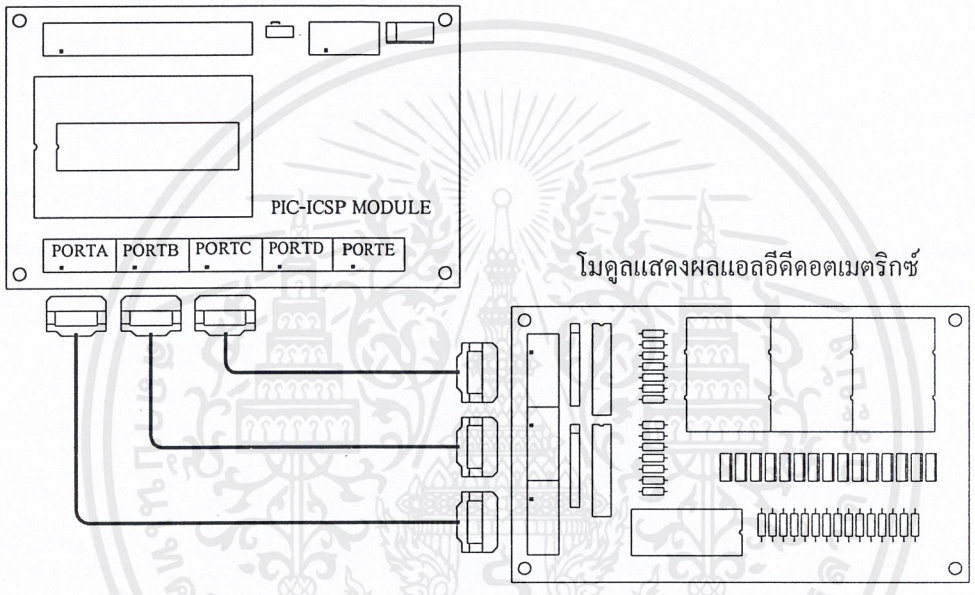
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4 การทดลองใช้งานโมดูลแสดงผลแอลอีดีดอตเมตริกซ์

โมดูลแสดงผลแอลอีดีดอตเมตริกซ์ แสดงผลได้ทั้งสีแดงและสีเขียว แต่เมื่อให้ทั้งสองสีติดพร้อมกันก็จะแสดงเป็นสีเหลือง การทดลองจะทำโดยให้แอลอีดีติดเป็นไฟวิ่งทีละคอลัมน์ โดยติดรอบละ 1 สี สลับกันไปทั้งสามสี เพื่อดูว่าแอลอีดีทุกดวงทำงานได้หรือไม่

ลำดับขั้นการทดลอง

1. ต่อโมดูลแสดงผลแอลอีดีดอตเมตริกซ์เข้ากับโมดูลหลัก แสดงดังรูปที่ 4.5



รูปที่ 4.5 การเชื่อมต่อใช้งานโมดูลแอลอีดีดอตเมตริกซ์

2. จากผังการทำงานของโปรแกรมที่ออกแบบไว้ ทำการเขียนโปรแกรมที่ใช้ในการทดลองการทำงานของโมดูลแอลอีดีดอตเมตริกซ์ โปรแกรมแสดงดังรูปที่ 4.6

```
; Port A Column
; Port B Green
; Port C Red
            cblock   0x20
            Count1, Count2          ; variable
            Endc
    list p=16f877
    include "p16f877.inc"
  __config _PWRTE_ON & _WDT_OFF & _LVP_OFF & _XT_OSC
            org      0x00
            bsf      STATUS,RP0
            movlw    0x06
            movwf    ADCON1         ; portA => digital port
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            clrf    TRISA
            clrf    TRISB
            clrf    TRISC           ; config RA, RB, RC Output All
            bcf     STATUS,RP0
            clrf    PORTB
            clrf    PORTC
Start       movlw   0xff
            Movwf   PORTB           ; data = green
            Call    ScanColumn
            clrf    PORTB
            movlw   0xff
            movwf   PORTC           ; data = red
            call    ScanColumn
            movlw   0xff
            movwf   PORTB           ; data = yellow
            call    ScanColumn
            clrf    PORTB
            clrf    PORTC
            goto    Start
ScanColumn  clrf    PORTA
Scan1       call    Delay
            Incf    PORTA,F         ; increment to next column
            Movlw   0x0f
            subwf   PORTA,W
            btfss   STATUS,Z        ; column = 0fH ?
            goto    Scan1           ; no
            return                  ; yes
Delay       movlw   0xff
            Movwf   Count1
Loop2       movlw   0xff
            Movwf   Count2
Loop1       decfsz  Count2,F
            Goto    Loop1
            Decfsz  Count1,F
            goto    Loop2
            return
            end
```

รูปที่ 4.6 โปรแกรมทดลองโมดูลแอลอีดีดอตเมตริกซ์

3. สั่งให้ไมโครคอนโทรลเลอร์ทำงาน สังเกตการติดของ LED ว่าติดครบทุกหลอดหรือไม่

ผลการทดลอง

แอลอีดีติดครบทุกดวง โดยจะติดทีละคอลัมน์วิ่งไปทางด้านซ้าย รอบละสี วนไปเรื่อยๆ ทั้งสามสี จากการทดลองนี้บ่งบอกว่าแอลอีดีทุกดวงของดอตเมตริกซ์สามารถใช้งานได้ ซึ่งก็สามารถเขียนโปรแกรมแสดงผลให้ติดตามรูปแบบที่ต้องการได้ โดยการเพิ่มความเร็วในการสแกนทางคอลัมน์ และเปลี่ยนข้อมูลที่ป้อนให้ทางด้านแถว

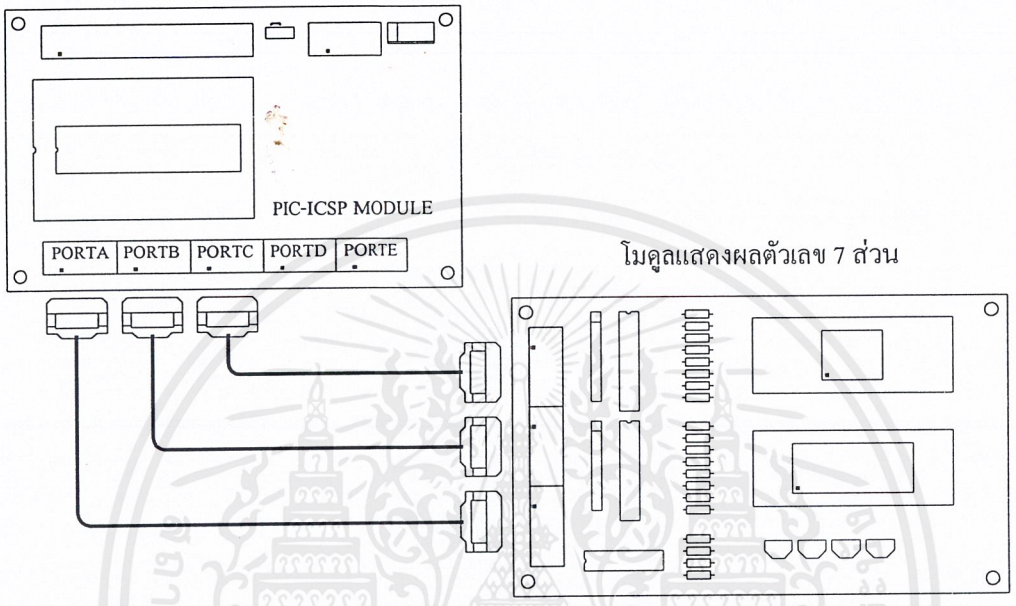
4.2.5 การทดลองใช้งานโมดูลแสดงผลตัวเลข 7 ส่วน

โมดูลแสดงผลตัวเลข 7 ส่วน ขนาด 8 หลัก โดยแบ่งเป็นขนาด 4 หลัก 2 ชุด แบ่งเป็นชนิดคอมมอนคาโถด และคอมมอนอาโนด ทดลองโดยเขียนโปรแกรมเคาน์เตอร์ให้กับคอมมอนคาโถดนับเลขขึ้นจาก 0 ถึง 9999 และคอมมอนอาโนดนับเลขลงจาก 9999 ถึง 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับขั้นการทดลอง

1. เชื่อมต่อโมดูลแสดงผลตัวเลข 7 ส่วนเข้ากับโมดูลหลัก โดยแสดงจุดต่อตามรูปที่ 4.7 พอร์ต A ใช้เลือกหลักที่ติด พอร์ต B และพอร์ต C ใช้ส่งข้อมูลเซ็กเมนต์ ของแต่ละคอมมอน



รูปที่ 4.7 การเชื่อมต่อใช้งานโมดูลแสดงผลตัวเลข 7 ส่วน

2. จากผังการทำงานของโปรแกรมที่ได้ออกแบบไว้ เขียนโปรแกรมทดลองการทำงานของโมดูลแสดงผลตัวเลข 7 ส่วน ดังโปรแกรมรูปที่ 4.8

```
; PORTA Column
; PORTB Common Anode  ; (9999 - 0000)
; PORTC Common Cathode; (0000 - 9999)
    list p=16f877
    include "p16f877.inc"
  __config _PWRTE_ON & _WDT_OFF & _LVP_OFF & _XT_OSC

            cblock   0x20
            Digit1, Digit2, Digit3, Digit4
            Column, CountD, Count1, Buff1
            Endc

            org      0x00
            bsf      STATUS,RP0
            movlw    0x06
            movwf    ADCON1          ; digital port
            clrf     TRISA
            clrf     TRISB
            clrf     TRISC
            bcf      STATUS,RP0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 clrf     PORTA                    ; porta Digit
          clrf     PORTC           ; portc common cathode
          movlw    0xff
          movwf    PORTB           ; portb comon anode
          clrf     Digit1
          clrf     Digit2
          clrf     Digit3
          clrf     Digit4
          clrf     Column

Start     movlw    0x10
          movwf    Count1          ; Delay counter
Lp1       call     SEG0
          decfsz   Count1,F
          goto     Lp1
          incf     Digit1,F
          goto     Start

SEG0      clrf     Column
          movf     Column,W
          movwf    PORTA
SEG11     movf     Digit1,W
          call     SEGMENT         ; call to table data segment
          movwf    Buff1
          sublw    0x00            ; check data by zero flag
          btfss    STATUS,Z        ; data = 0 ? (end of index)
          goto     SEG12           ; if no
          clrf     Digit1          ; if yes
          incf     Digit2,F

          goto     SEG11
SEG12     movf     Buff1,W
          movwf    PORTC           ; send to segment
          movf     Digit1,w
          sublw    0x09            ; 9 - accumulater(w)


          call     SEGMENT
          xorlw    0xff            ; invert for common anode
          movwf    PORTB
          call     Delay
          movf     Column,W
          addlw    0x04            ; next to 4 digit
          movwf    PORTA
          call     Delay
          clrf     PORTC
          movlw    0xff
          movwf    PORTB

          incf     Column,F
          movf     Column,W
          movwf    PORTA
SEG21     movf     Digit2,W
          call     SEGMENT
          movwf    Buff1
          sublw    0x00
          btfss    STATUS,Z
          goto     SEG2
          clrf     Digit2
          incf     Digit3,F
          goto     SEG21
SEG2      movf     Buff1,W
          movwf    PORTC
          movf     Digit2,w
          sublw    0x09
          call     SEGMENT
          xorlw    0xff
          movwf    PORTB
          call     Delay
          movlw    0x04
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            addwf   Column,W
            movwf   PORTA
            call    Delay
            clrf    PORTC
            movlw   0xff
            movwf   PORTB

            incf    Column,F
            movf    Column,W
            movwf   PORTA
SEG31       movf    Digit3,W
            call    SEGMENT
            movwf   Buff1
            sublw   0x00
            btfss   STATUS,Z
            goto    SEG32
            clrf    Digit3
            incf    Digit4,F
            goto    SEG31
SEG32       movf    Buff1,W
            movwf   PORTC
            movf    Digit3,w
            sublw   0x09


            call    SEGMENT
            xorlw   0xff

      movwf         PORTB
            call    Delay
            movlw   0x04
            addwf   Column,W
            movwf   PORTA
            call    Delay
            clrf    PORTC
            movlw   0xff
            movwf   PORTB

            incf    Column,F
            movf    Column,W
            movwf   PORTA
SEG41       movf    Digit4,W
            call    SEGMENT
            movwf   Buff1
            sublw   0x00
            btfss   STATUS,Z
            goto    SEG42
            clrf    Digit4
            goto    SEG41
SEG42       movf    Buff1,W
            movwf   PORTC
            movf    Digit4,w
            sublw   0x09
            call    SEGMENT
            xorlw   0xff
            movwf   PORTB
            call    Delay
            movlw   0x04
            addwf   Column,W
            movwf   PORTA
            call    Delay
            clrf    PORTC
            movlw   0xff
            movwf   PORTB
            return

SEGMENT     addwf   PCL,F           ; data segment
            Dt      3f, 06, 5b, 4f, 66, 6d, 7d, 07
            Dt      7f, 6f, 00
Delay       movlw   0xff
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Movwf    CountD
Dell        decfsz   CountD,F
            Goto     Dell
            Return
            End
```

รูปที่ 4.8 โปรแกรมทดลองโมดูลแสดงผลตัวเลข 7 ส่วน

3. สั่งไมโครคอนโทรลเลอร์ให้ทำงาน สังเกตดูทุกๆ เซกเมนต์ว่าติดครบหรือไม่

ผลการทดลอง

การทำงานของโมดูลเป็นไปตามที่ได้เขียนโปรแกรม แอลอีดีเซกเมนต์ทุกส่วนติดใช้งานได้ แสดงว่าโมดูลแสดงผลตัวเลข 7 ส่วน ใช้งานได้

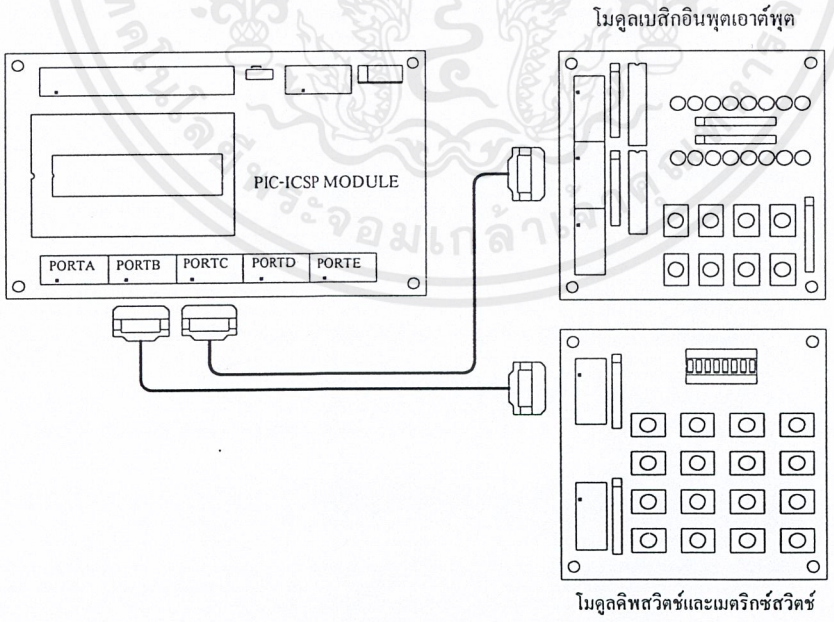
4.2.6 การทดลองใช้งานโมดูลดิพสวิตช์และเมตริกซ์สวิตช์

โมดูลป้อนระดับลอจิกเป็นอินพุต เพื่อให้รู้ว่าทำงานได้หรือไม่ จึงต้องมีส่วนแสดงผลเพื่อแสดงการทำงาน โดยใช้ส่วนของแอลอีดีแสดงผลในโมดูลเบสิกอินพุตเอาต์พุตร่วมในการทดลอง

1) การทดลองใช้งานดิพสวิตช์

ลำดับขั้นการทดลอง

1. เชื่อมต่อส่วนของดิพสวิตช์เข้ากับพอร์ต B แสดงดังรูปที่ 4.9



รูปที่ 4.9 การเชื่อมต่อใช้งานดิพสวิตช์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. จากผังการทำงานโปรแกรมที่ได้ออกแบบไว้ ทำการเขียนโปรแกรมทดลองการใช้งานดิพสวิตช์ โปรแกรมแสดงดังรูปที่ 4.10

```
        list p=16f876
     include "p16f876.inc"
  __config _PWRTE_ON & _WDT_OFF & _LVP_OFF & _XT_OSC

            org     0x00
            bsf     STATUS,RP0
            movlw   0xff
            movwf   TRISB           ; PORTB => dip switch
            clrf    TRISC           ; PORTC => LED display
            bcf     STATUS,RP0

Start       movf    PORTB,W         ; read dip switch
            Movwf   PORTC           ; mov to LED
            goto    Start

            end
```

รูปที่ 4.10 โปรแกรมทดลองดิพสวิตช์

3. เมื่อโปรแกรมลงบนไมโครคอนโทรลเลอร์เสร็จก็สั่งทำงาน จากนั้นทดลองกดดิพสวิตช์ ซึ่งแอลอีดีบนโมดูลแสดงผลต้องติดข้อมูลทั้ง 8 บิตของดิพสวิตช์

ผลการทดลอง

การติดของแอลอีดีแต่ละบิต มีสถานะเหมือนดิพสวิตช์เป็นไปตามข้อกำหนด แสดงว่าภาคดิพสวิตช์สามารถทำงานได้ถูกต้อง

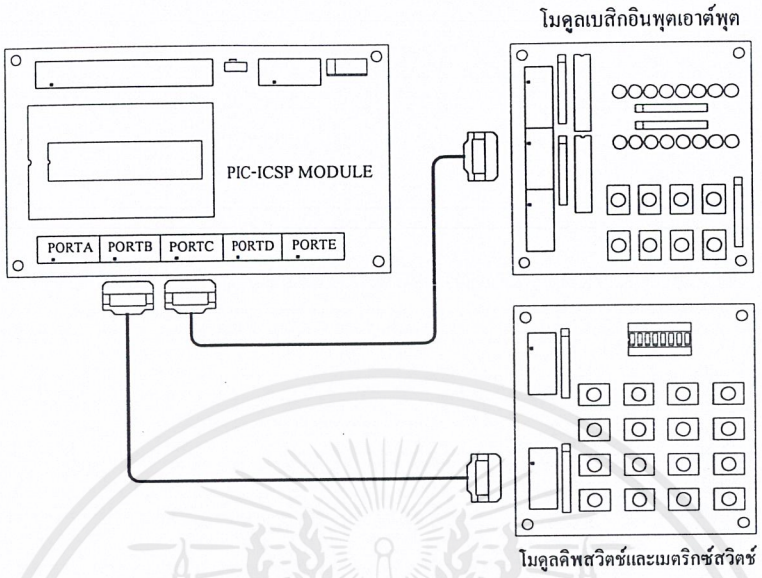
2) การทดลองใช้งานเมตริกซ์สวิตช์

เมตริกซ์สวิตช์เป็นอุปกรณ์อินพุตสำหรับสร้างลอจิก เพื่อป้อนให้ไมโครคอนโทรลเลอร์ การทดลองจึงต้องใช้โมดูลแสดงผลร่วมด้วยจะทดลองโดยการแสดงผลเป็นบิตเลขฐานสองตามสวิตช์เมตริกซ์ที่กด จาก 0 ถึง F

ลำดับขั้นการทดลอง

1. เชื่อมต่อเมตริกซ์สวิตช์เข้ากับโมดูลหลักและโมดูลแสดงผล แสดงดังรูปที่ 4.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 การเชื่อมต่อใช้งานเมตริกซ์สวิตช์

2. จากผังการทำงานโปรแกรมที่ได้ออกแบบไว้ ทำการเขียนโปรแกรมทดลองการใช้งานเมตริกซ์สวิตช์ โปรแกรมแสดงดังรูปที่ 4.12

```
list p=16f876
    include "p16f876.inc"
  __config _PWRTE_ON & _WDT_OFF & _LVP_OFF & _XT_OSC

            cblock   0x20           ; define variable start address 0x20
            DataBin
            Endc

            org      0x00
            bsf      STATUS,RP0
            movlw    b'11110000'
            movwf    TRISB          ; Key Matrix on PORTB
            clrf     TRISC          ; Display PORTC
            bcf      STATUS,RP0
            movlw    0x0f
            movwf    PORTB
            clrf     PORTC


Start       clrf     DataBin
;------------- scan row ------------
            bcf      PORTB,0        ; select row 1
            call     Key
            bsf      PORTB,0
            bcf      PORTB,1        ; select row 2
            call     Key
            bsf      PORTB,1
            bcf      PORTB,2        ; select row 3
            call     Key
            bsf      PORTB,2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            bcf     PORTB,3         ; select row 4
            call    Key


            bsf     PORTB,3
            goto    Start

;----------- scan column -----------
Key         btfss   PORTB,4         ; check switch press ?
            call    Display         ; yes, to display
            incf    DataBin,F       ; no , increment name of switch
            btfss   PORTB,5         ; check next switch ...
            goto    Display
            incf    DataBin,F
            btfss   PORTB,6
            goto    Display
            incf    DataBin,F
            btfss   PORTB,7

            goto    Display
            incf    DataBin,F
            return

Display     movf    DataBin,W       ; move name of switch
            Movwf   PORTC           ; to LED
            Return

            end
```

รูปที่ 4.12 โปรแกรมทดลองเมตริกสวิตช์

3.โปรแกรมลงบนไมโครคอนโทรลเลอร์แล้วสั่งทำงาน แล้วทดลองใช้งานเมตริกซ์สวิตช์ เมื่อกดปุ่มใด แอลอีดีแสดงผลจะต้องแสดงค่าตามปุ่มที่กด จาก 0 ถึง F

ผลการทดลอง

เมื่อทดลองใช้งานเมตริกสวิตช์ ให้ผลลัพธ์ตามที่ได้เขียนไว้ในโปรแกรม โดยเมื่อกดสวิตช์จากสวิตช์ 0 ถึง สวิตช์ F แอลอีดีก็แสดงค่าถูกต้อง แสดงว่าเมตริกซ์สวิตช์ในโมดูลใช้งานได้

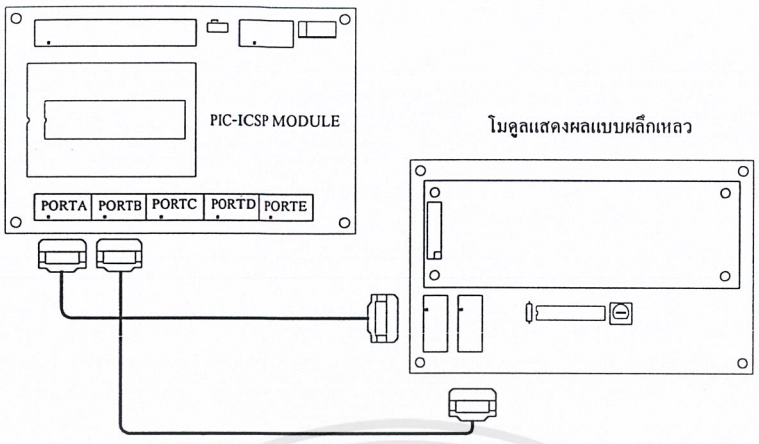
4.2.7 การทดลองใช้งานโมดูลแสดงผลแบบผลึกเหลว

การทดลองโมดูลแสดงลแบบผลึกเหลวขนาด 24 อักษร คูณ 2 บรรทัด ทำโดยการส่งข้อมูลที่เป็นข้อความให้ตัวโมดูลแสดงผล มีขั้นตอนดังนี้

ลำดับขั้นการทดลอง

1. เชื่อมต่อโมดูลแสดงผลแบบผลึกเหลวเข้ากับโมดูลหลัก โดยใช้การเชื่อมต่อแบบ 8 บิต แสดงดังรูปที่ 4.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.13 การเชื่อมต่อใช้งานโมดูลแสดงผลแบบผลึกเหลวแบบ 8 บิต

2. จากผังการทำงานโปรแกรมที่ได้ออกแบบไว้ ทำการเขียนโปรแกรมทดลองการใช้งานโมดูลแสดงผลแบบผลึกเหลวแบบ 8 บิต โปรแกรมแสดงดังรูปที่ 4.14

```
; PortB to Lcd Data 8 bit
; PortA to Control 3 bit low

    list p=16f877
    include "p16f877.inc"
  __config _PWRTE_ON & _WDT_OFF & _LVP_OFF & _XT_OSC

          cblock   0x20          ; define variable
          Reg1, Cnt1, Cnt2
          Count1, Count2, Count3
          Endc

  #define   E        0           ; define constant
  #define   RS       1
  #define   RW       2
  #define   LCD_Con  PORTA       ; PORTA control bus
  #define   LCD_Data PORTB       ; PORTB data bus

          org      0x00
Initial   bsf      STATUS,RP0
          movlw    0x06
          movwf    ADCON1
          clrf     TRISA         ; output port
          clrf     TRISB         ; output port
          bcf      STATUS,RP0
          clrf     LCD_Con
          clrf     LCD_Data


;---------------Start---------------
Start     call     Delay1
          call     InitialLCD
          call     TitleText
          call     Delay3
          goto     Start
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;--------------initial--------------
InitialLCD  clrf    LCD_Con         ; control to command
            clrf    Reg1            ; Reg1 for index pointer
IniLCD1     movf    Reg1,W
            call    Table1          ; call for command
            andlw   0xff            ; AND for zero flag
            btfsc   STATUS,Z        ; end of index ?
            return                  ; yes , return
            movwf   LCD_Data        ; no  , send to data bus
            call    Pulse           ; Pulse for Enable
            incf    Reg1,F          ; next command
            goto    IniLCD1

Table1      addwf   PCL,F           ; table of command
            dt      38, 0c, 06, 01, 00

TitleText   call    Delay1
            call    ClrScreen
            clrf    Reg1            ; Reg1 for index pointer
            bsf     LCD_Con,RS      ; control to data
            call    Dlay5ms
TT1         movf    Reg1,W
            call    Table2          ; call for ascii of charecter
            andlw   0xff
            btfsc   STATUS,Z        ; check end of sentence ?
            goto    TT2             ; yes
            movwf   LCD_Data        ; no , send to data bus
            call    Pulse           ; Pulse for Enable
            call    Delay2
            incf    Reg1,F          ; next character
            goto    TT1

TT2         bcf     LCD_Con,RS      ; control to command
            call    Dlay5ms
            movlw   0xc0            ; new row (row 2)
            movwf   LCD_Data
            call    Pulse
            call    Delay1
            clrf    Reg1
            bsf     LCD_Con,RS
            call    Dlay5ms

TT3         movf    Reg1,W
            call    Table3
            andlw   0xff
            btfsc   STATUS,Z
            goto    TT4
            movwf   LCD_Data
            call    Pulse
            call    Delay2
            incf    Reg1,F
            goto    TT3

TT4         clrf    LCD_Con
            clrf    LCD_Data
            return

Table2      addwf   PCL,F
            dt      "PIC 16F8XX Projects", 00
Table3      addwf   PCL,F
            dt      "ELECTRONICS & COMPUTERS", 00

;-------------clear screen ---------
ClrScreen   clrf    LCD_Con
            call    Dlay125us
            movlw   0x01
            movwf   LCD_Data
            call    Pulse

            return
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;-------------- Pulse --------------
Pulse       bsf     LCD_Con,E
            call    Dlay125us
            bcf     LCD_Con,E
            call    Dlay125us
            return

;-------------- Delay --------------
Dlay125us   movlw   0x2a
            movwf   Cnt1
Dlay1       decfsz  Cnt1,F
            goto    Dlay1
            return

Dlay5ms     movlw   0x29
            movwf   Cnt2
Dlay2       call    Dlay125us
            decfsz  Cnt2,F
            goto    Dlay2
            return

Delay1      movlw   0xff
            movwf   Count1
Dely1       decfsz  Count1,F
            goto    Dely1
            return

Delay2      movlw   0xa0
            movwf   Count2
Dely2       call    Delay1
            decfsz  Count2,F
            goto    Dely2
            return

Delay3      movlw   0x05
            movwf   Count3
Dely3       call    Delay2
            decfsz  Count3,F
            goto    Dely3
            return

            end
```

รูปที่ 4.14 โปรแกรมทดลองโมดูลแสดงผลแบบผลึกเหลว

3. โปรแกรมลงไมโครคอนโทรลเลอร์แล้วสั่งทำงาน โมดูลแสดงผลจะต้องแสดงข้อความ “PIC 16F87X Projects” ในบรรทัดแรก และแสดงข้อความ “ELECTRONICS & COMPUTERS” ในบรรทัดที่สอง

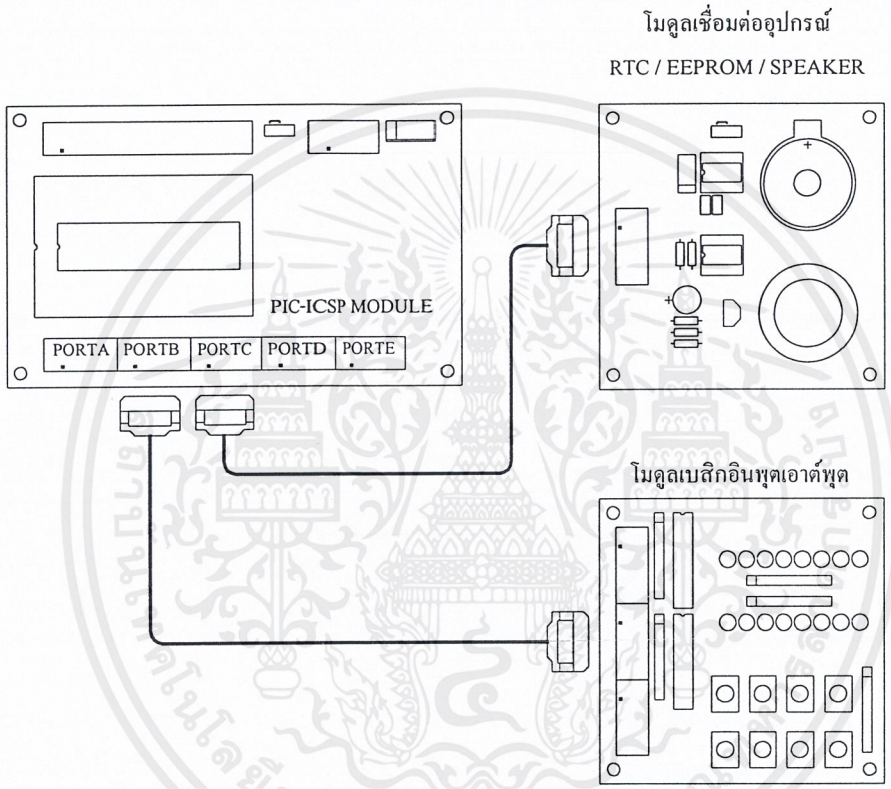
ผลการทดลอง

เมื่อสั่งทำงาน โมดูลแสดงผลแสดงข้อความตามโปรแกรมได้ถูกต้อง ตัวอักษรปรากฏทีละตัวอย่างช้าๆ คล้ายการพิมพ์ แล้วทดลองปรับความสว่างของส่วนแสดงผลที่ VR ชื่อ BRIGHTNESS ได้ผล แสดงว่าโมดูลแสดงผลแบบผลึกเหลวสามารถใช้งานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.8 การทดลองใช้งานโมดูลเชื่อมต่ออุปกรณ์ RTC / EEPROM / SPEAKER

ในโมดูลนี้มีอุปกรณ์ต่อใช้งานอยู่ 3 ตัว การทดลองจะใช้งานทีละตัว ซึ่งอุปกรณ์ทั้ง 3 ตัวมีคอนเนคเตอร์สำหรับต่อใช้งานร่วมกัน คือ เชื่อมต่อกับพอร์ต 1 พอร์ต ต่อใช้งานลักษณะเหมือนกัน ซึ่งจะเชื่อมต่อพร้อมโมดูลสำหรับแสดงผลการทดลอง แสดงการต่อใช้งานดังรูปที่ 4.15



รูปที่ 4.15 การเชื่อมต่อใช้งานอุปกรณ์ทั้ง 3 ชนิดของ โมดูลเชื่อมต่ออุปกรณ์

1) การทดลองเชื่อมต่ออุปกรณ์ระบบบัส I^2C

อุปกรณ์ RTC และ EEPROM บนโมดูล จะใช้การเชื่อมต่อแบบมาตราฐาน I^2C รับส่งข้อมูล ซึ่งการทำงานจะมีลักษณะคล้ายกัน ดังนั้นจะทดลองเชื่อมต่อกับอุปกรณ์ RTC เท่านั้น เพื่อทดลองการรับส่งข้อมูลแบบ I^2C นี้

ไอซีเบอร์ DS1307 เป็น RTC ที่มีการเชื่อมต่อใช้งานแบบ I^2C ส่งข้อมูลฐานเวลาไปให้แก่ไมโครคอนโทรลเลอร์ การทดลองใช้งานจะทำโดยอ่านค่าเวลาออกมาเป็นวินาที แล้วแสดงผลที่แอลอีดีในโมดูลแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับขั้นการทดลอง

1. เชื่อมต่อ RTC เข้ากับโมดูลหลักตามรูปที่ 4.20

2. จากผังการทำงานโปรแกรมที่ได้ออกแบบไว้ ทำการเขียนโปรแกรมทดลองการใช้งานอ่านข้อมูลวินาทีจาก RTC โปรแกรมแสดงดังรูปที่ 4.16

```
    list P=16f877
         include "p16f877.inc"
        __config _PWRTE_ON & _WDT_OFF & _LVP_OFF & _XT_OSC

    #define     SDA       PORTC,3
    #define     SCL       PORTC,2

                cblock    0x20
                dt1, dt2, XBUF, count
                ADD, DAT, DOUT
                Endc

                org       0x00
                bsf       STATUS,RP0      ; select bank 1
                movlw     b'11110011'
                movwf     TRISC
                clrf      TRISB
                bcf       STATUS,RP0      ; select bank 0
                clrf      PORTC
                clrf      PORTB
                movlw     0x00            ; Write SECOND to address 00
                movwf     ADD
                movlw     0x0             ;  Start second = 00
                movwf     DAT
                call      wr              ; call write sub
                call      Sdel            ; delay more than 10 mS

    ReadS       movlw     0x00            ; Read second
                movwf     ADD
                call      rd              ; call read sub
                movf      DOUT,w
                movwf     PORTB           ; display second

                call      Sdel
                goto      ReadS
    ;_____________________________________
    ;              Write data
    ;-------------------------------------
    ; Input    : ADD,DAT
    ; Output   : EEPROM
    ;-------------------------------------
    wr          bcf       SDA
                bcf       SCL
                call      start
                movlw     0xD0            ; Control data & slave address 00H
                movwf     XBUF
                call      TB
                movlw     TRISC
                movwf     FSR
                bsf       INDF,3          ; SDA input (hi)
                bsf       SCL             ; clock hi  9th
                btfsc     SDA
                goto      wr
                bcf       SCL             ; clock lo  9th
                call      losda

                movf      ADD,w
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            movwf   XBUF


call        TB
            movlw   TRISC
            movwf   FSR
            bsf     INDF,3          ; SDA input (hi)
            bsf     SCL             ; clock hi  9th
            btfsc   SDA
            goto    wr
            bcf     SCL             ; clock lo  9th
            call    losda

            movf    DAT,w
            movwf   XBUF
            call    TB
            movlw   TRISC
            movwf   FSR
            bsf     INDF,3          ; SDA input (hi)
            bsf     SCL             ; clock hi  9th
            btfsc   SDA
            goto    wr
            bcf     SCL             ; clock lo  9th
            call    losda
            bsf     SCL
            nop
            nop
            nop
            nop
            bsf     SDA             ; stop bit
            return
;_____________________________________
;              Read data
;-------------------------------------
; Input   : ADD
; Output  : DOUT
;-------------------------------------
rd          clrf    DOUT            ; DOUT = 0
            movlw   TRISC
            movwf   FSR
            bcf     INDF,2
            bcf     INDF,3
            bcf     SDA
            bcf     SCL
            call    start
            movlw   0xD0            ; Control data & slave address 00H
            movwf   XBUF
            call    TB
            movlw   TRISC
            movwf   FSR
            bsf     INDF,3          ; SDA input (hi)
            bsf     SCL             ; clock hi  9th
            btfsc   SDA
            goto    rd
            bcf     SCL             ; clock lo  9th
            call    losda

            movf    ADD,w
            movwf   XBUF
            call    TB
            movlw   TRISC
            movwf   FSR
            bsf     INDF,3          ; SDA input (hi)
            bsf     SCL             ; clock hi  9th
            btfsc   SDA
            goto    rd
            bcf     SCL             ; clock lo  9th
            call    losda

            bsf     SCL
            bsf     SDA             ; stop bit
            bcf     SDA
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            bcf     SCL             ; start bit


            movlw   0xD1            ; Control read
            movwf   XBUF
            call    TB
            movlw   TRISC
            movwf   FSR
            bsf     INDF,3          ; SDA input (hi)
            bsf     SCL             ; clock hi  9th
            btfsc   SDA
            goto    rd
            bcf     SCL             ; clock lo  9th
            call    losda

            bcf     STATUS,C        ; clear carry flag
            movlw   8
            movwf   count           ; counter = 8 (bits)
            movlw   TRISC
            movwf   FSR
            bsf     INDF,3          ; SDA input
;------------ Read bit -------------
N4          bsf     SCL             ; clock hi
            btfss   TRISC,3
            goto    N5              ; Din = 0
            bsf     DOUT,0          ; Din = 1

N5          rlf     DOUT,f          ; shift left 1 bit
            bcf     SCL             ; clock lo
            decfsz  count,f
            goto    N4
            rrf     DOUT,f          ; shift data to right 1 bit

            bcf     INDF,3          ; SDA output (lo)
            bsf     INDF,3          ; SDA hi
            bsf     SCL             ; clock hi
            bcf     SCL             ; clock lo
            bsf     SCL             ; clock hi
            bsf     INDF,3          ; SDA hi
            return

;_____________________________________
;             SDA low
;-------------------------------------
losda       bcf     SDA
            movlw   TRISC
            movwf   FSR
            bcf     INDF,3          ; SDA lo
            return
;_____________________________________
;             Send bit
;-------------------------------------
;   Input  : XBUF
;   Output : EEPROM
;-------------------------------------
TB          bcf     STATUS,C        ; clear carry flag
            movlw   8
            movwf   count           ; counter for 8 bit
N1          rlf     XBUF,f          ; Shift data to Carry flag
            btfsc   STATUS,C
            goto    N2              ; data bit = 1
            goto    N3              ; data bit = 0
N2          movlw   TRISC
            movwf   FSR
            bsf     INDF,3          ; SDA hi
N3          bsf     SCL
            bcf     SCL
            bcf     SDA             ; SDA lo
            movlw   TRISC
            movwf   FSR
            bcf     INDF,3          ; SDA output
            decfsz  count,f         ; Dec count 1, skip next instruction if
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
count=0
            goto    N1              ; count <> 0
            return

;_____________________________________
;               Start bit
;-------------------------------------
start       bsf     STATUS,RP0      ; select bank 1
            bsf     TRISC,2         ; SDA = hi
            bsf     TRISC,3         ; SCL = hi
            bcf     STATUS,RP0      ; select bank 0
            nop
            nop
            nop
            nop
            bsf     STATUS,RP0      ; select bank 1
            bcf     TRISC,3         ; SDA lo
            nop
            nop
            nop
            nop
            bcf     TRISC,2         ; SCL lo
            bcf     STATUS,RP0      ; select bank 0
            return
;_____________________________________
;               Short delay
;-------------------------------------
SDel        movlw   0x10
            movwf   dt1
sd2         movlw   0xff
            movwf   dt2
sd1         decfsz  dt2
            goto    sd1
            decfsz  dt1
            goto    sd2
            return

            end
```

รูปที่ 4.16 โปรแกรมทดลองอ่านข้อมูลวินาทีจาก RTC

3. สั่งไมโครคอนโทรลเลอร์ทำงาน ถ้าใช้งานได้ แอลอีดีในโมดูลแสดงผลเป็นค่าของวินาที

ผลการทดลอง

แอลอีดีในโมดูลแสดงผลติดแสดงค่าวินาทีเป็นรหัส BCD โดย 4 บิตบนแสดง BCD ของตัวเลขวินาทีหลักสิบ และ 4 บิตล่างแสดงหลักหน่วย จาก 00 ถึง 59 แสดงว่าเชื่อมต่อกับ RTC แล้วรับส่งข้อมูลได้

2) การทดลองเชื่อมต่ออุปกรณ์ลำโพง

การเชื่อมต่อจะใช้สายสัญญาณ 1 บิต สำหรับส่งสัญญาณพัลส์ความถี่ประมาณ 1 kHz ไปให้ลำโพง ก็จะสามารถกำเนิดเสียงออกมาได้

ลำดับขั้นการทดลอง

1. เชื่อมต่อลำโพงเข้ากับโมดูลหลักดังรูปที่ 4.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. จากผังการทำงานโปรแกรมที่ได้ออกแบบไว้ ทำการเขียนโปรแกรมทดลองการใช้งานสร้างพัลส์ความถี่ประมาณ 1 kHz ส่งให้ลำโพง โปรแกรมแสดงดังรูปที่ 4.17

```
      list p=16f877
     include "p16f877.inc"
   __config _PWRTE_ON & _WDT_OFF & _LVP_OFF & _XT_OSC

            org     0x00
            goto    Initial
            org     0x04
            goto    IntRout         ; interrupt service routine

Initial     bsf     STATUS,RP0
            movlw   b'11010000'     ; prescaler devision 2
            movwf   OPTION_REG
            movlw   b'11101111'
            movwf   TRISC           ; speaker bit 4
            bcf     STATUS,RP0
            bcf     PORTC,4

Main        movlw   b'10100000'     ; enable T0IE, clear flag T0IF
            movwf   INTCON
            bsf     INTCON,T0IF     ; call to interrupt service routine
            goto    $               ; wait for interrupt

IntRout     bcf     INTCON,T0IF     ; clear interrupt flag
            movlw   0x06            ; counter for overflow => 250
            movwf   TMR0
            movlw   b'00010000'
            xorwf   PORTC,F         ; invert data for pulse generate
            retfie

            end
```

รูปที่ 4.17 โปรแกรมทดลองลำโพง ความถี่ 1 kHz

3. สั่งไมโครคอนโทรลเลอร์ทำงาน

ผลการทดลอง

ลำโพงเปล่งเสียงออกมา แสดงว่าการเชื่อมต่อกับอุปกรณ์ลำโพงใช้งานได้

4.2.9 การทดลองใช้งานโมดูลแปลงสัญญาณแอนะลอกและดิจิตอล

โมดูลนี้ประกอบไปด้วยวงจร 3 วงจร คือ วงจรปรับแรงดัน เพื่อสร้างสัญญาณแอนะลอกสำหรับทดลองการแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอล ซึ่งตัวแปลงสัญญาณจะมีอยู่แล้วในไมโครคอนโทรลเลอร์ PIC16F87X วงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอก 2 วงจรคือ วงจร R-2R Ladder และวงจร ใช้ไอซีเบอร์ DAC0808 ซึ่งจะแบ่งการทดลองออกเป็นข้อย่อยดังนี้

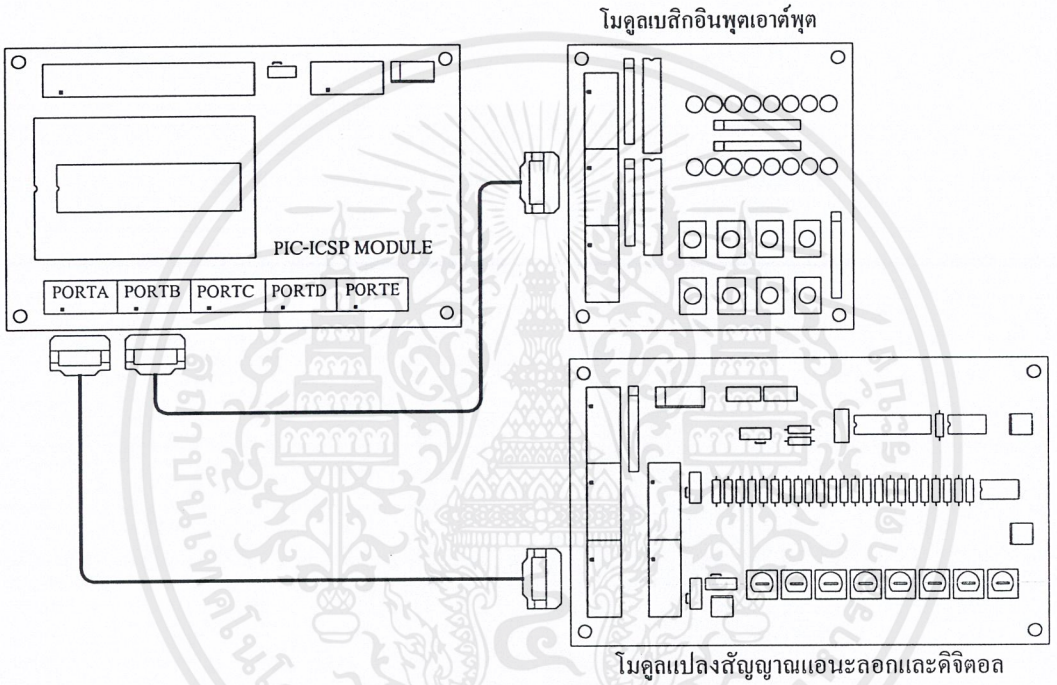
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) การทดลองแปลงสัญญาณแอนะลอกเป็นดิจิตอล

ทดลองเชื่อมต่อไมโครคอนโทรลเลอร์กับวงจรปรับแรงดัน เพื่อสร้างสัญญาณแอนะลอกเข้าที่ขา AN0 คือพอร์ต A0 ให้ไมโครคอนโทรลเลอร์แปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอลแล้วส่งออกมาแสดงผลที่โมดูลแสดงผลแบบแอลอีดี

ขั้นตอนการทดลอง

1. เชื่อมต่อวงจรปรับแรงดันเข้ากับโมดูลหลักและโมดูลแสดงผลดังรูปที่ 4.18



รูปที่ 4.18 การเชื่อมต่อวงจรปรับแรงดันเพื่อทดลองแปลงสัญญาณแอนะลอกเป็นดิจิตอล

2. จากผังการทำงานที่ได้ออกแบบไว้ เขียนโปรแกรมสั่งงานไมโครคอนโทรลเลอร์ เพื่อทำการแปลงสัญญาณแอนะลอกที่พอร์ต AN0 เป็นสัญญาณดิจิตอลส่งข้อมูลออกมาที่พอร์ต B เพื่อแสดงผล แสดงดังโปรแกรมรูปที่ 4.19

```
List p=16f877
  include "p16f877.inc"
 __config _WDT_OFF & _PWRTE_ON & _LVP_OFF & _XT_OSC

            org     0x00
            bsf     STATUS,RP0
            movlw   0x0e
            movwf   ADCON1          ; Left Justified, Analog Bit0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            movlw   0xff
            movwf   TRISA
            clrf    TRISB
            bcf     STATUS,RP0
            movlw   b'01000001'    ; Fosc/8, Channel, A/D On
            movwf   ADCON0
            clrf    PORTB

Start       btfsc   ADCON0,GO_DONE ; check DONE
            Goto    Start

            Movf    ADRESH,W       ; read adc result
            Movwf   PORTB          ; mov to LED display
            bsf     ADCON0,GO_DONE ; set GO
            goto    Start

            end
```

รูปที่ 4.19 โปรแกรมทดลองแปลงสัญญาณแอนะลอกเป็นดิจิตอล

3. โปรแกรมลงบนไมโครคอนโทรลเลอร์แล้วสั่งทำงาน ทดลองปรับตัวต้านทานปรับค่าชื่อ AN0 เพื่อปรับแรงดันแอนะลอกให้ไมโครคอนโทรลเลอร์แปลงสัญญาณเป็นดิจิตอลแล้วสังเกตผลที่แอลอีดีในโมดูลแสดงผล

ผลการทดลอง

แอลอีดีในโมดูลแสดงผลแสดงค่าสัญญาณดิจิตอลเป็นเลขฐานสอง เมื่อปรับตัวต้านทานไปด้านขวา แอลอีดีจะแสดงค่าเพิ่มมากขึ้นจนเป็น FF เมื่อปรับไปด้านซ้าย ค่าก็จะลดลง แสดงว่าการทดลองแปลงสัญญาณแอนะลอกเป็นดิจิตอลใช้งานได้

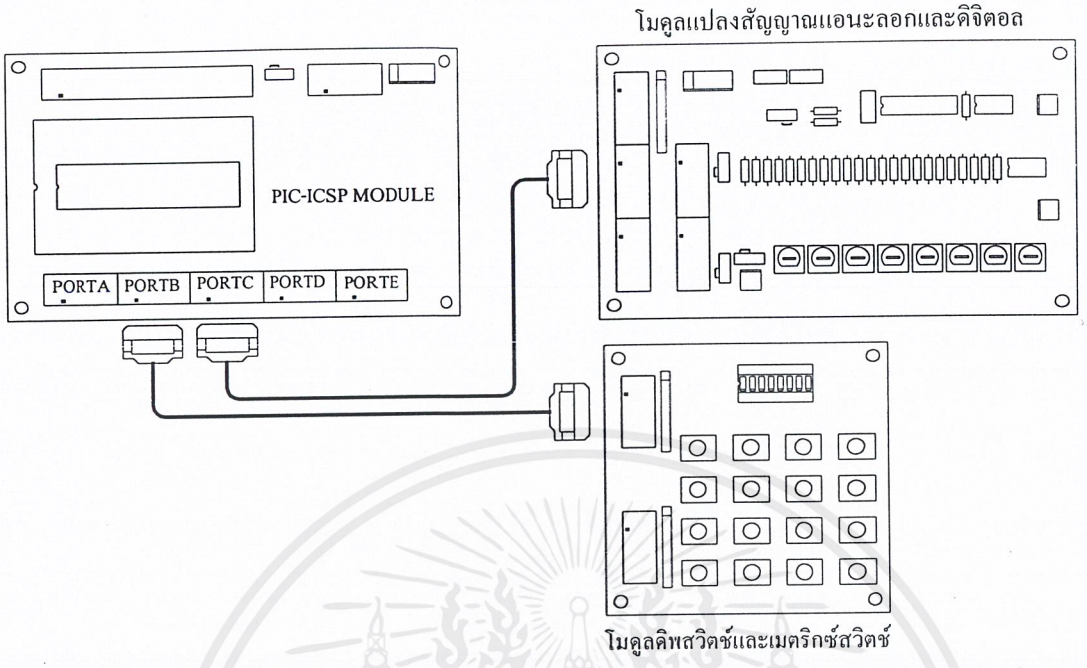
2) การทดลองแปลงสัญญาณดิจิตอลเป็นแอนะลอกด้วยวงจร R-2R Ladder

สามารถแปลงสัญญาณดิจิตอลได้ทั้ง 8 บิต หรือ10 บิต เพื่อเป็นการรองรับสัญญาณดิจิตอลที่ไมโครคอนโทรลเลอร์แปลงมาจาก ADC ภายใน ซึ่งมีขนาด 10 บิต จะใช้สัญญาณดิจิตอล 8 บิต จากโมดูลดิพสวิตช์เพื่อทดลองการทำงานโดยใช้มัลติมิเตอร์วัดสัญญาณแอนะลอกที่แปลงได้

ลำดับขั้นการทดลอง

1. เมื่อต่อวงจร R-2R Ladder เข้ากับโมดูลหลัก และโมดูลดิพสวิตช์ ดังรูปที่ 4.20 และจ่ายแรงดันจากโมดูลภาคจ่ายไฟเข้าที่คอนเนคเตอร์ชื่อ SUPPLY ของโมดูลแปลงสัญญาณแอนะลอกและดิจิตอลด้วย เนื่องจากต้องจ่ายแรงดัน ±12V ให้แก่ออปแอมป์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.20 การเชื่อมต่อวงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอกแบบ R-2R Ladder

2. จากผังการทำงานที่ได้ออกแบบไว้ เขียนโปรแกรมทดลองการทำงาน ส่งข้อมูลดิจิตอลจากดิพสวิตช์ไปยังวงจร R-2R Ladder เพื่อแปลงเป็นสัญญาณแอนะลอก ดังรูปที่ 4.21

```
list p=16f877
  include "p16f877.inc"
  __config _WDT_OFF & _PWRTE_ON & _LVP_OFF & _XT_OSC

            org     0x00
            bsf     STATUS,RP0
            movlw   0xff
            movwf   TRISB
            clrf    TRISC
            bcf     STATUS,RP0

Start       movf    PORTB,W         ; read dip switch
            Movwf   PORTC           ; mov digital signal to DAC
            goto    Start

            end
```

รูปที่ 4.21 โปรแกรมรับข้อมูลดิจิตอลจากดิพสวิตช์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.โปรแกรมลงไมโครคอนโทรลเลอร์แล้วสั่งทำงาน จากนั้นใช้มัลติมิเตอร์วัดแรงดันสัญญาณแอนะลอกที่คอนเนคเตอร์ชื่อ R-2R OUT แล้วทดลองปรับดิพสวิตช์ หลายๆ ค่า เพื่อดูขนาดของสัญญาณเอาต์พุต

ผลการทดลอง

เมื่อปรับดิพสวิตช์ตามค่าเลขฐานสอง เพิ่มค่าขึ้นเรื่อยๆ แรงดันเอาต์พุตของวงจรค่าเพิ่มขึ้นจนถึงค่าที่มากที่สุดคือ FF จะให้แรงดันเอาต์พุตประมาณ 10 V แสดงว่าวงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอกแบบ R-2R Ladder ทำงานได้

3) การทดลองวงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอกด้วยไอซี DAC0808

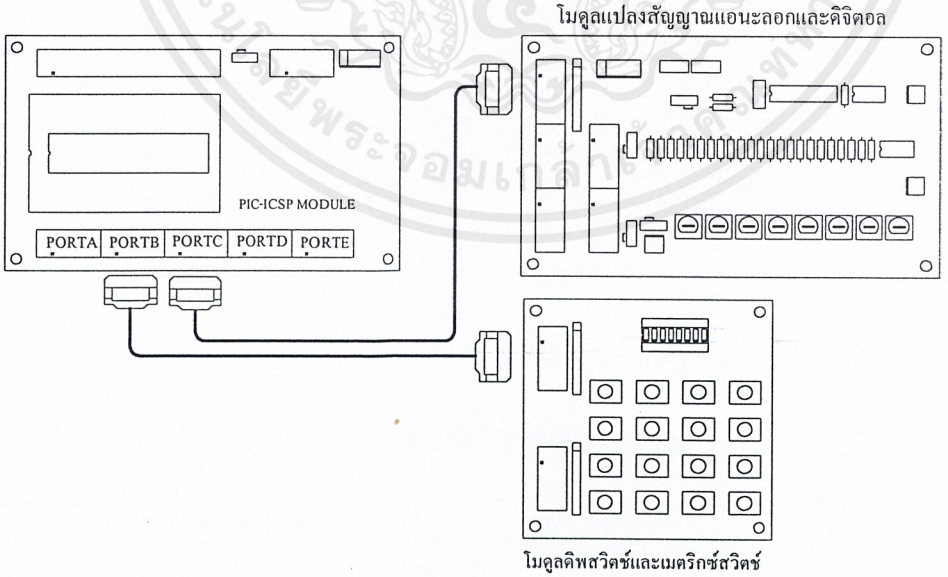
ไอซีเบอร์นี้สามารถแปลงสัญญาณดิจิตอลขนาด 8 บิตเป็นแอนะลอกได้ โดยการทดลองใช้งานจะทำเหมือนกับวงจรแบบ R-2R Ladder ที่ได้กล่าวมาแล้ว

ลำดับขั้นการทดลอง

1. เชื่อมต่อวงจรแปลงสัญญาณแอนะลอกด้วยไอซี DAC0808 กับโมดูลหลักและดิพสวิตช์แสดงดังรูปที่ 4.22 พร้อมจ่ายไฟให้โมดูลแปลงสัญญาณ

2. โปรแกรมที่ใช้สามารถใช้โปรแกรมเดียวกันกับการทดลองวงจร R-2R Ladder ได้ตามโปรแกรมรูปที่ 4.21 ทำการโปรแกรมลงบนไมโครคอนโทรลเลอร์แล้วสั่งทำงาน

3. ทำการวัดแรงดันเอาต์พุตของวงจรที่คอนเนคเตอร์ชื่อ DAC0808 OUT ด้วยมัลติมิเตอร์แล้วทดลองปรับดิพสวิตช์ เพื่อสร้างสัญญาณดิจิตอล สังเกตแรงดันที่ได้



รูปที่ 4.22 การเชื่อมต่อวงจร DAC0808 เข้ากับโมดูลหลักและดิพสวิตช์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลอง

เมื่อทดลองปรับดิพสวิตช์เป็นเลขฐานสองค่าต่างๆ จะให้แรงดันออกมาตามค่าที่ปรับ โดยค่าสูงสุดเมื่อปรับดิพสวิตช์เป็น FF ให้เอาต์พุตที่แรงดันประมาณ 5.2 โวลต์ แสดงว่าวงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอกแบบใช้ไอซี DAC0808 ทำงานได้

4.2.10 การทดลองใช้งานโมดูลสื่อสารข้อมูลอนุกรม

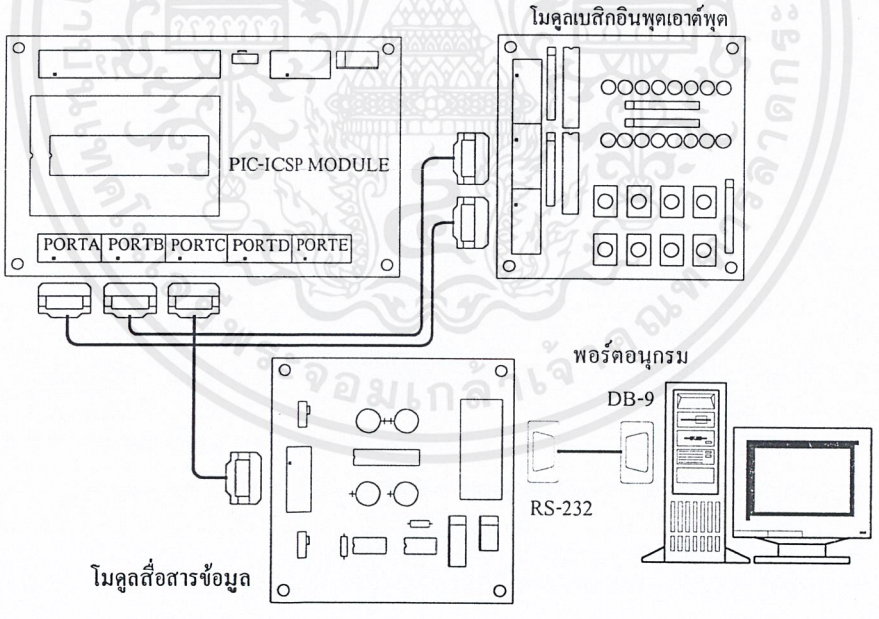
ในโมดูลจะประกอบไปด้วยวงจรสื่อสารข้อมูลอนุกรมแบบ RS-232 และแบบ RS485 โดยจะแบ่งการทดลองออกเป็นสองส่วนดังนี้

1) การทดลองสื่อสารข้อมูลอนุกรมแบบ RS-232

ทดลองโดยการเชื่อมต่อรับส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์กับไมโครคอนโทรลเลอร์ โดยแสดงผลข้อมูลเป็นรหัสแอสกีบนโมดูลแสดงผลแอลอีดี

ลำดับขั้นการทดลอง

1. เชื่อมต่อโมดูลสื่อสารข้อมูลอนุกรมแบบ RS-232 เข้ากับโมดูลหลัก และพอร์ตอนุกรมของคอมพิวเตอร์ แสดงดังรูปที่ 4.23



รูปที่ 4.23 การเชื่อมต่อส่งสัญญาณข้อมูลอนุกรมแบบ RS-232 กับคอมพิวเตอร์

2. จากผังการทำงานที่ได้ออกแบบไว้ เขียนโปรแกรมทดลองการทำงาน ส่งข้อมูลระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์ แสดงดังรูปที่ 4.24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   list p=16f877
     include "p16f877.inc"
   __config _PWRTE_ON & _WDT_OFF & _LVP_OFF & _XT_OSC
tempSw      equ      0x20

            org      0x00
            bsf      STATUS,RP0      ; select bank 1
            movlw    0x06
            movwf    ADCON1
            movlw    0xff
            movwf    TRISA
            clrf     TRISB
            movlw    0x19            ; BAUD rate 2400 , (25 decimal)
            movwf    SPBRG
            clrf     TXSTA           ; 8 bits data, no, 1 bit stop
            bcf      STATUS,RP0      ; select bank0
            bsf      RCSTA,SPEN      ; Asynchronous serial port enable
            bsf      RCSTA,CREN      ; continuous receive
            bsf      STATUS,RP0      ; select bank1
            bsf      TXSTA,TXEN      ; Transmit enable
            bcf      STATUS,RP0      ; select bank0
            clrf     PORTB           ; clear LED display
Start       call     Receive
            call     ScanSW
            addlw    0x00
            btfss    STATUS,Z        ; check press ?
            call     Sender          ; yes
            goto     Start           ; no
;-------------- transmission --------------
Sender      movwf    PORTB           ; display
            Movwf    TXREG
            Movlw    TXSTA
            Movwf    FSR
wait1       btfss    INDF,1
            goto     wait1
            return                   ; TXREG empty or TRMT=1
;--------------- reception ----------------
Receive     btfss    PIR1,RCIF       ; Check RCIF bit on PIR1 register
(Peripheral interrupts)
            Return                   ; RCREG empty or RCIF=0
            movf     RCREG,W         ; RCREG full or RCIF=1
            movwf    PORTB
            return

ScanSW      movlw    0
            Btfss    PORTA,0
            Movlw    30              ; ascii of "0"
            Btfss    PORTA,1
            Movlw    31              ; ascii of "1"
            Btfss    PORTA,2
            Movlw    32
            Btfss    PORTA,3
            Movlw    33
            Btfss    PORTA,4
            Movlw    34
            Btfss    PORTA,5
            Movlw    35

            Subwf    tempSw,F
            Movwf    tempSw          ; store old switch
            Btfsc    STATUS,C
            Retlw    0               ; if no press return 0
            Return                   ; if press return data of new       switch

            end
```

รูปที่ 4.24 โปรแกรมทดลองรับส่งข้อมูลกับคอมพิวเตอร์อนุกรมแบบ RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. โปรแกรมลงบนไมโครคอนโทรลเลอร์ แล้วสั่งทำงาน

4. เปิดโปรแกรม MSCOM.EXE จากคอมพิวเตอร์ เพื่อใช้ในการสื่อสารข้อมูลอนุกรมแบบ RS-232 กับอุปกรณ์ต่อพ่วง

5. สั่งไมโครคอนโทรลเลอร์ทำงาน โดยทดลองกดคีย์บอร์ดคอมพิวเตอร์ เพื่อส่งข้อมูลไปแสดงผลยังแอลอีดี แล้วกดสวิตช์ในโมดูลเพื่อส่งข้อมูลไปยังคอมพิวเตอร์

ผลการทดลอง

เมื่อทดลองทำการกดคีย์บอร์ด แอลอีดีที่อยู่ในโมดูลแสดงผลติดเป็นรหัสแอสกีของอักษรที่กดคีย์บอร์ด และเมื่อกดสวิตช์บนโมดูลโปรแกรมที่ใช้รับข้อมูลในคอมพิวเตอร์ก็แสดงตัวเลขตามสวิตช์ที่กด แสดงว่าการรับส่งข้อมูลอนุกรมแบบ RS-232 กับคอมพิวเตอร์สามารถทำงานได้

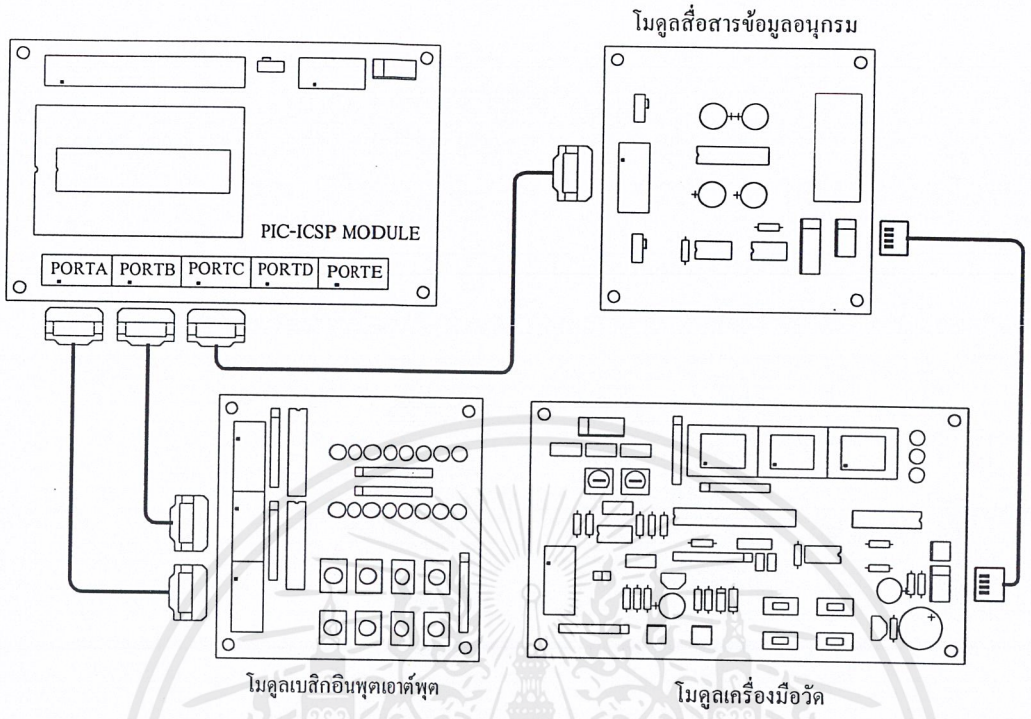
2) การทดลองสื่อสารข้อมูลอนุกรมแบบ RS-485

ใช้โมดูลเครื่องมือวัดร่วมในการทดลอง เพราะในโมดูลดังกล่าวมีชุดทดลองรับส่งข้อมูลแบบ RS-485 อยู่ จึงนำมาใช้รับส่งข้อมูลกับโมดูลสื่อสารข้อมูลอนุกรมเพื่อการทดลองการทำงานไปพร้อมๆ กัน

ลำดับขั้นการทดลอง

1. เชื่อมต่อโมดูลต่างๆ ดังรูปที่ 4.25 เชื่อมต่อสายสัญญาณเข้าที่คอนเนคเตอร์ชื่อ RS-485 จากโมดูลสื่อสารข้อมูลอนุกรมกับโมดูลเครื่องมือวัด แล้วจ่ายไฟเลี่ยงเข้าที่โมดูลหลัก และโมดูลเครื่องมือวัด จากนั้นเซ็ตจั๊มเปอร์ในโมดูลสื่อสารข้อมูลอนุกรม จั๊มเปอร์ชื่อ RS-232 เซ็ตไว้ที่ OFF จั๊มเปอร์ชื่อ SELECT เซ็ตไว้ที่ RS-485

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.25 การเชื่อมต่อโมดูลทดลองส่งสัญญาณข้อมูลแบบ RS-485

2. จากผังการทำงานที่ได้ออกแบบไว้ เขียนโปรแกรมทดลองการรับส่งข้อมูลดังรูปที่ 4.26

```
list p=16f877
    include "p16f877.inc"
  __config _PWRTE_ON & _WDT_OFF & _LVP_OFF & _XT_OSC

  #define   DIR       PORTC,5
            cblock    0x20
            tempSw, tmDy1, tmDy2
            endc

            org       0x00
            bsf       STATUS,RP0    ; select bank 1
            movlw     0x06
            movwf     ADCON1
            movlw     0xff
            movwf     TRISA
            clrf      TRISB
            bcf       TRISC,5       ; dir
            movlw     0x19          ; BAUD rate 2400 , (25 decimal)
            movwf     SPBRG
            clrf      TXSTA         ; 8 bits data, no parity, 1 stop
            bcf       STATUS,RP0    ; select bank0
            clrf      RCSTA
            bsf       RCSTA,SPEN    ; Asynchronous serial port enable
            bsf       RCSTA,CREN    ; continuous receive
            movlw     TXSTA
            movwf     FSR
            bsf       INDF,TXEN     ; Transmit enable
            clrf      tempSw
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
main        call    RX
            call    scank
            addlw   0
            btfsc   STATUS,Z
            goto    main
            call    TX
            goto    main
;_____________________________________________
;                scan key
;---------------------------------------------
scank       movlw   0
            btfss   PORTA,0
            movlw   1
            btfss   PORTA,1
            movlw   2
            btfss   PORTA,2
            movlw   3
            btfss   PORTA,3
            movlw   4
            btfss   PORTA,4
            movlw   5
            btfss   PORTA,5
            movlw   6

            subwf   tempSw,F
            movwf   tempSw      ; store old switch
            btfsc   STATUS,C
            retlw   0           ; if no press return 0
            return              ; if press return data of new switch

;_____________________________________________
;               reception
;---------------------------------------------
RX          bcf     DIR         ; direction RX
            Btfss   PIR1,RCIF
            Return
            movf    RCREG,W
            movwf   PORTB
            return

;_____________________________________________
;              Transmission
;---------------------------------------------
; input     W
; output    line RS485
;---------------------------------------------
TX          bsf     DIR         ; direction TX
            Movwf   TXREG
TXwait      movlw   TXSTA
            Movwf   FSR
            Btfss   INDF,TRMT
            goto    Txwait
            call    Delay
            return

Delay       movlw   0xff
            Movwf   tmDy1
D1          movlw   0xff
            Movwf   tmDy2
D2          decfsz  tmDy2,F
            goto    D2
            decfsz  tmDy1,F
            goto    D1
            return
            end
```

รูปที่ 4.26 โปรแกรมทดลองรับส่งข้อมูลอนุกรมแบบ RS-485

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. โปรแกรมลงบนไมโครคอนโทรลเลอร์แล้วสั่งทำงาน ทดลองโดยเลือกโหมดรับข้อมูลอนุกรม RS-485 ในโมดูลเครื่องมือวัดโดยกดสวิตช์ชื่อ RX แล้วกดสวิตช์โมดูลเบสิกอินพุตเอาต์พุตเพื่อส่งข้อมูลไปยังโมดูลเครื่องรับ จากนั้นก็กดสวิตช์ชื่อ TX ในโมดูลเครื่องมือวัดเพื่อส่งข้อมูลที่กำลังแสดงผลอยู่ไปยังแอลอีดีในโมดูลเบสิกอินพุตเอาต์พุต

ผลการทดลอง

เมื่อกดสวิตช์บนโมดูลเบสิกอินพุตเอาต์พุต ปรากฏตัวตามสวิตช์ที่กดที่ส่วนแสดงผลของโมดูลเครื่องมือวัด พร้อมมีเสียงบี๊บ และเมื่อลองกดปุ่ม TX ในโมดุลเครื่องมือวัด แอลอีดีในโมดูลแสดงผลก็แสดงค่าเป็นตัวเลขสุดท้ายของตัวเลขที่ปรากฏอยู่ในโมดูลเครื่องมือวัด แสดงว่าการรับส่งข้อมูลแบบ RS-485 ทำงาน โมดูลใช้งานได้

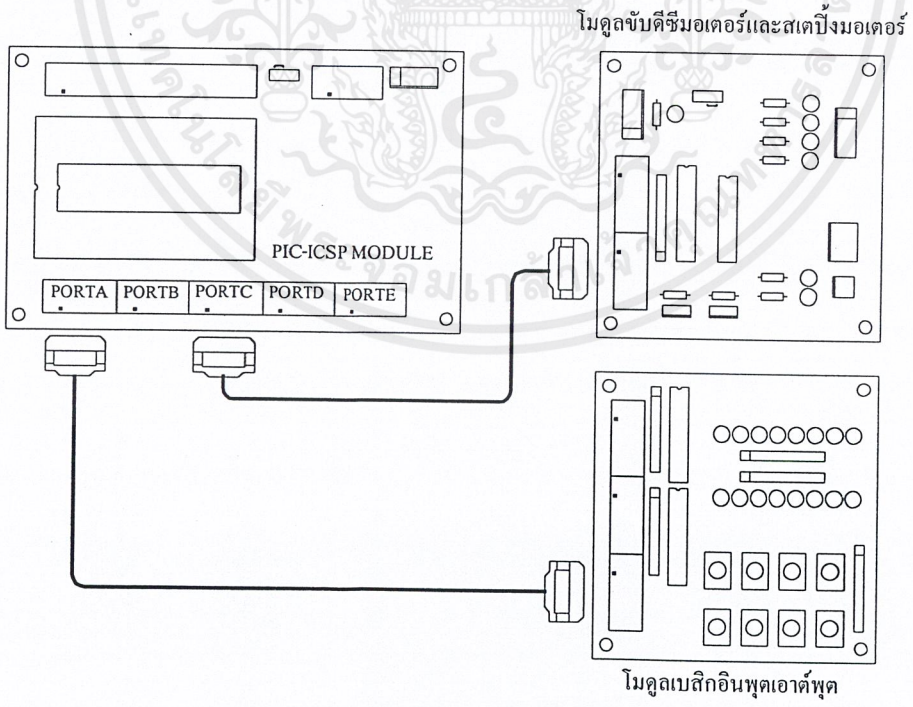
4.2.11 การทดลองใช้งานโมดูลขับดีซีมอเตอร์และสเตปปิ้งมอเตอร์

1) การทดลองวงจรขับดีซีมอเตอร์

การควบคุมดีซีมอเตอร์ จะใช้สายข้อมูล 2 บิต เพื่อควบคุมการหมุน คือหมุนซ้าย หมุนขวา และหยุด และสามารถเลือกแรงดันที่ใช้ขับได้ 2 ค่าคือ 5 โวลต์ กับ 12 โวลต์

ลำดับขั้นการทดลอง

1. เชื่อมต่อโมดูลสำหรับทดลองขับดีซีมอเตอร์ตามรูปที่ 4.27



รูปที่ 4.27 การเชื่อมต่อโมดูลขับดีซีมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. จากผังการทำงานที่ได้ออกแบบไว้ เขียนโปรแกรมควบคุมดีซีมอเตอร์ตามรูปที่ 4.28

```
 list p=16f877
  include "p16f877.inc"
__CONFIG _WDT_OFF & _PWRTE_ON & _LVP_OFF & _XT_OSC

            org     0x00
            bsf     STATUS,RP0      ; Bank 1
            movlw   0x06
            movwf   ADCON1
            movlw   0xff
            movwf   TRISA           ; PORTA = Input
            movlw   b'11111001'
            movwf   TRISC           ; PORTC => Drive Motor , bit1=P1, bit2=P2
            bcf     STATUS,RP0      ; Bank 0
            clrf    PORTC

START       movf    PORTC,W
            Btfss   PORTA,0
            Movlw   0x02            ; 01 to (P1,P2)
            Btfss   PORTA,1
            Movlw   0x04            ; 10 to (P1,P2)
            Btfss   PORTA,2
            Movlw   0x00            ; 00 to (P1,P2)
            Btfss   PORTA,3
            Movlw   0x06            ; 11 to (P1,P2)
            Movwf   PORTC
            Goto    START

            End
```

รูปที่ 4.28 โปรแกรมควบคุมดีซีมอเตอร์

3. ทำการโปรแกรมลงไมโครคอนโทรลเลอร์แล้วสั่งทำงาน แล้วทดลองกดสวิตช์ ที่โมดูลเบสิกอินพุต จาก SW0 ถึง SW4 และทดลองใช้แรงดันสูง 12 โวลต์ขับมอเตอร์โดยป้อนแรงดันจากโมดูลแหล่งจ่ายเข้าที่ SUPPLY แล้วเซ็ตจั๊มเปอร์ชื่อ SELECT +V ไปที่ 12V แล้วทดลองกดสวิตช์อีกครั้ง

ผลการทดลอง

เมื่อกดสวิตช์ SW0 มอเตอร์หมุน ตามเข็มนาฬิกา สวิตช์ SW1 จะหมุนทวนเข็ม ส่วนสวิตช์ SW2 และ SW3 มอเตอร์หยุดหมุน และเมื่อทดลองใช้แรงดัน 12 โวลต์ควบคุม มอเตอร์หมุนเร็วขึ้น แสดงว่าการควบคุมดีซีมอเตอร์ทำงานได้

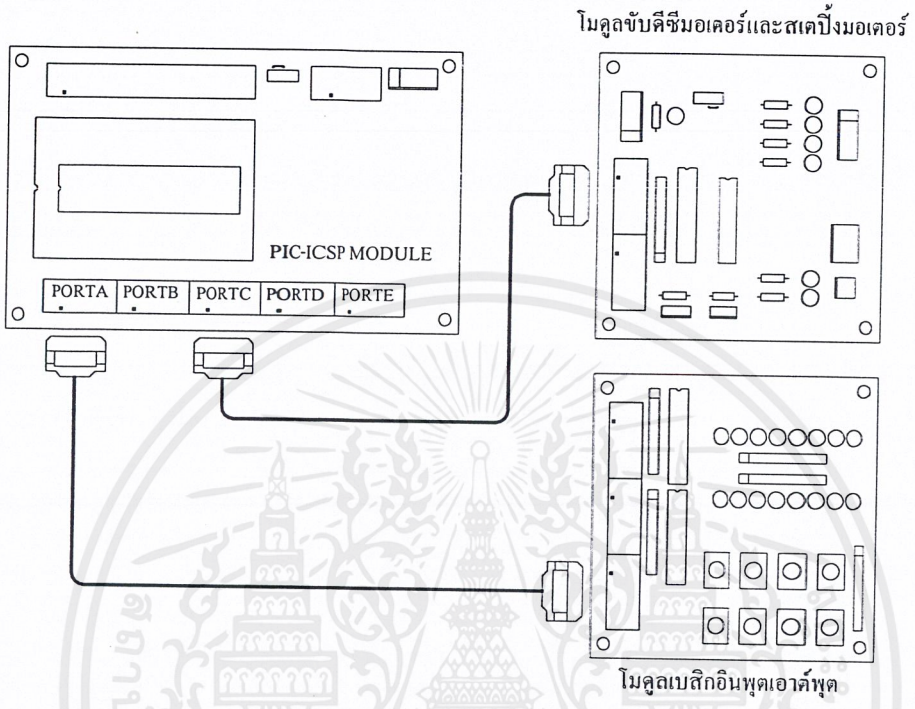
2) การทดลองวงจรขับสเตปปิ้งมอเตอร์

วงจรขับสเตปปิ้งมอเตอร์แบบ ยูนิโพลาร์ จะทดลองโดยการควบคุมการหมุนแบบเวฟ ทดลองการทำงานของโมดูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับขั้นการทดลอง

1. เชื่อมต่อโมดูลขับสเตปปิ้งมอเตอร์เพื่อทดลองการทำงานตามรูปที่ 4.29



รูปที่ 4.29 การเชื่อมต่อโมดูลขับสเตปปิ้งมอเตอร์

2. จากผังการทำงานที่ได้ออกแบบไว้ เขียนโปรแกรมควบคุมสเตปปิ้งมอเตอร์แบบเวฟ ตามโปรแกรมรูปที่ 4.30

```
      list p=16f877
    include "p16f877.inc"
__config _WDT_OFF & _PWRTE_ON & _LVP_OFF & _XT_OSC

COUNT1      equ     0x20
COUNT2      equ     0x21
SPEED       equ     0x22

            org     0x00
            bsf     STATUS,RP0      ; select bank 1
            movlw   0x06
            movwf   ADCON1
            movlw   0xff
            movwf   TRISA
            clrf    TRISC           ; PORTC Stepping motor
            bcf     STATUS,RP0
            movlw   0x80
            movwf   SPEED           ; initial speed
Start       movlw   0x10
            btfss   PORTA,0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 addwf     SPEED,F                ; if press SW0 => speed up
           btfss     PORTA,1
           subwf     SPEED,F      ; if press SW1 => speed down
           clrf      PORTC        ; clear all phase
           bsf       STATUS,C     ; phase1 set
Loop       rlf       PORTC,F      ; shift next phase
           btfsc     PORTC,4
           goto      Start
           call      Delay
           goto      Loop

Delay      movf      SPEED,W
           movwf     COUNT1
Delay1     movlw     0xFF
           movwf     COUNT2
Delay2     decfsz    COUNT2,1
           goto      Delay2
           decfsz    COUNT1,1
           goto      Delay1
           return

           end
```

รูปที่ 4.30 โปรแกรมควบคุมสเตปปิ้งมอเตอร์แบบเวฟ

3.โปรแกรมลงบนไมโครคอนโทรลเลอร์แล้วสั่งทำงาน แล้วทดลองปรับความเร็วในการหมุน โดยกดสวิตช์ SW0 สำหรับลดความเร็วและ SW1 สำหรับเพิ่มความเร็ว

ผลการทดลอง

เมื่อสั่งทำงาน มอเตอร์จะหมุน แล้วลองกดสวิตช์ปรับความเร็ว มอเตอร์ก็เปลี่ยนความเร็วในการหมุน แสดงว่าการควบคุมสเตปปิ้งมอเตอร์ใช้งานได้

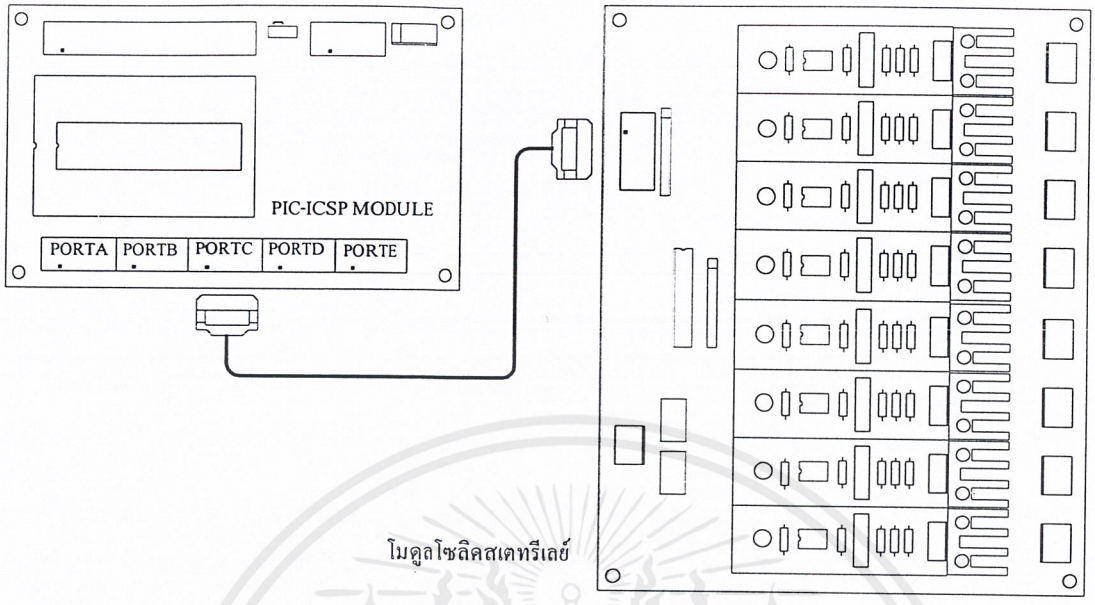
4.2.12 การทดลองใช้งานโมดูลโซลิดสเตทรีเลย์

การทดลองจะใช้ในการขับโหลดเป็นหลอดไฟ เพื่อทดลองการทำงานของวงจร

ลำดับขั้นการทดลอง

1. เชื่อมต่อโมดูลโซลิดสเตทรีเลย์เข้ากับโมดูลหลักตามรูปที่ 4.31 ต่อโหลดที่เป็นหลอดไฟเข้าที่คอนเนคเตอร์ LOAD ทั้ง 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.31 การเชื่อมต่อโมดูลโซลิดสเตทรีเลย์เข้ากับโมดูลหลัก

2. จากผังการทำงานที่ได้ออกแบบไว้ เขียนโปรแกรมควบคุมหลอดไฟ โดยโปรแกรมที่เขียนเป็นโปรแกรมควบคุมเป็นไฟวิ่ง ดังแสดงในรูปที่ 4.32

```
list p=16f877
  include "p16f877.inc"
__CONFIG _WDT_OFF & _PWRTE_ON & _LVP_OFF & _XT_OSC
            org     0x00
            bsf     STATUS,RP0      ; Bank 1
            clrf    TRISC
            bcf     STATUS,RP0      ; Bank 0
            clrf    PORTC           ; off light
            bsf     STATUS,C
Start       rlf     PORTC,F         ; shift from carry flag for run light
            call    Delay
            goto    Start
Delay       movlw   0xff
            movwf   0x20
Loop2       movlw   0xff
            movwf   0x21
Loop1       decfsz  0x21,F
            goto    Loop1
            decfsz  0x20,F
            goto    Loop2
            return

            end
```

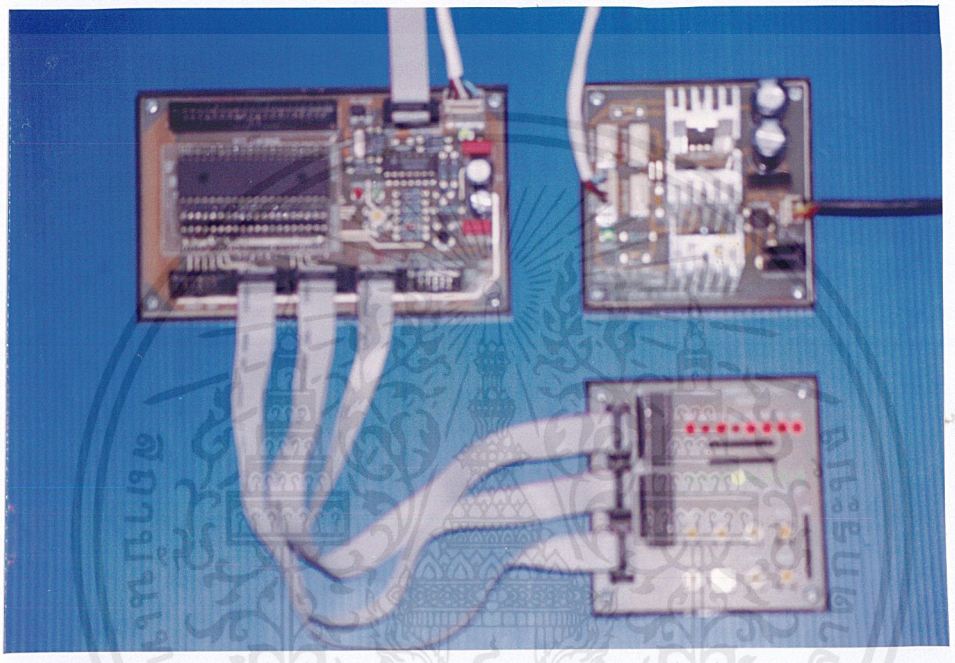
รูปที่ 4.32 โปรแกรมควบคุมโซลิดสเตทรีเลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. โปรแกรมลงบนไมโครคอนโทรลเลอร์แล้วสั่งการทำงาน

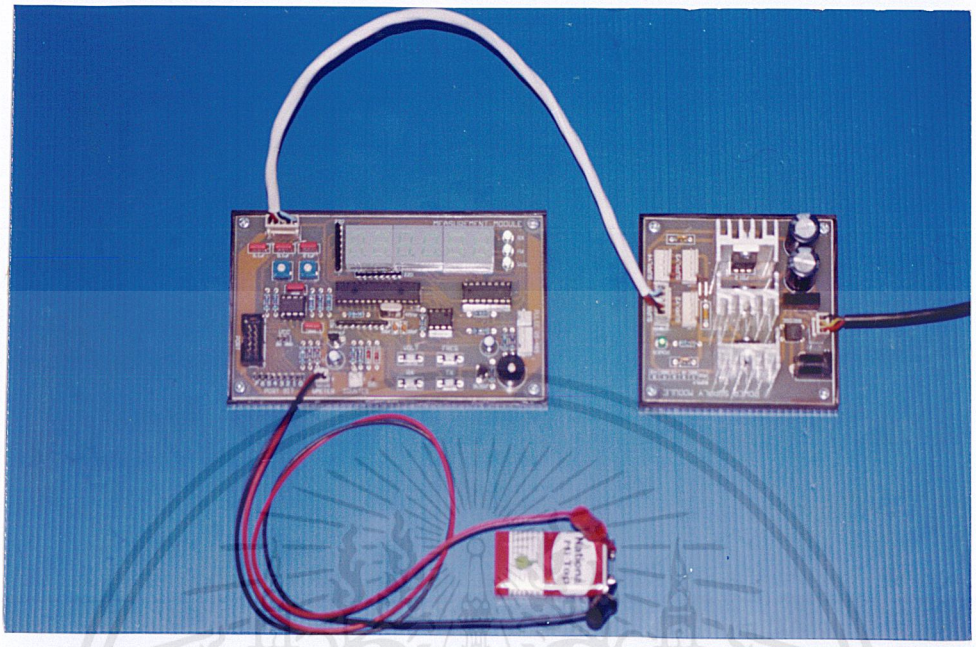
ผลการทดลอง

เมื่อสั่งโปรแกรมทำงาน หลอดไฟก็ติด พร้อมกับแอลอีดีแสดงผลการติด เมื่อเปลี่ยนหลอดไฟไปต่อยังคอนเนคเตอร์โหลดต่อไปก็ติดทั้งหมดทุกตัว แสดงว่าโมดูลโซลิดสเตทรีเลย์ใช้งานได้

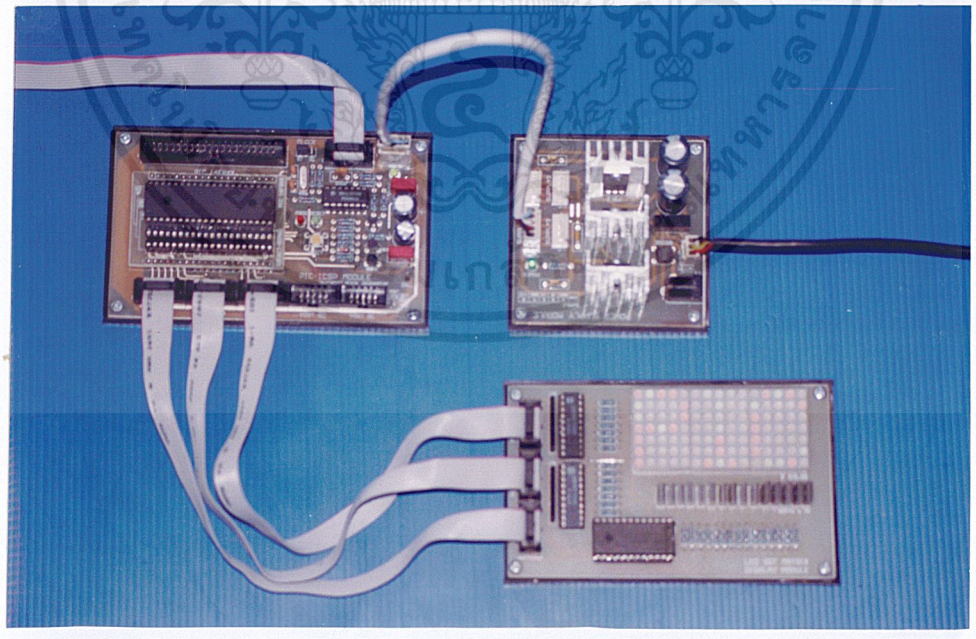


รูปที่ 4.33 ผลการทดลองโมดูลเบสิกอินพุตเอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

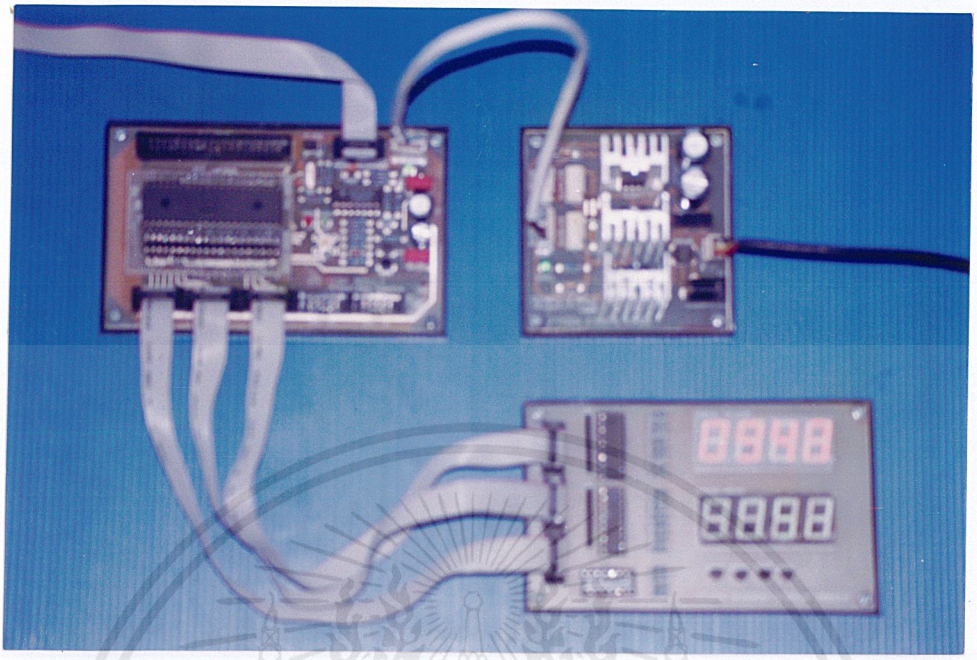


รูปที่ 4.34 ผลการทดลองโมดูลเครื่องมือวัด (เครื่องมือวัดแรงดัน)

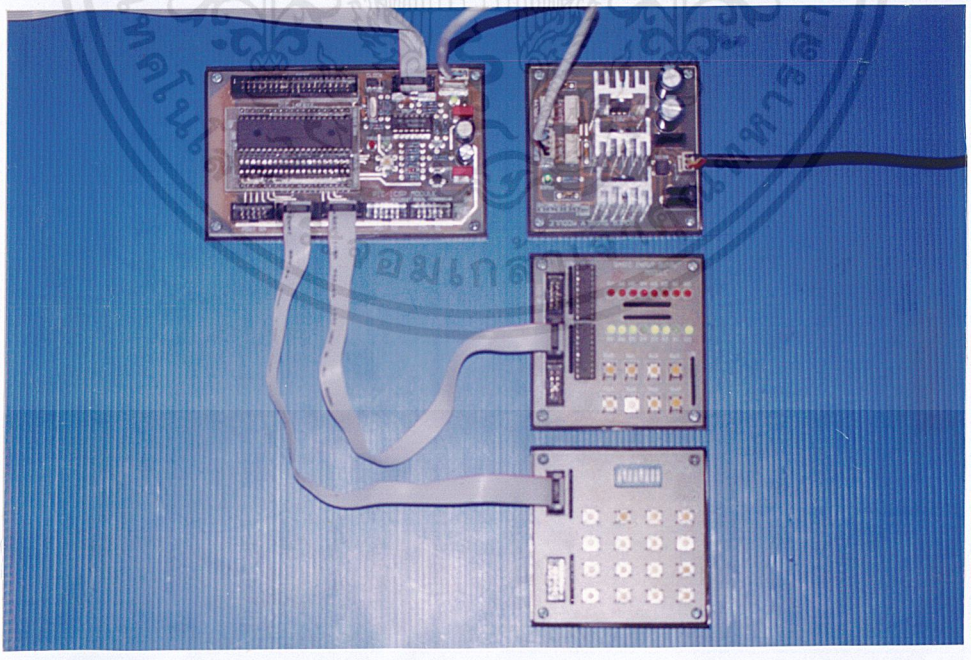


รูปที่ 4.35 ผลการทดลองโมดูลแอลอีดีดอตเมตริกซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

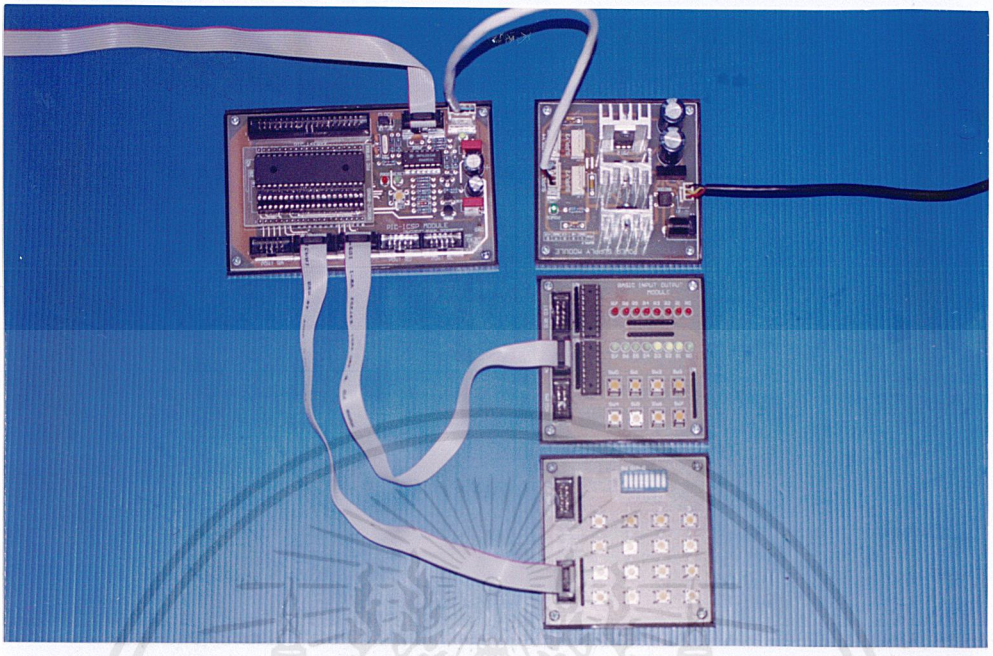


รูปที่ 4.36 ผลการทดลองโมดูลแสดงผลตัวเลข 7 ส่วน

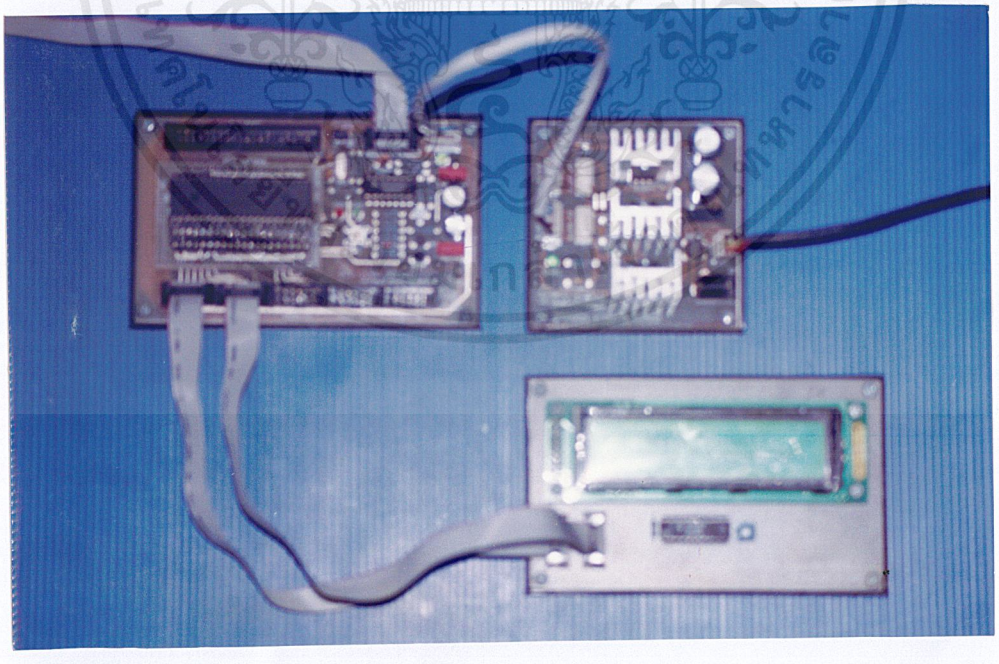


รูปที่ 4.37 ผลการทดลองดิพสวิตช์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

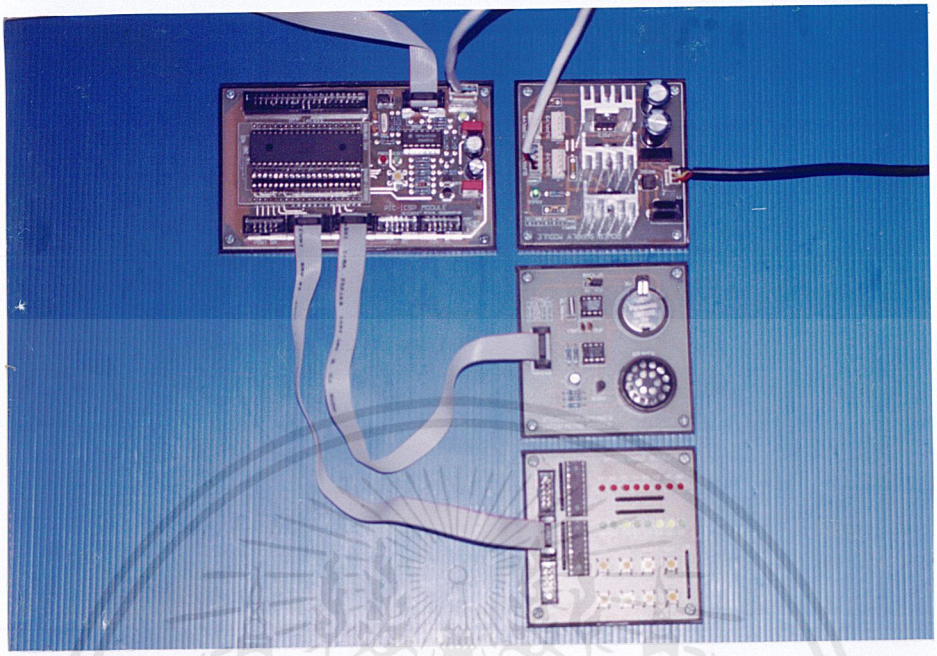


รูปที่ 4.38 ผลการทดลองเมตริกสวิตช์

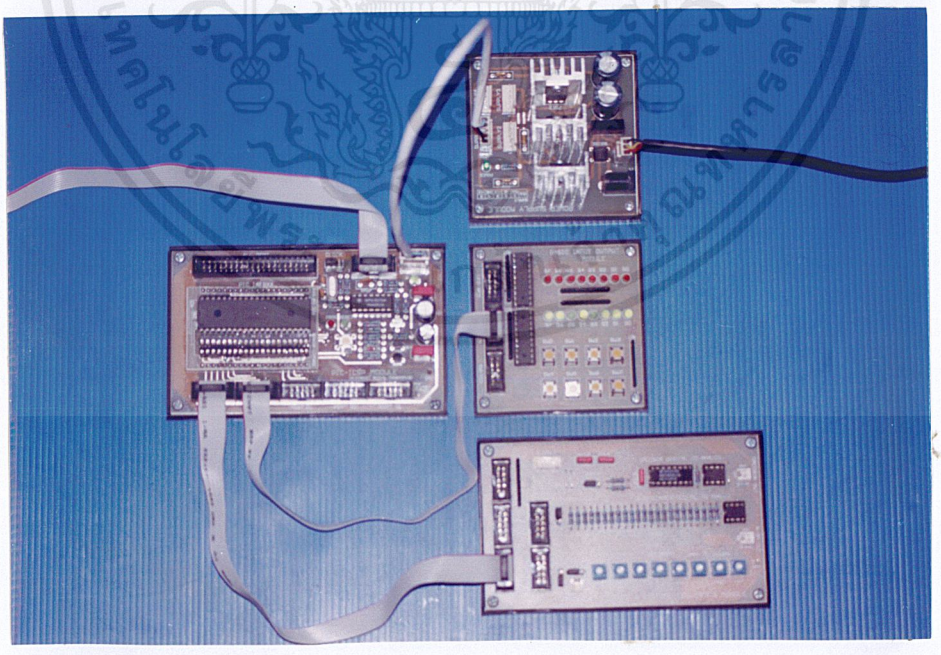


รูปที่ 4.39 ผลการทดลอง โมดูลแสดงผลแบบผลึกเหลว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

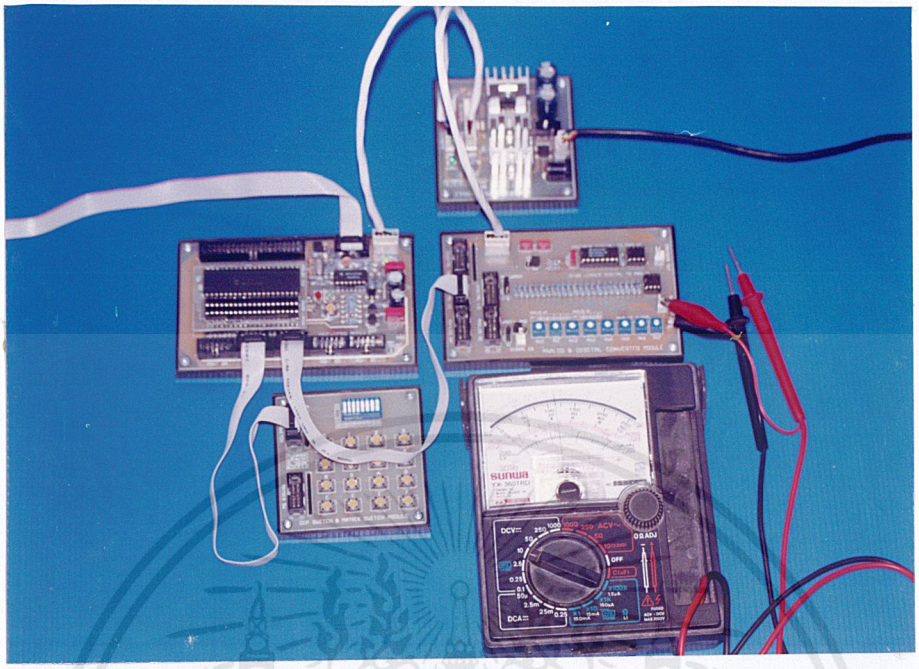


รูปที่ 4.40 ผลการทดลองอ่านข้อมูลวินาทีจาก RTC

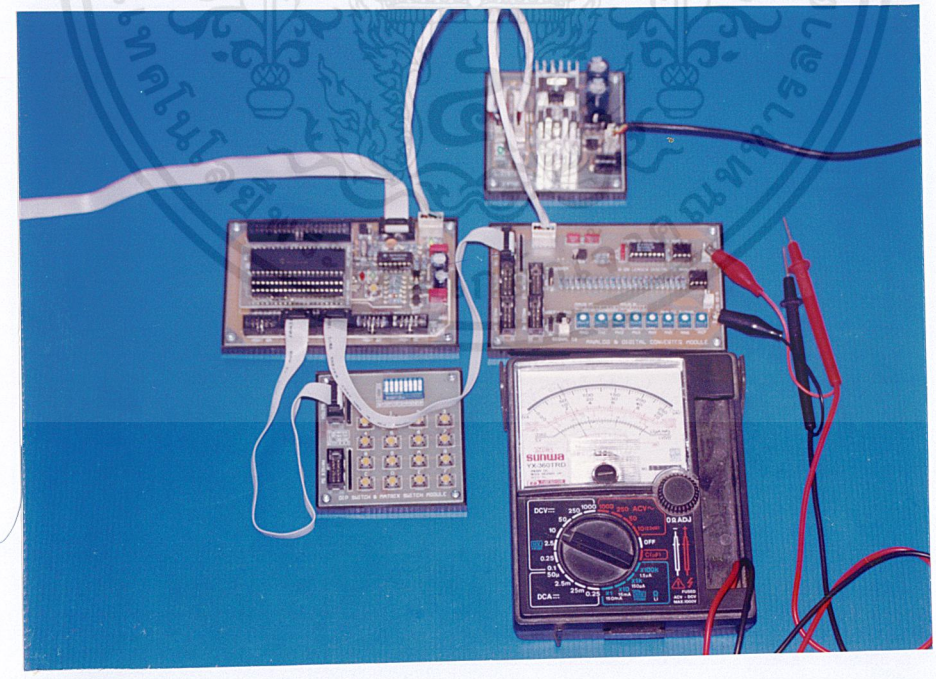


รูปที่ 4.41 ผลการทดลองแปลงสัญญาณแอนะลอกเป็นดิจิตอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

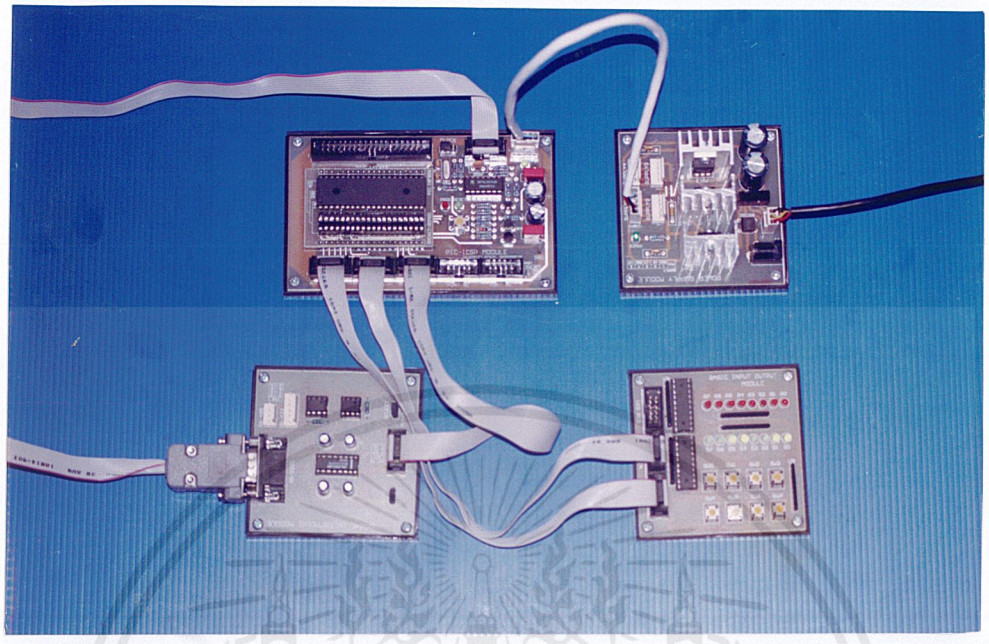


รูปที่ 4.42 ผลการทดลองวงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอกแบบ R-2R Ladder

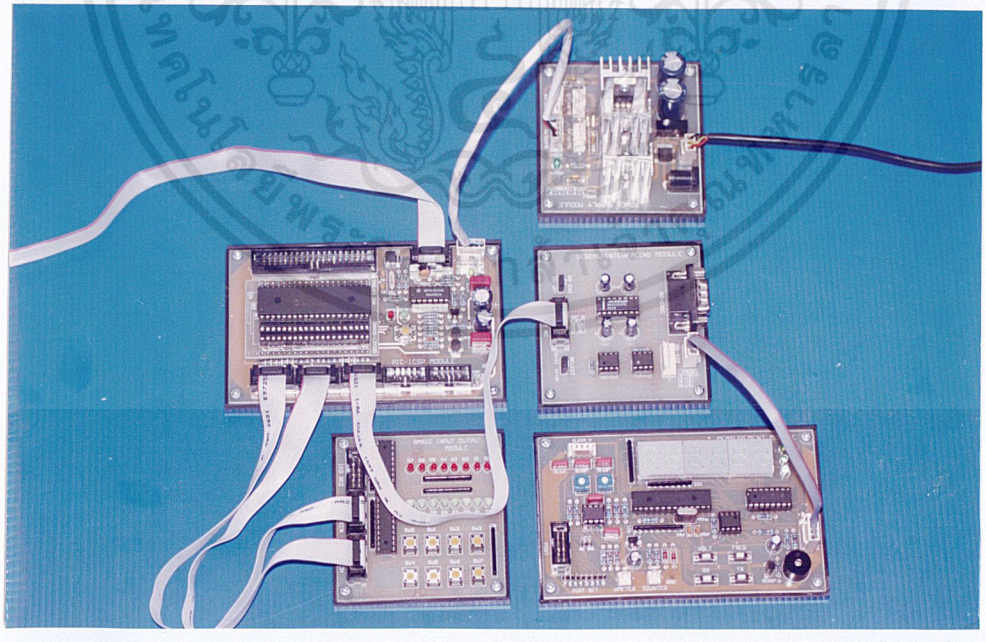


รูปที่ 4.43 ผลการทดลองวงจร DAC0808 เข้ากับโมดูลหลักและดิพสวิตช์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

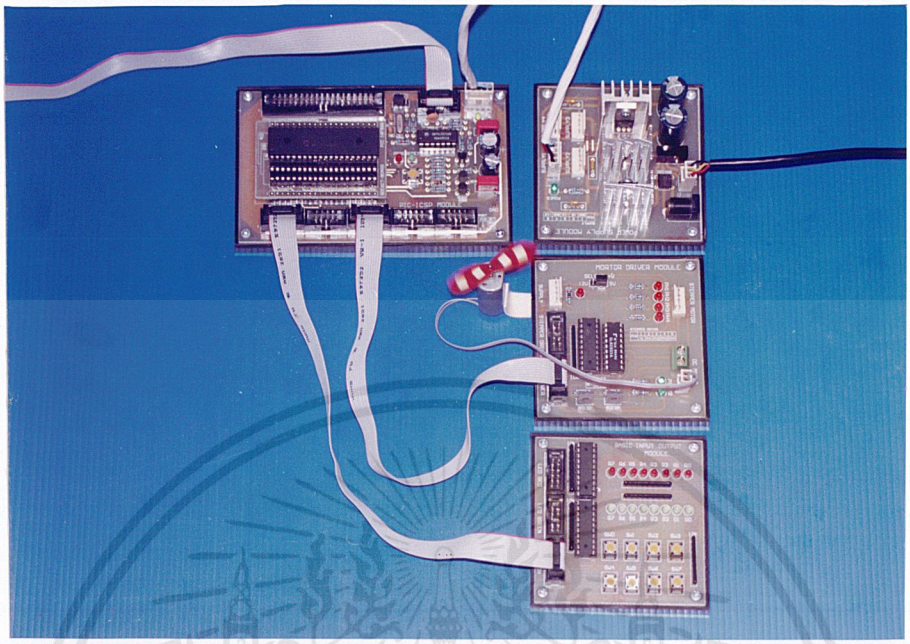


รูปที่ 4.44 ผลการทดลองรับส่งข้อมูลกับคอมพิวเตอร์อนุกรมแบบ RS-232

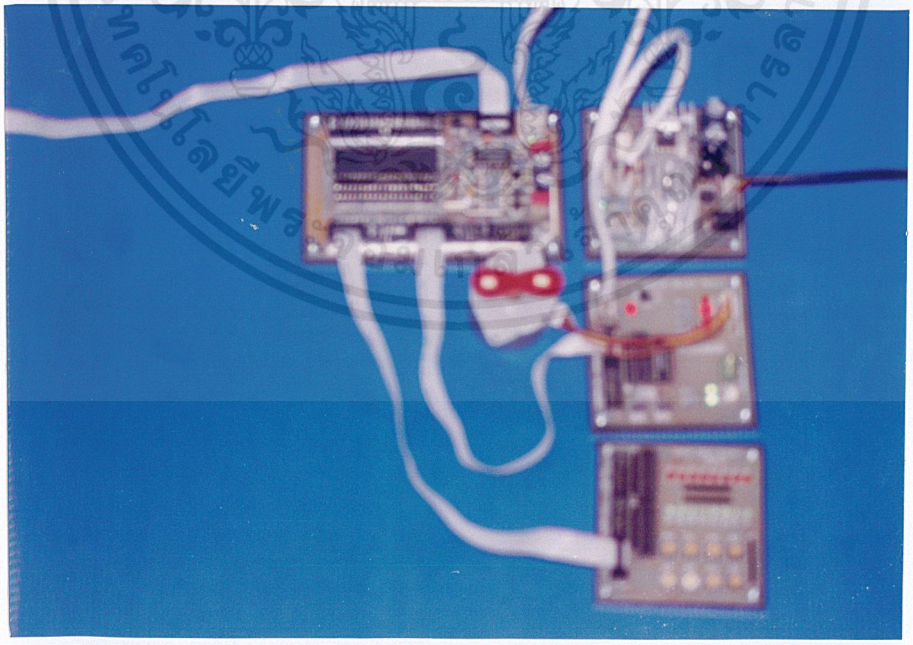


รูปที่ 4.45 ผลการทดลองส่งสัญญาณข้อมูลแบบ RS-485

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

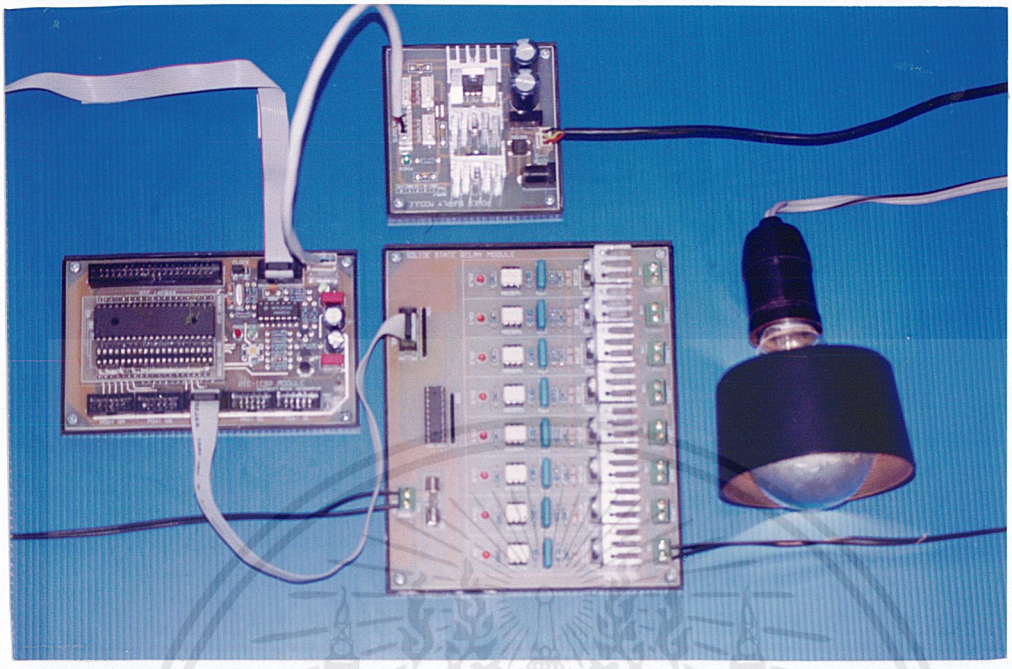


รูปที่ 4.46 ผลการทดลองควบคุมดีซีมอเตอร์



รูปที่ 4.47 ผลการทดลองควบคุมสเตปปิ้งมอเตอร์แบบเวฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.48 ผลการทดลองควบคุมโซลิดสเตทรีเลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุป ปัญหาแนวทางแก้ไขและพัฒนา

5.1 บทสรุป

ชุดปฏิบัติการทดลองไมโครคอนโทรลเลอร์ PIC16F87X สร้างขึ้นเพื่อใช้ศึกษาทฤษฎีและหลักการ ตลอดจนการนำไปใช้งาน ของไมโครคอนโทรลเลอร์ PIC16F87X ซึ่งได้กำหนดขอบเขตของโครงงานไว้ดังนี้

5.1.1 โมดูลสำหรับเชื่อมต่อของชุดปฏิบัติการทดลองไมโครคอนโทรลเลอร์ PIC16F87X

1) โมดูลหลัก PIC-ICSP
2) โมดูลเครื่องมือวัด
    2.1) โวลต์มิเตอร์ ±25 โวลต์
    2.2) เครื่องวัดความถี่ 2Hz - 1MHz
    2.3) ทดสอบการรับส่งข้อมูล RS-485
3) โมดูลเบสิกอินพุตเอาท์พุตพื้นฐาน
    3.1) แอลอีดีแสดงผลขนาด 8 บิต จำนวน 2 ชุด
    3.2) สวิตช์สร้างสภาวะลอจิกขนาด 8 บิต
4) โมดูลแสดงผลแอลอีดีดอตเมตริกซ์ ขนาด 7 × 15
5) โมดูลแสดงผลตัวเลข 7 ส่วน
    5.1) ชนิดคอมมอนคาโถดขนาด 4 หลัก
    5.2) ชนิดคอมมอนอาโนดขนาด 4 หลัก
6) โมดูลดิพสวิตช์ขนาด 8 บิต และเมตริกซ์สวิตช์ขนาด 4 × 4
7) โมดูลแสดงผลแบบผลึกเหลวขนาด 24 × 2
8) โมดูลเชื่อมต่ออุปกรณ์ RTC, EEPROM, ลำโพง
9) โมดูลแปลงสัญญาณแอนะลอกและดิจิตอล
    9.1) วงจรปรับแรงดัน 0 ถึง 5 โวลต์ 8 ชุด
    9.2) วงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอกแบบ R-2R Ladder ขนาด 10 บิต
    9.3) วงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอกแบบ DAC0808 ขนาด 8 บิต
10) โมดูลสื่อสารข้อมูลอนุกรม
    10.1)วงจรสื่อสารข้อมูลอนุกรม RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10.2) วงจรสื่อสารข้อมูลอนุกรม RS-485 / RS-422

11) โมดูลขับดีซีมอเตอร์และสเตปปิ้งมอเตอร์

12) โมดูลโซลิดสเตทรีเลย์ขนาด 8 ช่อง

13) โมดูลทดลองวงจร

14) โมดูลภาคจ่ายไฟ

5.1.2 ใบงานการทดลอง

ใบงานที่ 1 การใช้งานพอร์ตของ PIC16F876

ใบงานที่ 2 รีจิสเตอร์ STATUS

ใบงานที่ 3 การจัดการข้อมูลระดับบิต

ใบงานที่ 4 การใช้คำสั่งทางลอจิก (Logic) เพื่อจัดการข้อมูลระดับบิต

ใบงานที่ 5 การเลื่อนข้อมูล

ใบงานที่ 6 การทำงานของ PIC16F876 กับวงจรกำเนิดสัญญาณนาฬิกาแบบ RC และแบบคริสตอล

ใบงานที่ 7 การอินเตอร์รัพต์ PIC16F876

ใบงานที่ 8 การใช้งานไทเมอร์เคาน์เตอร์ภายใน PIC16F876 (Timer/Counter)

ใบงานที่ 9 การทำงานในโหมดสลีป (Sleep mode) ของ PIC16F876

ใบงานที่ 10 การเชื่อมต่อ PIC16F876 กับจอแสดงผลแบบผลึกเหลว (LCD)

ใบงานที่ 11 การใช้งาน PIC16F876 ขับ LED ตัวเลข 7 ส่วน

ใบงานที่ 12 การเชื่อมต่อ PIC16F876 กับสวิตช์เมตริกซ์ขนาด 4x4

ใบงานที่ 13 การเชื่อมต่อ PIC16F876 กับมอเตอร์ไฟฟ้ากระแสตรง (DC Motor)

ใบงานที่ 14 การเชื่อมต่อ PIC16F876 กับวงจรมอดูเลชันทางความกว้างของพัลส์ (PWM)

ใบงานที่ 15 การเชื่อมต่อ PIC16F876 กับสเตปปิ้งมอเตอร์ (Stepping Motor)

ใบงานที่ 16 การใช้งาน PIC16F876 แปลงสัญญาณแอนะลอกเป็นดิจิตอล

ใบงานที่ 17 การใช้งาน PIC16F876 กับวงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอก

ใบงานที่ 18 การใช้งาน PIC16F876 เชื่อมต่อกับโซลิดสเตทรีเลย์ (Solid State Relay : SSR)

ใบงานที่ 19 การสื่อสารข้อมูลกับคอมพิวเตอร์ผ่านพอร์ตอนุกรม RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ปัญหาและแนวทางแก้ไข

5.2.1 ปัญหาทางด้านฮาร์ดแวร์

ปัญหาที่ 1 การออกแบบวงจรของแต่ละโมดูลในแผงทดลองวงจร อุปกรณ์บางตัวมีขาที่ไม่ตรงกับช่องเสียบอุปกรณ์ ทำให้ต้องใช้อุปกรณ์ตัวอื่นแทนในการทดลอง เช่น แอลอีดีดอตเมตริกซ์ ซึ่งมีขาเป็นจำนวนมากที่ไม่ตรงกับช่องเสียบอุปกรณ์ จึงได้ใช้ แอลอีดีตัวอื่นมาทดลองแทนแอลอีดีดอตเมตริกซ์ที่จะใช้จริง ซึ่งผลในการทดลองนั้นใช้ได้ แต่เมื่อสร้างแผ่นวงจรพิมพ์แล้วลงอุปกรณ์ตัวจริงบนแผ่นวงจรพิมพ์แล้ว ปรากฏว่าแอลอีดีดอตเมตริกที่ใช้จริงนั้น สว่างไม่พอเนื่องมาจากคุณสมบัติของแอลอีดีที่ใช้ทดลองกับแอลอีดีดอตเมตริกซ์ที่ใช้จริงนั้นไม่เหมือนกัน จึงต้องเปลี่ยนโครงสร้างของวงจรใหม่ ซึ่งเป็นการเสียเวลาเป็นอย่างมากในการออกแบบแผ่นวงจรพิมพ์ใหม่

แนวทางแก้ไข การทดลองวงจร ควรทดลองด้วยอุปกรณ์ที่จะใช้จริงๆ ทั้งหมด เพื่อให้ผลการทดลองมีความสมบูรณ์มากที่สุด ก่อนจะนำไปออกแบบแผ่นวงจรพิมพ์ เพื่อไม่ให้เกิดความผิดพลาด เสียเวลาไปโดยเปล่าประโยชน์

ปัญหาที่ 2 การโปรแกรมข้อมูลลงไมโครคอนโทรลเลอร์ ของโมดูลหลัก PIC-ICSP ถ้าพอร์ต B บิต 6 และบิต 7 ถูกต่ออยู่กับอุปกรณ์ที่มีสถานะต่ำ เช่น อุปกรณ์อินพุต หรืออุปกรณ์ที่มีอิพิแดนซ์ต่ำ จะทำให้ไม่สามารถโปรแกรมข้อมูลลงไปได้ เนื่องจาก 2 บิตดังกล่าว ใช้ในการส่งสัญญาณข้อมูลสำหรับโปรแกรม แต่เกิดการสูญเสียไปกับโหลดที่ต่ออยู่

แนวทางแก้ไข ถอดอุปกรณ์ที่ต่อพ่วงอยู่นั้นออกก่อนการโปรแกรม เมื่อโปรแกรมเสร็จจึงเสียบกลับเพื่อทดลองตามเดิม

ปัญหาที่ 3 การออกแบบลายทองแดงของแผ่นวงจรพิมพ์บริเวณจุดบัดกรีเล็กเกินไป ซึ่งทำให้แผ่นวงจรพิมพ์สึกหรอง่าย ลายทองแดงขาดเร็ว แผ่นวงจรพิมพ์ไม่ทนทาน

แนวทางแก้ไข การออกแบบควรใช้ขนาดของลายทองแดงให้ใหญ่ขึ้นกว่าเดิม โดยเฉพาะจุดบัดกรี แต่ก็ไม่ควรเดินให้ติดกันมากจนเกินไปเพราะอาจทำให้เกิดการลัดวงจรได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.2 ปัญหาทางด้านซอฟต์แวร์

ปัญหาที่ 1 คำสั่งของไมโครคอนโทรลเลอร์ PIC นั้นมีอยู่น้อย จำเป็นจะต้องใช้คำสั่งหลายๆ คำสั่งมาประกอบกันขึ้นเป็นกลุ่มคำสั่งใหม่ที่ประมวลผลบางอย่าง เช่น ถ้าหากจะทำการหารตัวเลขจะต้องรวมคำสั่งหลายๆ คำสั่งเข้าด้วยกัน ซึ่งเป็นเรื่องที่ค่อนข้างยาก และทำให้เกิดจุดผิดพลาดในโปรแกรมได้ง่าย

แนวทางแก้ไข การเขียนโปรแกรม ควรจะเขียนให้อยู่ในรูปแบบเป็นโครงสร้างให้มากที่สุด เน้นให้หาจุดผิดพลาดเพื่อแก้ไขโปรแกรมได้ง่าย เช่น แยกโปรแกรมแต่ละส่วนที่ทำหน้าที่ต่างกันออกจากกัน เป็นโปรแกรมย่อย และเพิ่มการเขียนคำอธิบายในแต่ละส่วนให้มากขึ้นเพื่อการทำความเข้าใจโปรแกรมได้ง่ายในภายหลัง

ปัญหาที่ 2 การดาวน์โหลดข้อมูลลงบนไมโครคอนโทรลเลอร์ใช้เวลานาน ยิ่งถ้าหากเป็นเบอร์ที่มีขนาดของหน่วยความจำโปรแกรมใหญ่ก็จะยิ่งนานขึ้น เพราะโปรแกรมที่ใช้รับส่งข้อมูลยังทำงานไม่เร็วพอ สาเหตุหนึ่งมาจากการใช้พอร์ตขนาน แต่ส่งข้อมูลในรูปอนุกรม ทำให้ข้อมูลผิดพลาดได้ง่ายถ้าส่งด้วยความเร็ว

แนวทางแก้ไข เขียนโปรแกรมรับส่งข้อมูลขึ้นใหม่ ด้วยภาษาคอมพิวเตอร์ที่ทำงานได้เร็วขึ้นเช่น ภาษาซี หรือภาษาแอสเซมบลี จะช่วยให้สามารถใช้งานพอร์ตได้อย่างมีประสิทธิภาพมากที่สุด

5.3 ข้อเสนอแนะและแนวทางในการพัฒนา

1. การเพิ่มโมดูลสำหรับการทดลอง ให้มากขึ้นเพื่อสามารถทดลองใช้เชื่อมต่อกับอุปกรณ์ได้อย่างหลากหลาย และจะได้เข้าใจถึงวิธีการใช้อุปกรณ์เหล่านั้นด้วย

2. สร้างโมดูลสำหรับต่อขยายพอร์ตเพิ่ม เพื่อให้สามารถต่อใช้งานโมดูลอื่นๆ ได้พร้อมกันหลายๆ ตัว สามารถประกอบกันเป็นงานที่ใหญ่ขึ้นได้

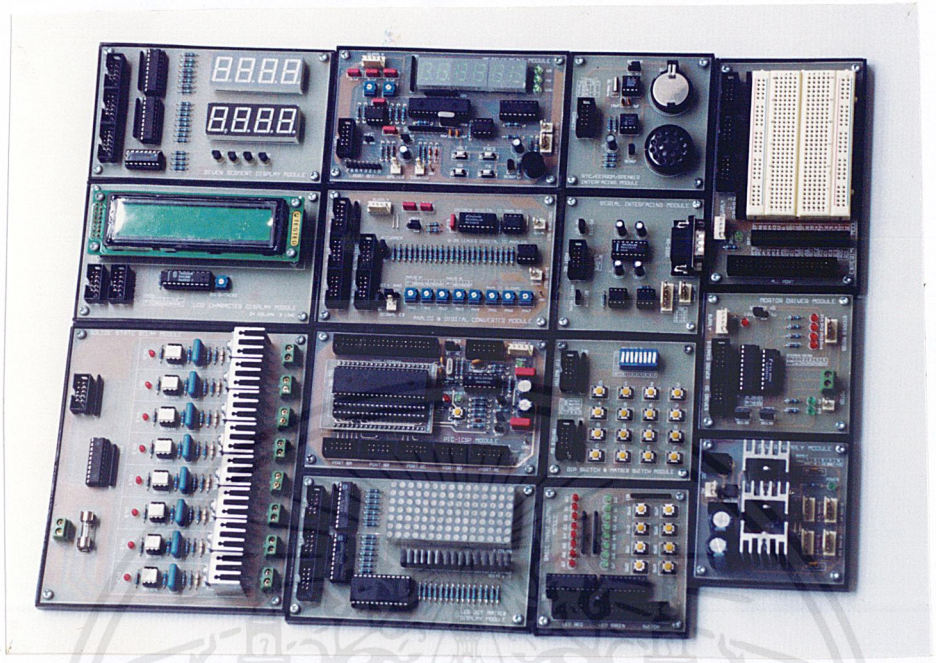
3. ศึกษาภาษาที่ใช้เขียนโปรแกรมสั่งไมโครคอนโทรลเลอร์ เช่น ภาษาซี หรือ ภาษาเบสิก ซึ่งจะทำให้สามารถเขียนสั่งงานไมโครคอนโทรลเลอร์ได้ง่ายขึ้น

4. รวมภาคจ่ายไฟเข้าเป็นกล่องเดียว ไม่จำเป็นต้องแยกออกมาเป็นหลายส่วน เช่นส่วนของหม้อแปลง ส่วนของวงจรควบคุมระดับแรงดัน เพราะเพิ่มความสะดวกในการใช้งาน และสายเชื่อมต่อไม่รุงรังเกินไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก
เครื่องต้นแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.1 เครื่องต้นแบบโมดูลทั้งหมด



รูปที่ ก.2 เครื่องต้นแบบโมดูลที่บรรจุในกระเป๋า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

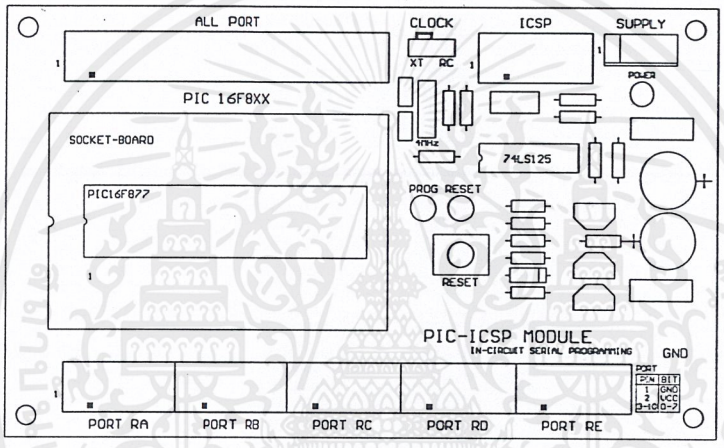
ภาคผนวก ข
คู่มือการใช้งานโมดูลและโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.1 คู่มือการใช้งานชุดปฏิบัติการทดลองไมโครคอนโทรลเลอร์ PIC16F8XX

ชุดปฏิบัติการทดลองไมโครคอนโทรลเลอร์ PIC16F8XX ได้ถูกออกแบบโดยเน้นให้เกิดความสะดวกในการทดลองใช้ไมโครคอนโทรลเลอร์ PIC16F8XX เชื่อมต่อกับอุปกรณ์เพื่อควบคุมการทำงาน โดยได้ออกแบบเป็นโมดูลสำหรับทดลองเรื่องต่างๆ เมื่อต้องการทำการทดลองเรื่องใดก็นำโมดูลนั้นมาเสียบสายเชื่อมต่อกับพอร์ตในโมดูลหลัก ซึ่งเป็นโมดูลของไมโครคอนโทรลเลอร์แล้วทำการโปรแกรมการทำงานลงบนไมโครคอนโทรลเลอร์ เพื่อการทดลองหรือพัฒนาโปรแกรม

1.1.1 โมดูลหลัก PIC-ICSP (PIC- In-Circuit Serial Programming Module)



รูปที่ ข.1 จุดเชื่อมต่อใช้งานของโมดูลหลัก PIC-ICSP

ALL PORT ใช้ต่อกับโมดูลทดลองวงจร ขาใช้งานของ PIC ทั้งหมดจะถูกเชื่อมต่อจากพอร์ตนี้

SOCKET-BOARD

เป็นซ็อกเก็ตแบบ ZIP สำหรับเสียบตัวไมโครคอนโทรลเลอร์ ทั้งแบบ 40, 28 และ 18 ขา โดยมีซ็อกเก็ตสำหรับเสียบ ZIP อีกทีให้ตรงกับจำนวนขาที่ต้องการใช้งาน

ICSP สำหรับเชื่อมต่อสายดาวน์โหลดข้อมูลไปยังคอมพิวเตอร์เพื่อการโปรแกรม

SUPPLY สำหรับต่อกับภาคจ่ายไฟ

PORTRA – PORTRE

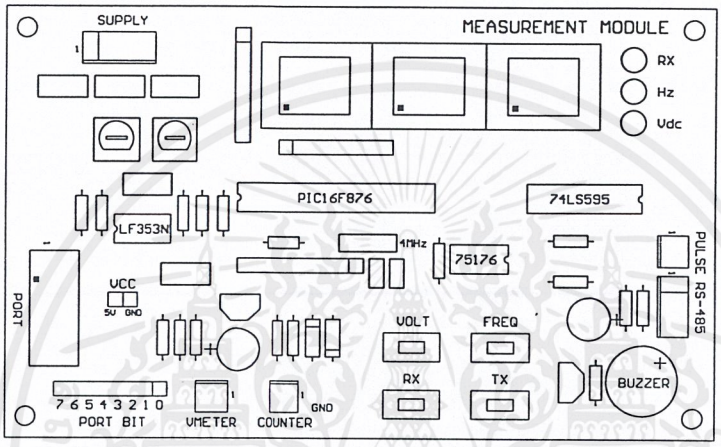
เป็นพอร์ตสำหรับต่อใช้งาน เชื่อมต่อกับโมดูลอื่นๆ

CLOCK จั๊มเปอร์เลือกสัญญาณนาฬิกาของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.1.2 โมดูลเครื่องมือวัด (Measurement Module)

กดสวิตช์เลือกโหมดที่จะวัด สวิตช์ VOLT คือ โวลต์มิเตอร์, FREQ คือ เครื่องวัดความถี่ RX คือ เครื่องรับข้อมูลอนุกรม RS-485, TX คือ เครื่องส่งข้อมูลอนุกรม RS-485 โดยสัญญาณที่จะส่งออกไปคือตัวเลขจากส่วนแสดงผลทั้ง 6 หลัก ส่งไปแบบอนุกรม เริ่มจากหลักที่ 6 ด้านซ้ายสุด LED ด้านซ้ายใช้แสดงโหมดปัจจุบัน



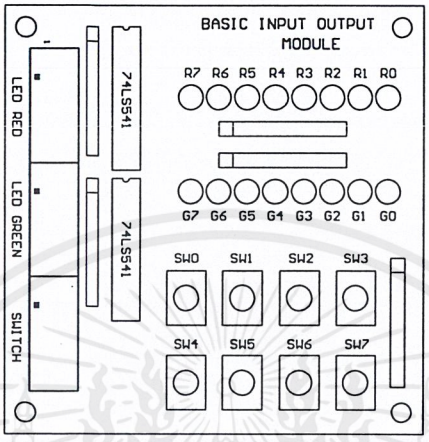
รูปที่ ข.2 จุดเชื่อมต่อใช้งานของโมดูลเครื่องมือวัด

SUPPLY	สำหรับจ่ายแรงดันให้โมดูล
VR (ด้านซ้าย)	สำหรับปรับอัตราขยายวงจรดิฟแอมป์ของวงจรเครื่องวัดแรงดัน
VR (ด้านขวา)	สำหรับปรับแรงดันอ้างอิงศูนย์ของวงจรเครื่องวัดแรงดัน
PORT	ต่อกับสายแพเพื่อการวัดที่สัญญาณในสายแพได้ง่ายขึ้น
PORT BIT	เป็นจุดสำหรับการให้โพรบวัดสัญญาณจาก PORT
VMETER	สำหรับต่อสายโพรบเพื่อวัดแรงดัน
COUNTER	สำหรับต่อสายโพรบเพื่อวัดความถี่
PULSE	สำหรับต่อใช้งานโปรแกรมสร้างสัญญาณพัลส์
RS-485	สำหรับทดสอบรับส่งข้อมูลอนุกรมแบบ RS-485

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.1.3 โมดูลเบสิกอินพุตเอาต์พุตพื้นฐาน (Basic Input Output Module)

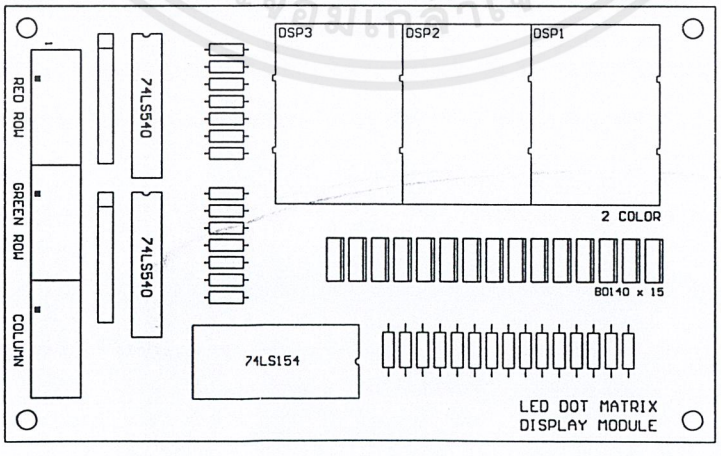
แอลอีดีจะแอคตีฟที่ลอจิก “1” ทั้งสองสี ที่พอร์ตต่อใช้งานจะมีตัวต้านทานต่อพูลดาวน์อยู่ สวิตช์จะแอกตีฟที่ลอจิก “0” เมื่อกดจะอ่านค่าได้ “0” ถ้าไม่กดจะเป็น “1”



รูปที่ ข.3 จุดเชื่อมต่อใช้งานของโมดูลเบสิกอินพุตเอาต์พุตพื้นฐาน

LED RED	พอร์ตใช้งานชุดแสดงผลแอลอีดีสีแดงขนาด 8 บิต
LED GREEN	พอร์ตใช้งานชุดแสดงผลแอลอีดีสีเขียวขนาด 8 บิต
SWITCH	พอร์ตใช้งานสวิตช์สร้างสภาวะลอจิกขนาด 8 บิต

1.1.4 โมดูลแสดงผลแอลอีดีดอตเมตริกซ์ (LED Dot Matrix Display Module



รูปที่ ข.4 จุดเชื่อมต่อใช้งานของโมดูลแสดงผลแอลอีดีดอตเมตริกซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะต้องส่งข้อมูลสำหรับให้แอลอีดีติดด้านแถว โดยแบ่งเป็นแถวสำหรับสีแดง และสีเขียว ใช้สัญญาณแถวละ 8 บิต รวม 2 แถว 16 บิต ถ้าต้องการให้ติดเป็นสีเหลืองก็ให้แถวที่ตรงกันของทั้งสองสีติดพร้อมกัน สภาวะที่จะทำให้หลอดติดคือ "1" ส่วนด้านหลัก ให้สัญญาณ 4 บิตในการเลือกให้หลักใดหลักหนึ่งติด จากหลัก 00H ถึง 0EH ถ้าจะไม่ให้มีแถวใดติดเลยสามารถส่งข้อมูล 0FH เข้ามาที่พอร์ตนี้ได้

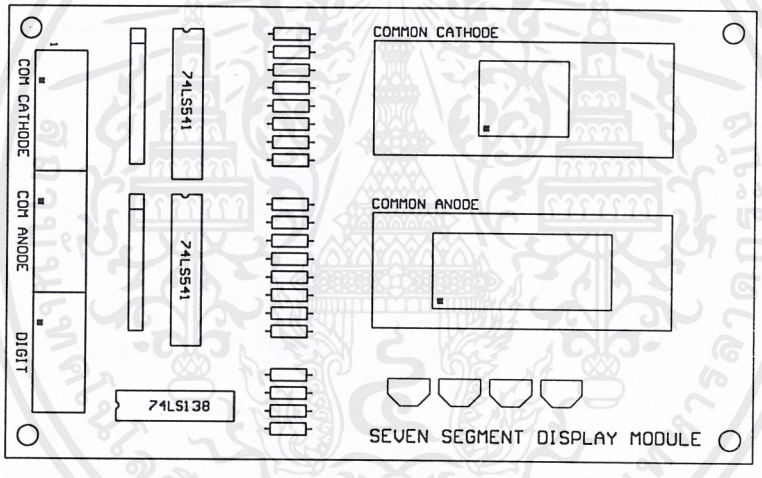
RED ROW พอร์ตส่งข้อมูลทางด้านแถวให้แอลอีดีสีแดง

GREEN ROW พอร์ตส่งข้อมูลทางด้านแถวให้แอลอีดีสีเขียว

COLUMN พอร์ตส่งข้อมูลที่ใช้เลือกคอลัมน์ ใช้สัญญาณ 4 บิตล่างของพอร์ต

1.1.5 โมดูลแสดงผลตัวเลข 7 ส่วน (Seven Segment Display Module)

พอร์ตเลือกหลักจะใช้สัญญาณ 3 บิตล่าง สำหรับเลือกให้ติดหลักใดหลักหนึ่งใน 8 หลัก



รูปที่ ข.5 จุดเชื่อมต่อใช้งานของโมดูลแสดงผลตัวเลข 7 ส่วน

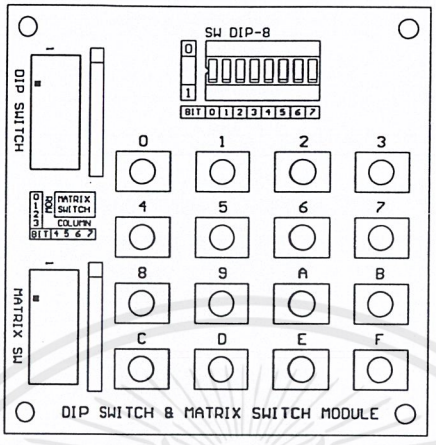
COM CATHODE พอร์ตส่งข้อมูลให้เซ็กเมนต์ของคอมมอนคาโถด

COM ANODE พอร์ตส่งข้อมูลให้เซ็กเมนต์ของคอมมอนอาโนด

DIGIT พอร์ตส่งข้อมูลเลือกหลัก ใช้สัญญาณ 3 บิตล่างของพอร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.1.6 โมดูลดิพสวิตช์และเมตริกซ์สวิตช์ (Dip Switch and Matrix Switch Module

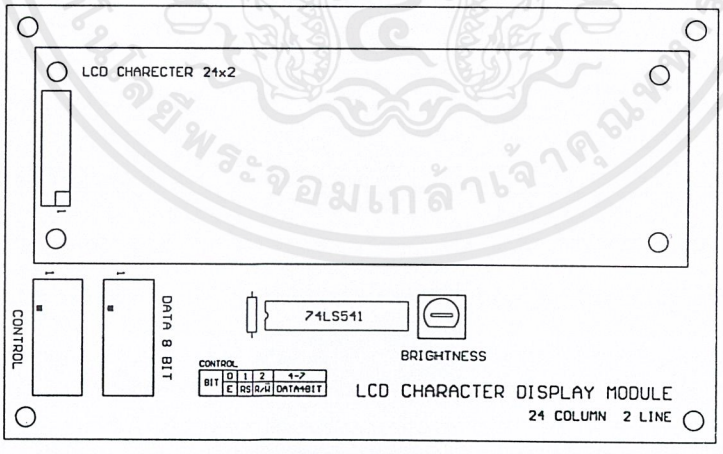


รูปที่ ข.6 จุดเชื่อมต่อใช้งานของโมดูลดิพสวิตช์และเมตริกซ์สวิตช์

DIP SWITCH พอร์ตสำหรับอ่านสภาวะของดิพสวิตช์

MATRIX พอร์ตใช้งานเมตริกซ์สวิตช์ ทุกบิตจะมีตัวต้านทานต่อพูลอัพอยู่

1.1.7 โมดูลแสดงผลแบบผลึกเหลว (LCD Character Display Module)



รูปที่ ข.7 จุดเชื่อมต่อใช้งานของโมดูลแสดงผลแบบผลึกเหลว

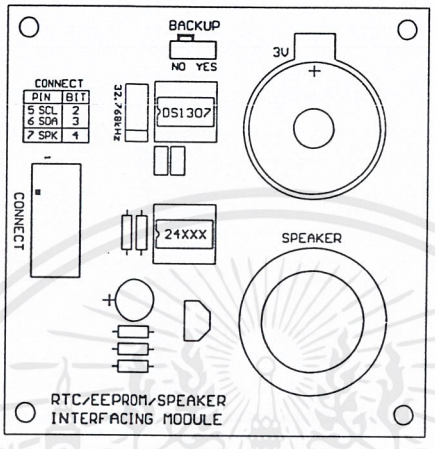
CONTROL พอร์ตสัญญาณควบคุม LCD ใช้สัญญาณ 3 จากบิตล่าง

DATA พอร์ตรับส่งสัญญาณข้อมูล 8 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.1.8 โมดูลเชื่อมต่ออุปกรณ์ RTC / EEPROM / SPEAKER (RTC / EEPROM / SPEAKER INTERFACING Module)

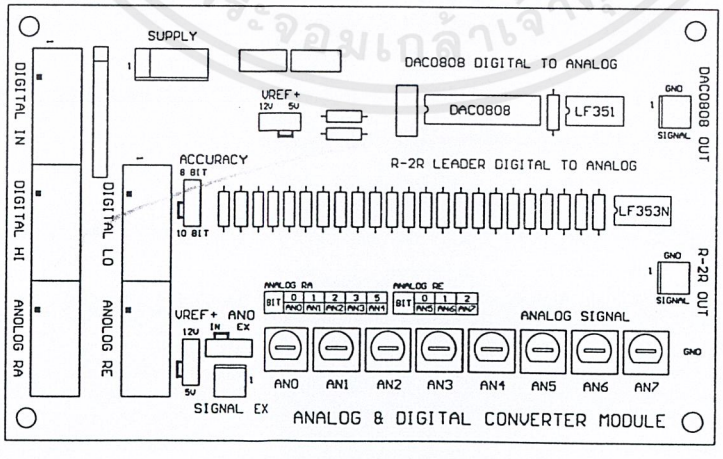
RTC และ EEPROM จะใช้สัญญาณ 2 บิตจากพอร์ต 1 พอร์ต รับส่งข้อมูลแบบ I^2C



รูปที่ ข.8 จุดเชื่อมต่อใช้งานของโมดูลเชื่อมต่ออุปกรณ์ RTC / EEPROM / SPEAKER

CONTROL พอร์ตเชื่อมต่อ ใช้สัญญาณบิต 2 และบิต 3 เป็นบัสสำหรับ I^2C ส่วนลำโพงใช้สัญญาณบิต 4

1.1.9 โมดูลแปลงสัญญาณแอนะลอกและดิจิตอล (Analog and Digital Converter Module)



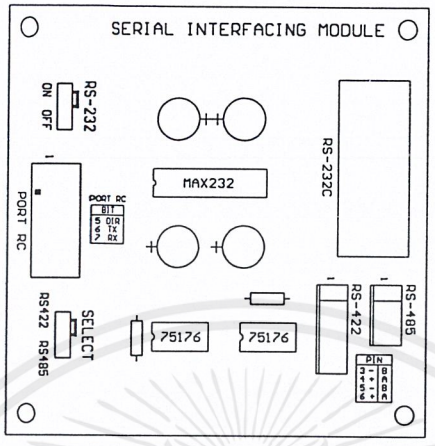
รูปที่ ข.9 จุดเชื่อมต่อใช้งานของโมดูลแปลงสัญญาณแอนะลอกและดิจิตอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUPPLY	สำหรับป้อนแรงดันจ่ายให้วงจร
DIGITAL IN	พอร์ตต่อสัญญาณดิจิตอลอินพุตขนาด 8 บิต ของวงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอกแบบใช้ไอซี DAC0808
DAC0808 OUT	สัญญาณเอาต์พุตแอนะลอกของ DAC0808
DIGITAL HI	พอร์ตต่อสัญญาณดิจิตอลอินพุต 8 บิตบน ของวงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอกแบบ R-2R Ladder
DIGITAL LO	พอร์ตต่อสัญญาณดิจิตอลอินพุต 2 บิตล่างของวงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอกแบบ R-2R Ladder จะต่ออยู่ที่ บิต 1 และบิต 0 ของพอร์ต
R-2R OUT	สัญญาณเอาต์พุตแอนะลอกของ R-2R Ladder
ANALOG RA	พอร์ตแอนะลอกอินพุตสำหรับป้อนให้ ADC ของไมโครคอนโทรลเลอร์ ที่พอร์ต A คือสัญญาณ AN0-AN4
ANALOG RE	พอร์ตแอนะลอกอินพุตสำหรับป้อนให้ ADC ของไมโครคอนโทรลเลอร์ ที่พอร์ต E คือสัญญาณ AN5-AN7
SIGNAL EX	สำหรับป้อนสัญญาณแอนะลอกจากภายนอก
VREF+	จั๊มเปอร์สำหรับเลือกแรงดันอ้างอิงให้แก่วงจรแปลงสัญญาณ
ACCURACY	จั๊มเปอร์สำหรับเลือกขนาดบิตของวงจร R-2R Ladder
AN0	จั๊มเปอร์เลือกสัญญาณจากภายนอกเข้าที่พอร์ต AN0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.1.10 โมดูลสื่อสารข้อมูลอนุกรม (Serial Interfacing Module)



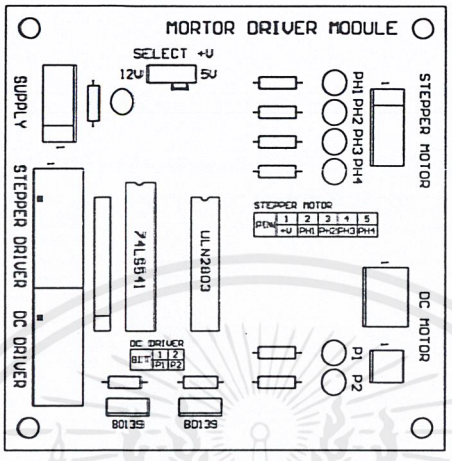
รูปที่ ข.10 จุดเชื่อมต่อใช้งานของโมดูลสื่อสารข้อมูลอนุกรม

เมื่อเซตจั๊มเปอร์ RS-232 ไปที่ ON จะเป็นการใช้มาตรฐานนี้ในการสื่อสารข้อมูล เมื่อต้องการใช้มาตรฐาน RS-485 หรือ RS-422 ให้เซตจั๊มเปอร์ RS-232 ไปที่ OFF และจึงเซตจั๊มเปอร์ SELECT เลือกมาตรฐานที่ต้องการ

PORT RC	สำหรับต่อเข้ากับพอร์ต C บิต 7(RX), บิต 6(TX) และบิต 5(DIR) ซึ่งเป็นจุดต่อสำหรับ USART ของไมโครคอนโทรเลอร์
RS-232C	คอนเนคเตอร์ DP-9 ตัวเมีย สำหรับเชื่อมต่อกับพอร์ตอนุกรมของคอมพิวเตอร์
RS-485	สำหรับส่งสัญญาณอนุกรมแบบ RS-485
RS-422	สำหรับส่งสัญญาณอนุกรมแบบ RS-422
RS-232	จั๊มเปอร์เพื่อเปิดปิดวงจรสื่อสาร RS-232
SELECT	จั๊มเปอร์เพื่อเลือกการใช้มาตรฐาน RS-485 หรือ RS-422

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.1.11 โมดูลขับดีซีมอเตอร์และสเตปปิ้งมอเตอร์ (DC Motor and STEPPING Motor Driver Module)

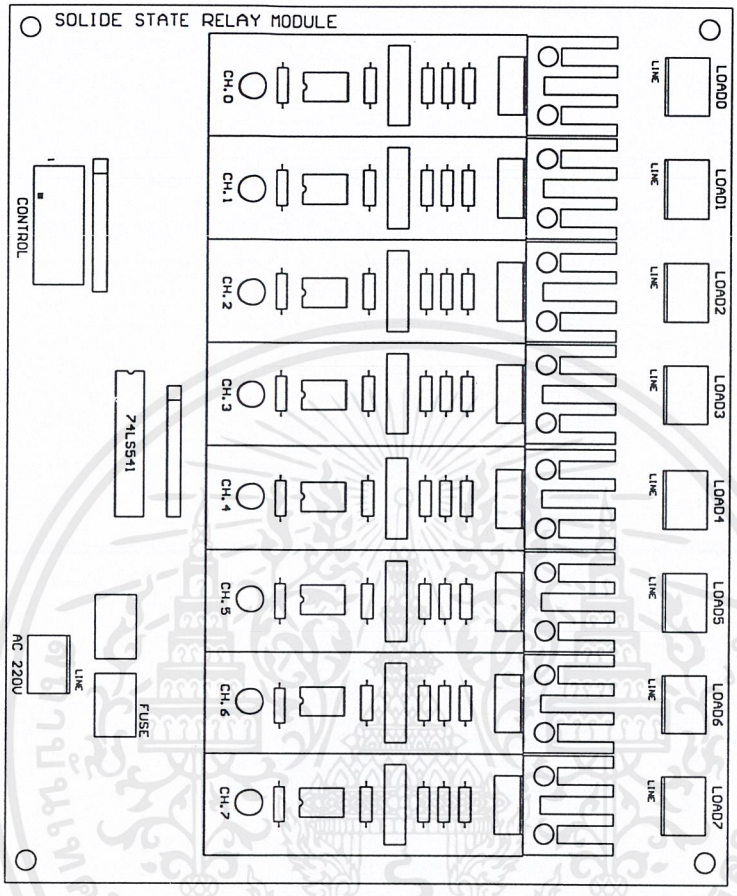


รูปที่ ข.11 จุดเชื่อมต่อใช้งานของโมดูลขับดีซีมอเตอร์และสเตปปิ้งมอเตอร์

SUPPLY	สำหรับป้อนแรงดันให้แก่โมดูลเมื่อต้องการใช้แรงดันสูงในการขับมอเตอร์
STEPPER DRIVER	พอร์ตสำหรับควบคุมการขับสเตปปิ้งมอเตอร์
STEPPER MOTOR	สำหรับต่อสเตปปิ้งมอเตอร์ชนิดยูนิโพลาร์
DC DRIVER	พอร์ตสำหรับควบคุมการขับดีซีมอเตอร์ ใช้สัญญาณบิต 1 และบิต 2
DC MOTOR	สำหรับต่อดีซีมอเตอร์ สามารถต่อได้ทั้งจากคอนเนคเตอร์สีเขียวและสีขาว
SELECT +V	จั๊มเปอร์เลือกแรงดันที่ใช้ขับมอเตอร์ในกรณีต้องการใช้แรงดันสูงขับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.1.12 โมดูลโซลิดสเตทรีเลย์ (Solid State Relay Module)



รูปที่ ข.12 จุดเชื่อมต่อใช้งานของโมดูลโซลิดสเตทรีเลย์

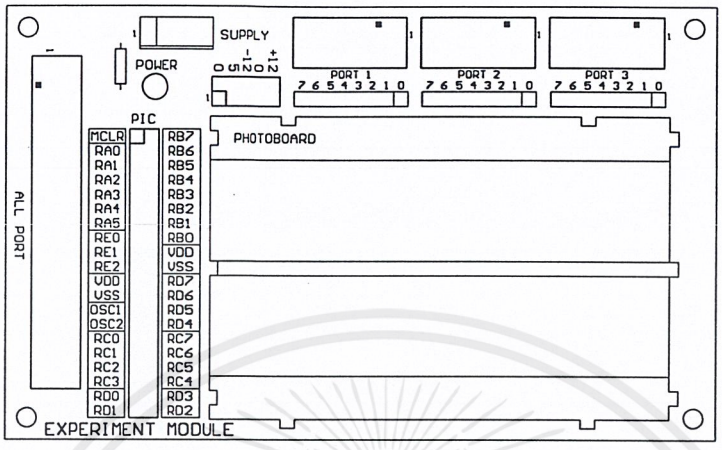
CONTROL พอร์ตสำหรับควบคุมโซลิดสเตทรีเลย์ ขนาด 8 บิต

AC 220V สำหรับต่อกับแรงดัน AC 220V เป็นแหล่งจ่ายให้แก่โหลด

LOAD0 – LOAD7 สำหรับต่อโหลดที่ต้องการควบคุม เส้นที่เขียนว่า LINE หมายถึง สายเส้นนั้นจะมีแรงดันไฟเลี้ยงอยู่ตลอดเวลา ส่วนเส้นที่ไม่ได้เขียนว่า LINE จะเป็นสายที่ผ่านโซลิดสเตทรีเลย์ จะไม่มีแรงดันเมื่อโซลิดสเตทรีเลย์อยู่ในสภาวะ OFF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.1.13 โมดูลทดลองวงจร (Circuit Experimental Module)



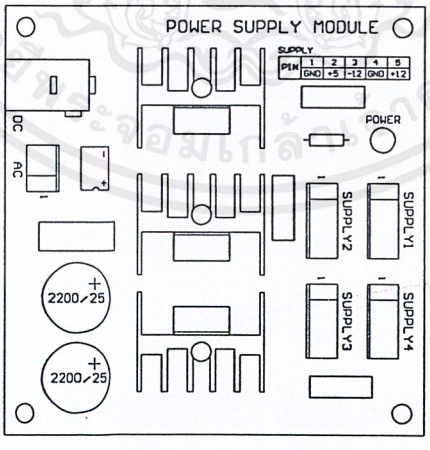
รูปที่ ข.13 จุดเชื่อมต่อใช้งานของโมดูลทดลองวงจร

ALL PORT พอร์ตเชื่อมต่อกับโมดูลหลัก รวมสัญญาณจาก PIC ทุกๆ ขาเพื่อการทดลอง

SUPPLY สำหรับต่อแหล่งจ่ายไฟเลี้ยง สำหรับการทดลอง

PORT 1 – PORT 3 คอนเนคเตอร์สำหรับเชื่อมต่อเพื่อทดลองกับโมดูลอื่นๆ

1.1.14 โมดูลภาคจ่ายไฟ (Power Supply Module)



รูปที่ ข.14 จุดเชื่อมต่อใช้งานของโมดูลภาคจ่ายไฟ

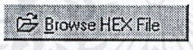
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคจ่ายไฟสามารถเลือกใช้ไฟจากหม้อแปลงที่มีเซ็นเตอร์แทปได้ เพื่อให้สามารถจ่ายไฟได้ครบคือ 5 โวลต์ และ ±12 โวลต์ หรือสามารถใช้กับอแดปเตอร์ขนาด 12V ได้ ถ้าหากไม่มีหม้อแปลง แต่เอาต์พุตจะไม่มีแรงดันไฟ –12 โวลต์ ซึ่งจะทำให้ไม่สามารถใช้งานโมดูลแปลงสัญญาณดิจิตอลและแอนะลอกได้ ส่วนโมดูลอื่นก็ยังใช้งานได้ตามปกติ

DC	สำหรับต่อกับดีซีอแดปเตอร์ขนาด 12 โวลต์ขึ้นไป
AC	สำหรับต่อกับหม้อแปลงขนาด 15-0-15 โวลต์
SUPPLY1 – SUPPLY 4	เป็นคอนเนคเตอร์สำหรับจ่ายไฟให้แก่โมดูลอื่นๆ

1.2 การใช้งานโปรแกรม PIC-ICSP Device

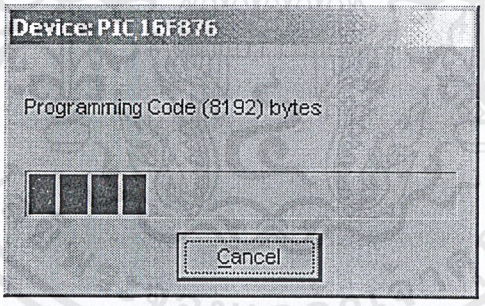
1) เลือกเบอร์ของไมโครคอนโทรลเลอร์ที่ใช้งานจาก 

2) เปิดไฟล์ Hex โดยคลิกปุ่ม 

จะปรากฏหน้าต่างเปิดไฟล์ ให้เลือกไฟล์ที่ต้องการโปรแกรม

3) สามารถโปรแกรมได้ทันทีโดยคลิกปุ่ม 

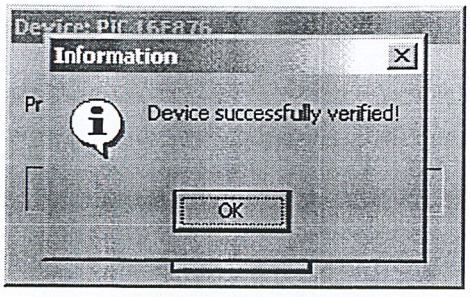
จะเริ่มทำการโปรแกรมโดยจะปรากฏหน้าต่างแสดงการโปรแกรมดังรูปที่ ข.15



รูปที่ ข.15 หน้าต่างแสดงขณะทำการโปรแกรม

ขณะโปรแกรมข้อมูลลงบนไมโครคอนโทรลเลอร์นี้ จะมี LED แสดงสถานะการโปรแกรมบนตัวโมดูล ซึ่งในขณะนี้ห้ามกดปุ่ม RESET เป็นอันขาด เพราะจะทำให้เกิดความผิดพลาดของข้อมูลขึ้น

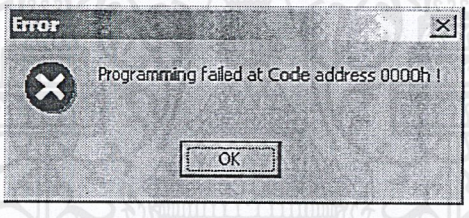
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.16 เมื่อโปรแกรมเสร็จสมบูรณ์

เมื่อโปรแกรมเสร็จ จะแสดงหน้าต่างดังรูปที่ ข.16 ให้คลิกปุ่ม OK จากนั้นทำการรัน CPU โดยการคลิกที่ปุ่ม RUN ไมโครคอนโทรลเลอร์ก็จะเริ่มทำงาน

กรณีที่เกิดความผิดพลาดขึ้นในขณะดาวน์โหลดจะเกิดหน้าต่างดังรูปที่ ข.17



รูปที่ ข.17 เมื่อเกิดความผิดพลาดขณะทำการโปรแกรม

ในกรณีเกิดความผิดพลาดขึ้นซึ่งอาจเกิดจากสาเหตุดังต่อไปนี้

1.1) การเชื่อมต่อสายสัญญาณระหว่างคอมพิวเตอร์และตัวโมดูลไม่สมบูรณ์

1.2) ไม่จ่ายไฟให้ตัวโมดูล

1.3) มีโมดูลที่จ่ายสถานะอินพุตให้กับตัว PIC-ICSP Module ต่ออยู่ที่ Port RB ทำให้ข้อมูลที่ส่งไปโปรแกรมเกิดการผิดพลาด ให้ถอดอุปกรณ์นั้นออก

1.4) ในระหว่างโปรแกรมห้ามกดปุ่ม RESET เป็นอันขาดไม่ว่าจะเป็นบนตัวโมดูล หรือบนโปรแกรม PIC-ICSP Devices เพราะจะเกิดความผิดพลาดขึ้นทันที

4) เมื่อโปรแกรมเสร็จ สามารถที่จะตรวจสอบโค้ดโปรแกรมบนตัวไมโครคอนโทรลเลอร์ได้ว่าข้อมูลตรงกันกับที่โปรแกรมไปหรือไม่โดยคลิกปุ่ม 

ก็จะแสดงหน้าต่างการเปรียบเทียบโค้ด ถ้ามีโค้ดไม่ตรงกัน จะแสดงหน้าต่างบอกความผิดพลาดแสดงขึ้นมา ให้ตรวจเช็ค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความผิดพลาดที่เกิดขึ้นซึ่งอาจจะมาจากการติดต่อระหว่างตัวโมดูลกับคอมพิวเตอร์ จากนั้นทำการโปรแกรมเข้าไปใหม่

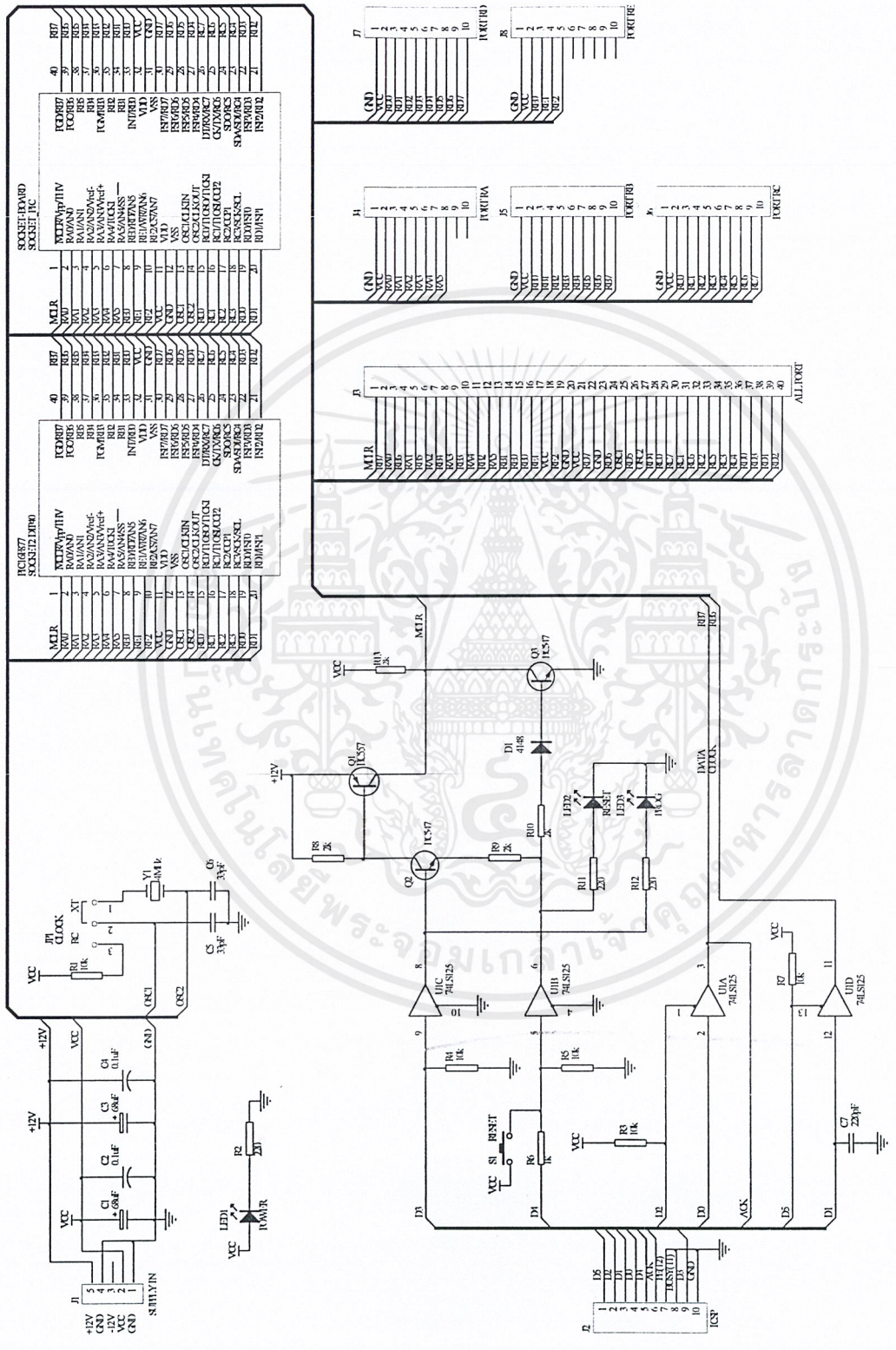
5) โปรแกรม PIC-ICSP Devices สามารถสั่งงานตัวโมดูลได้คือ เมื่อคลิกที่ปุ่ม RESET ใน Control Board ก็จะเป็นการ Reset ตัว CPU ให้เริ่มต้นทำงานใหม่ และเมื่อกดปุ่ม STOP ก็จะเป็นการหยุดการประมวลผลของ CPU โดยจะมีลักษณะเหมือนการกด Reset ค้างไว้โดยเมื่อต้องการให้ CPU ประมวลผลอีกครั้ง ก็คลิกที่ปุ่ม RUN ตัว CPU ก็จะเริ่มการทำงานใหม่อีกครั้ง

6) สามารถเรียกใช้โปรแกรมอื่นๆ ขึ้นมาทำงานได้ คือ โปรแกรม MPLAB ซึ่งเป็นโปรแกรมสำหรับจัดการกับการเขียนโปรแกรมภาษาแอสเซมบลี สั่งตัวไมโครคอนโทรลเลอร์ โปรแกรม IC-Prog เป็นโปรแกรมสำหรับจัดการการดาวน์โหลดไฟล์ Hex ลงบนตัวไมโครคอนโทรลเลอร์ โปรแกรม MPASM เป็นตัว Assembler สร้างไฟล์ .Hex ซึ่งสำหรับการเรียกใช้โปรแกรม MPLAB และ MPASM นั้น จะต้องมีโปรแกรมดังกล่าวติดตั้งอยู่บนเครื่องคอมพิวเตอร์แล้วเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

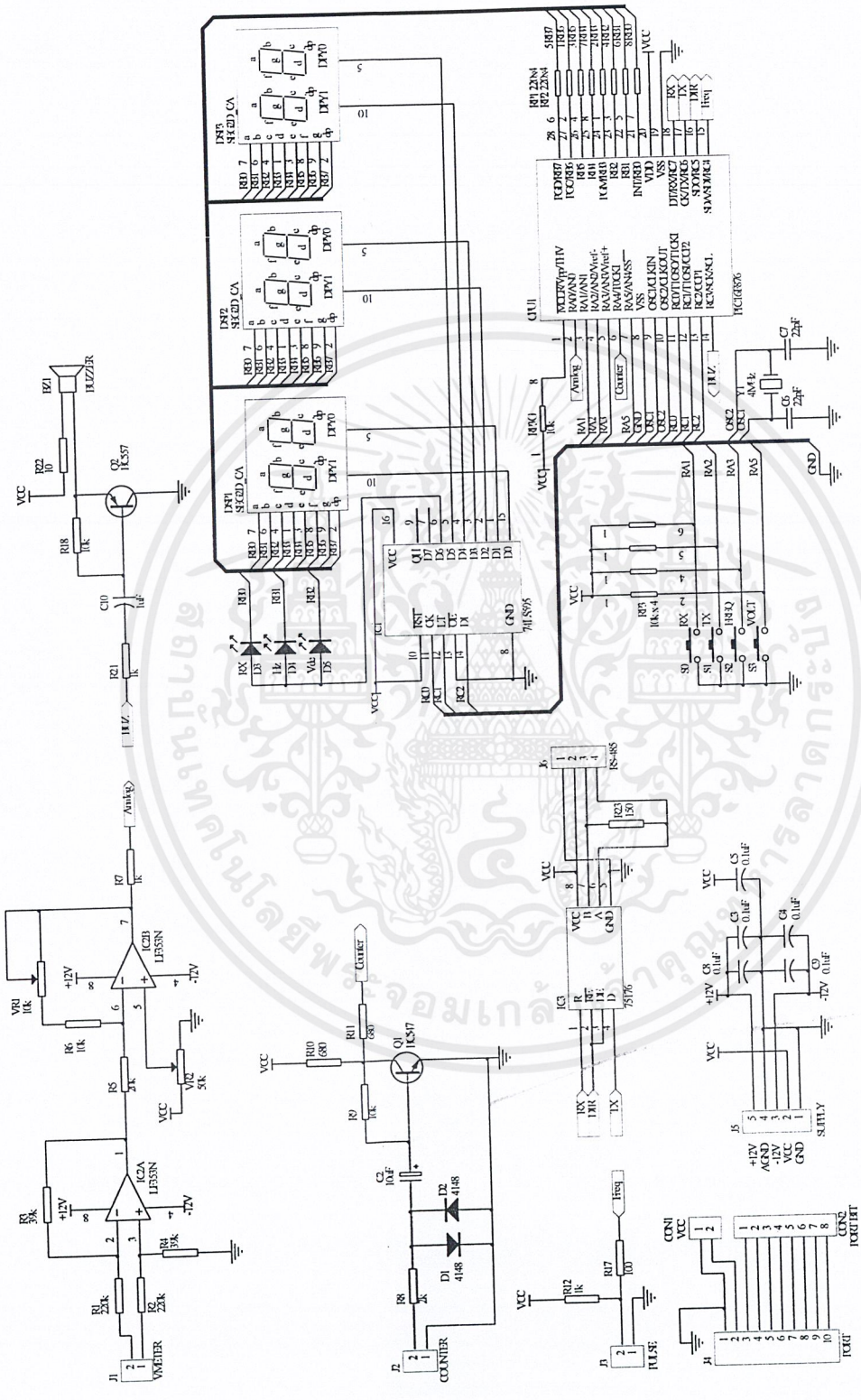
ภาคผนวก ค
วงจร และแผ่นวงจรพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



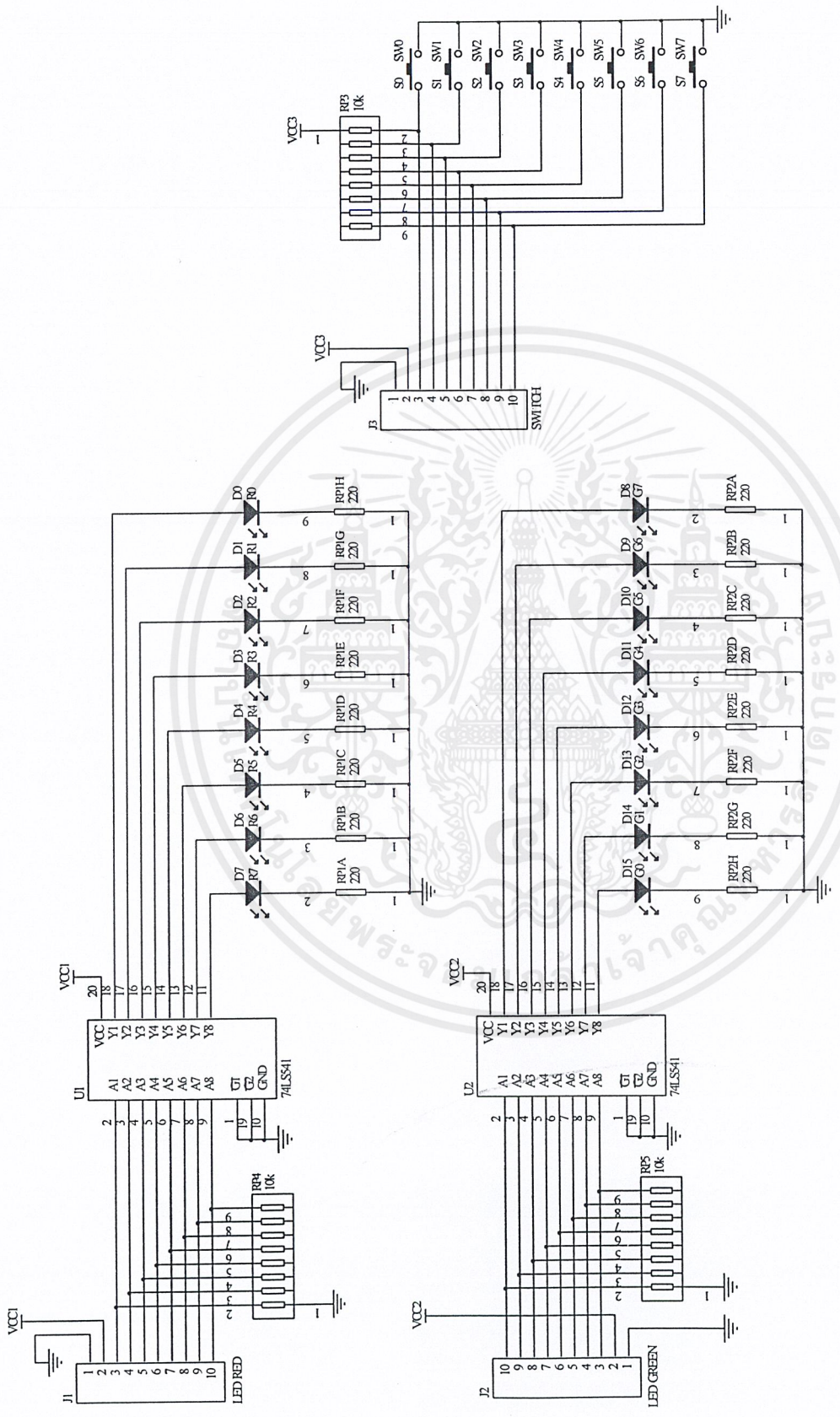
รูปที่ ค.1 วงจรสมบูรณ์ของโมดูลหลัก PIC-ICSP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



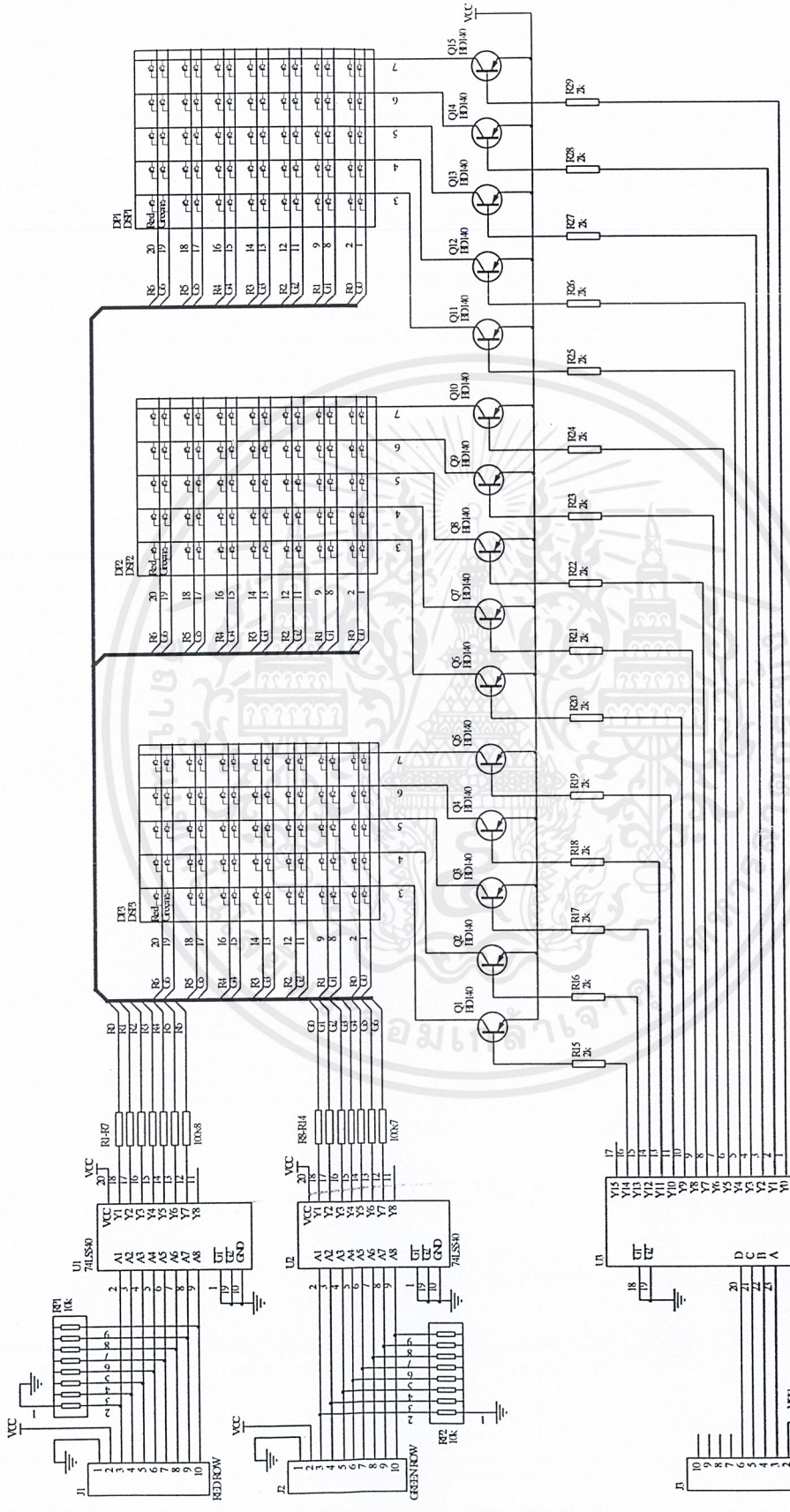
รูปที่ ค.2 วงจรสมบูรณ์ของโมดูลเครื่องมือวัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



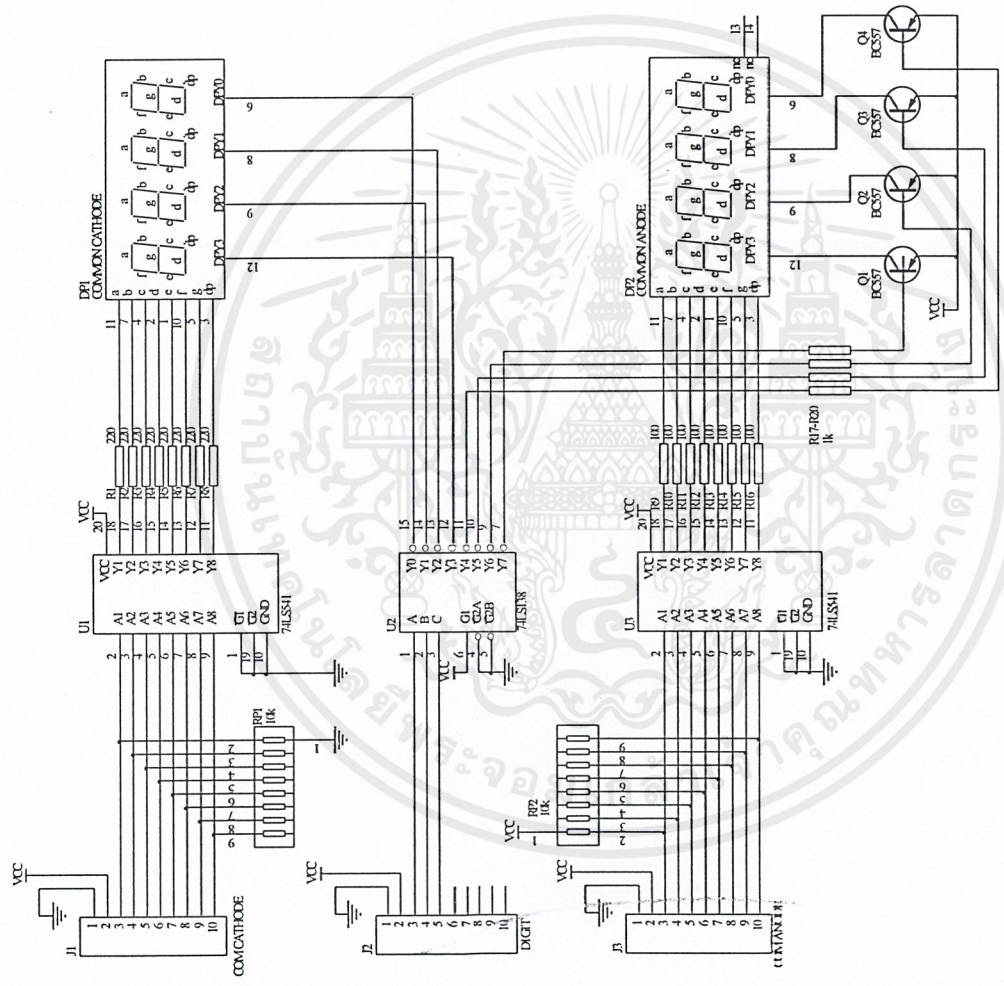
รูปที่ ค.3 วงจรสมบูรณ์ของโมดูลเบสิกอินพุตเอาต์พุตพื้นฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



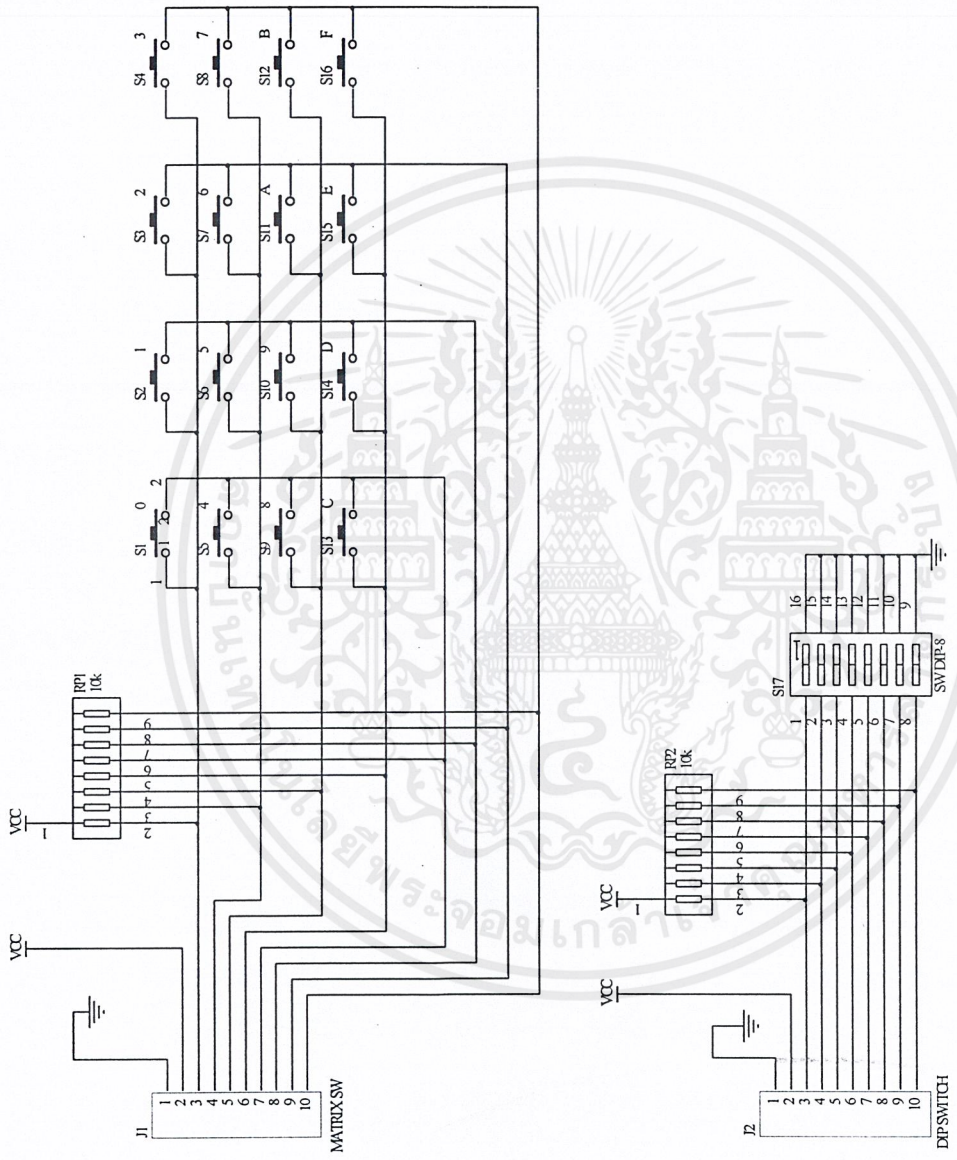
รูปที่ ค.4 วงจรสมบูรณ์ของโปรแกรมแสดงผลแอลอีดีดอตเมตริกซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



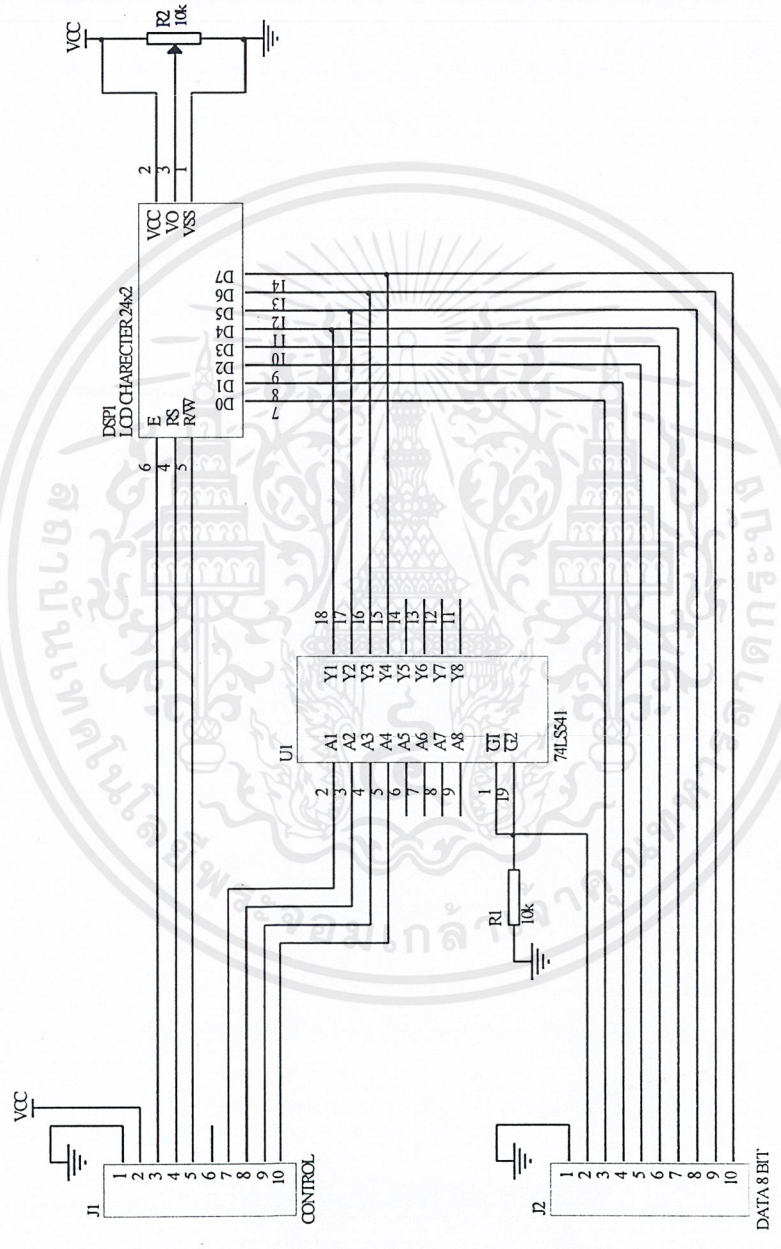
รูปที่ ค.5 วงจรสมบูรณ์ของโมดูลแสดงผลตัวเลข 7 ส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



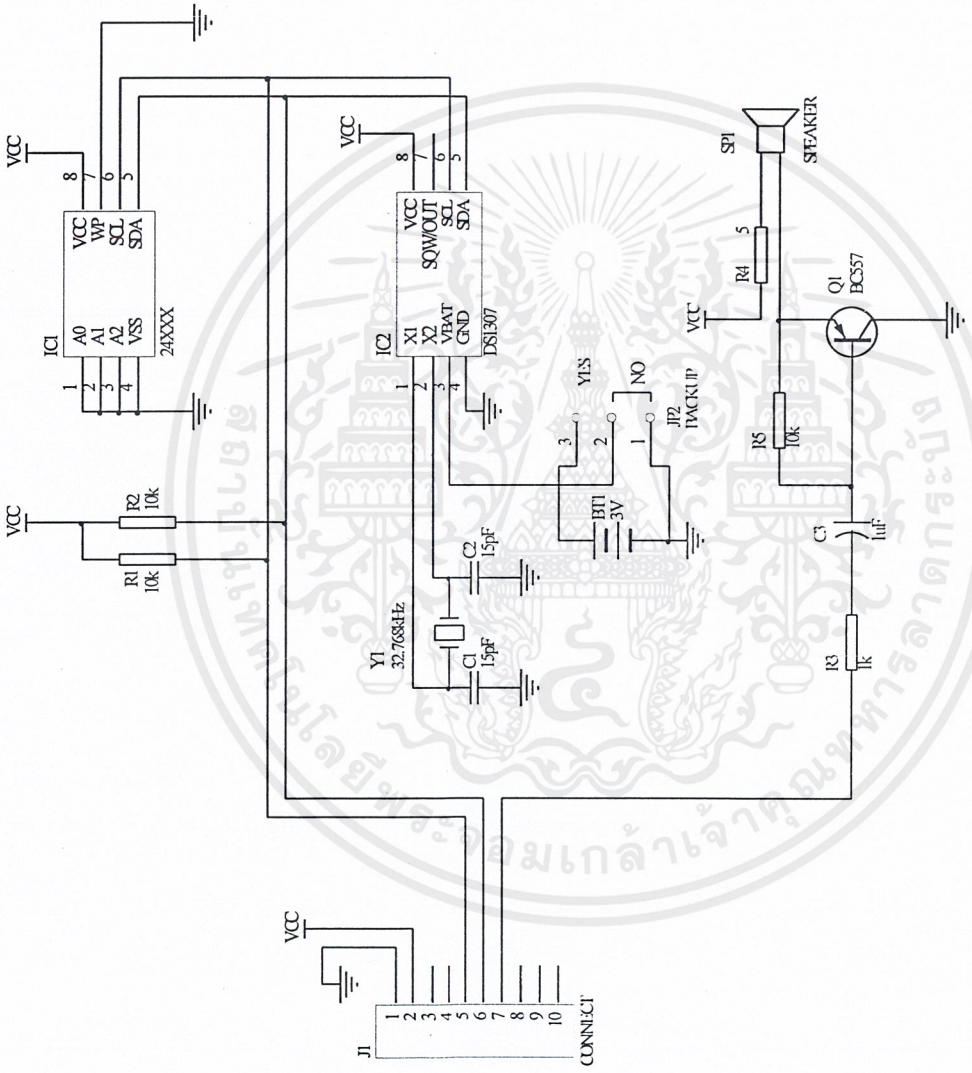
รูปที่ ค.6 วงจรสมบูรณ์ของโมดูลดิพสวิตช์และเมตริกซ์สวิตช์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



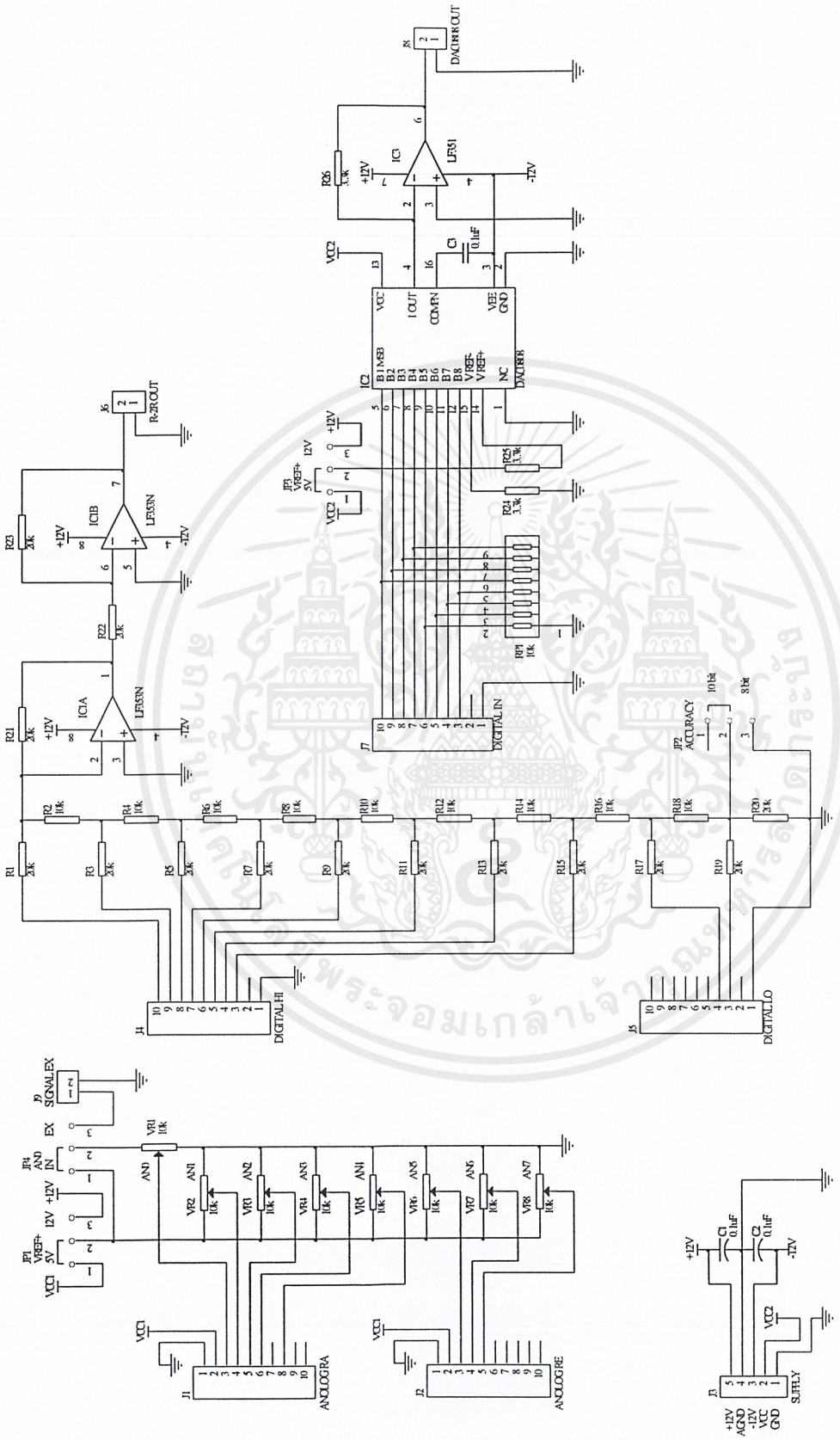
รูปที่ ค.7 วงจรสมบูรณ์ของโมดูลแสดงผลแบบตึกหลอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



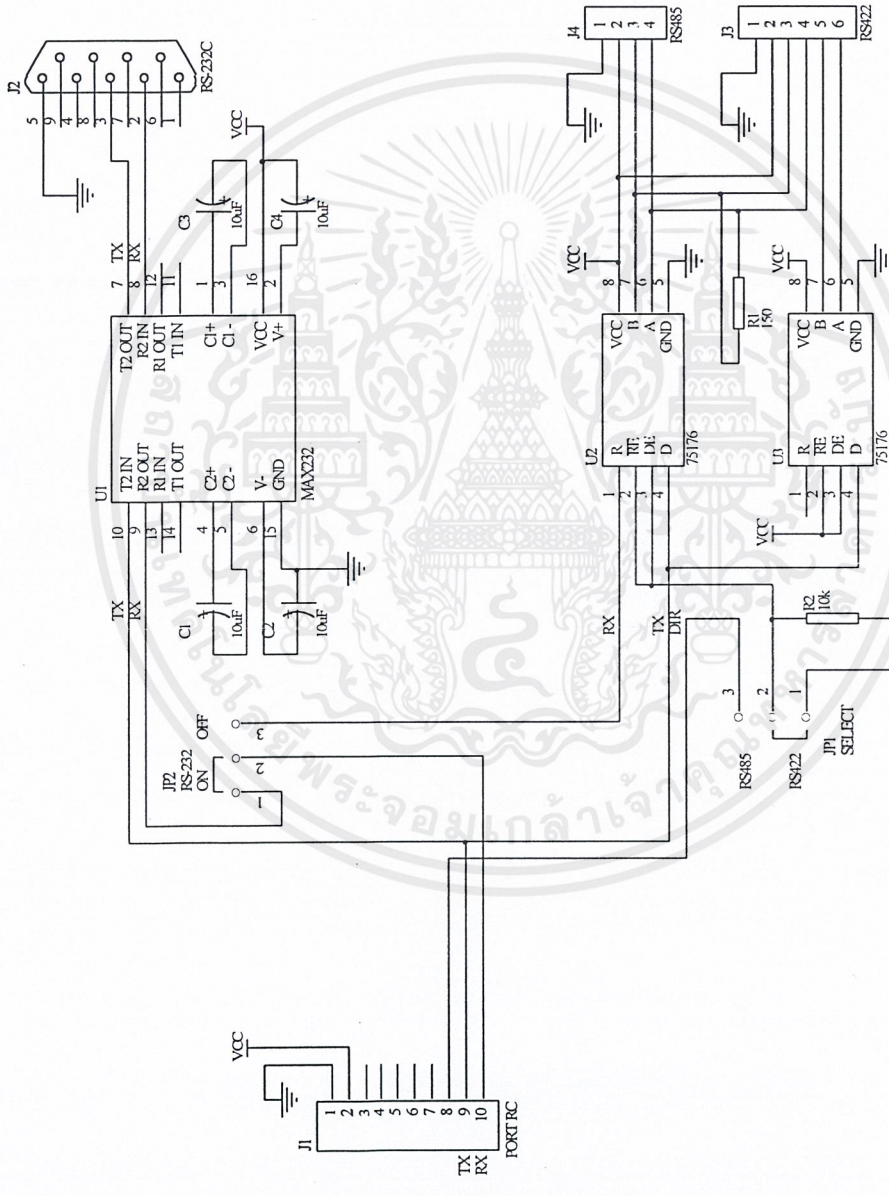
รูปที่ ค.8 วงจรสมบูรณ์ของโมดูลเชื่อมต่ออุปกรณ์ RTC / EEPROM / SPEAKER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



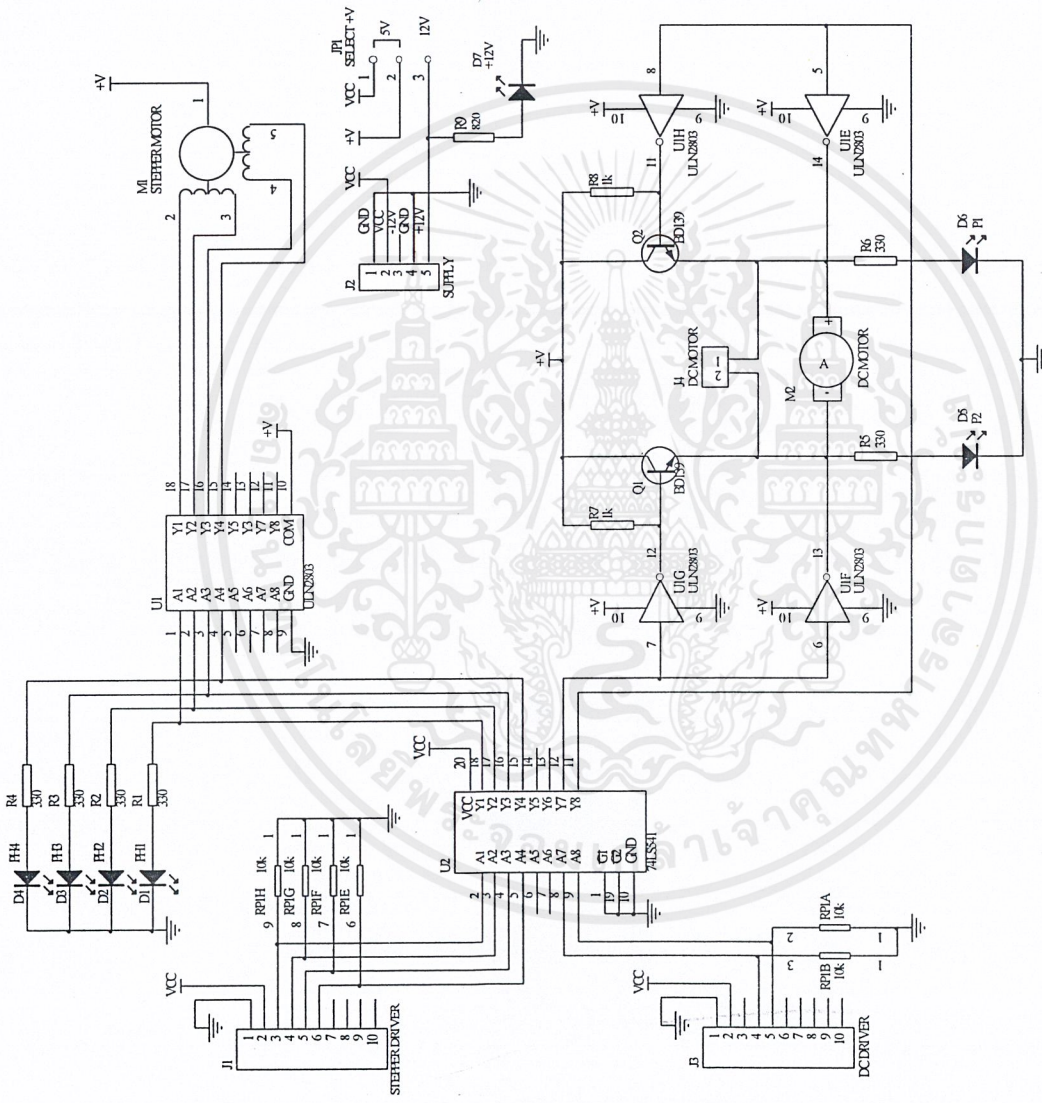
รูปที่ ค.9 วงจรสมบูรณ์ของโมดูลแปลงสัญญาณแอนะลอกและดิจิตอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



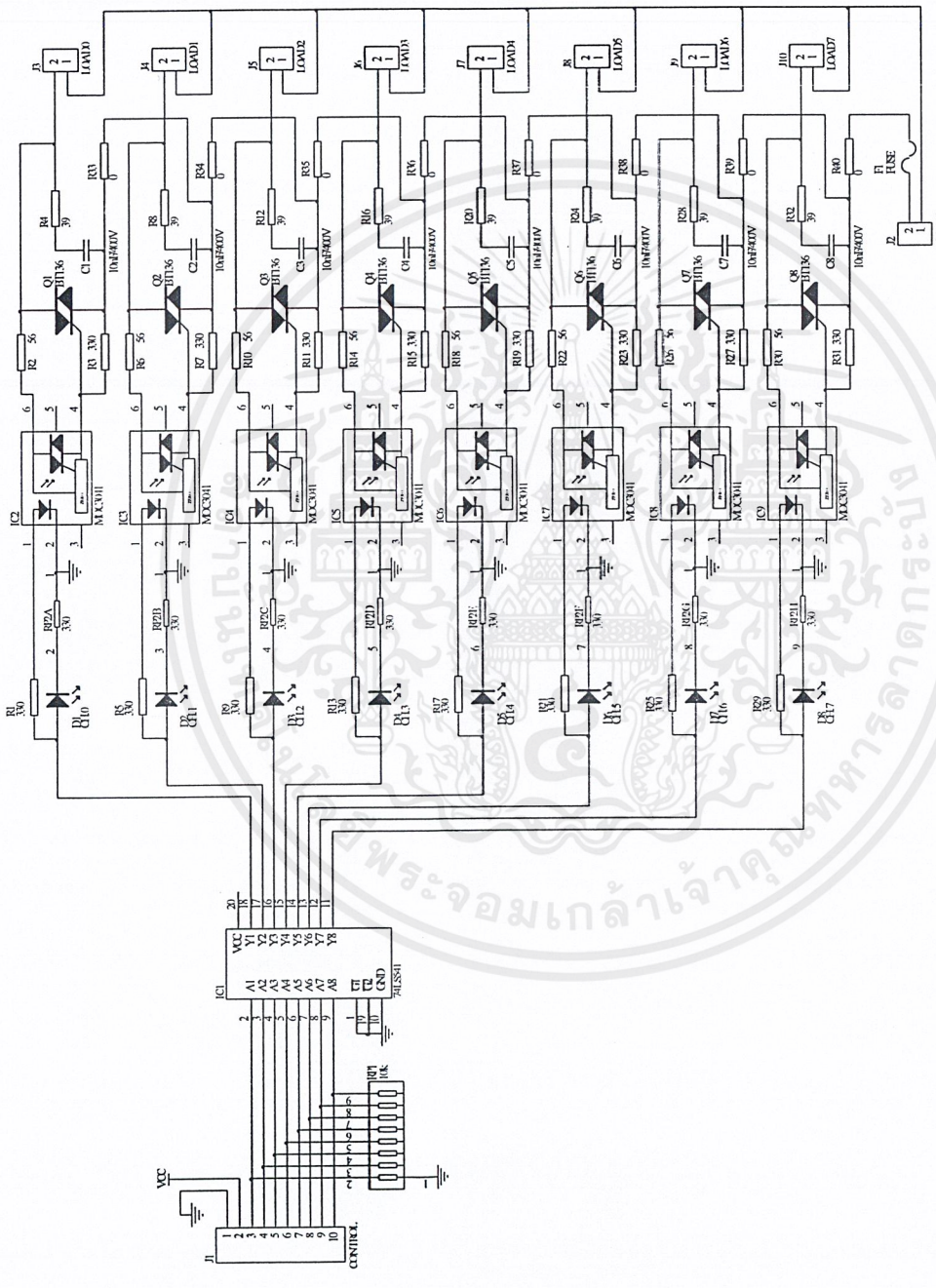
รูปที่ ค.10 วงจรสมบูรณ์ของโมดูลสื่อสารข้อมูลอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



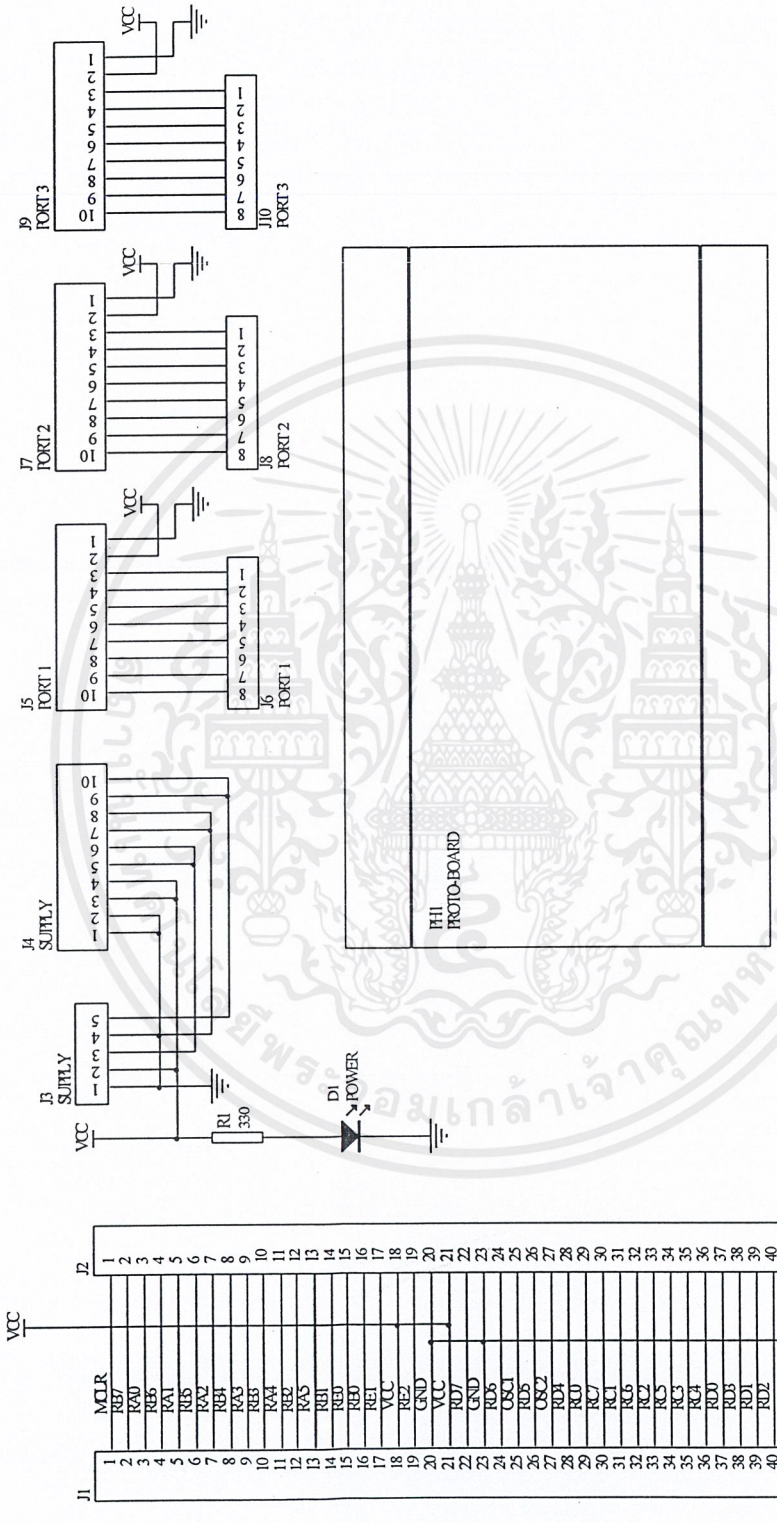
รูปที่ ค.11 วงจรสมบูรณ์ของโมดูลขับดีซีมอเตอร์และสเตปปิ้งมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



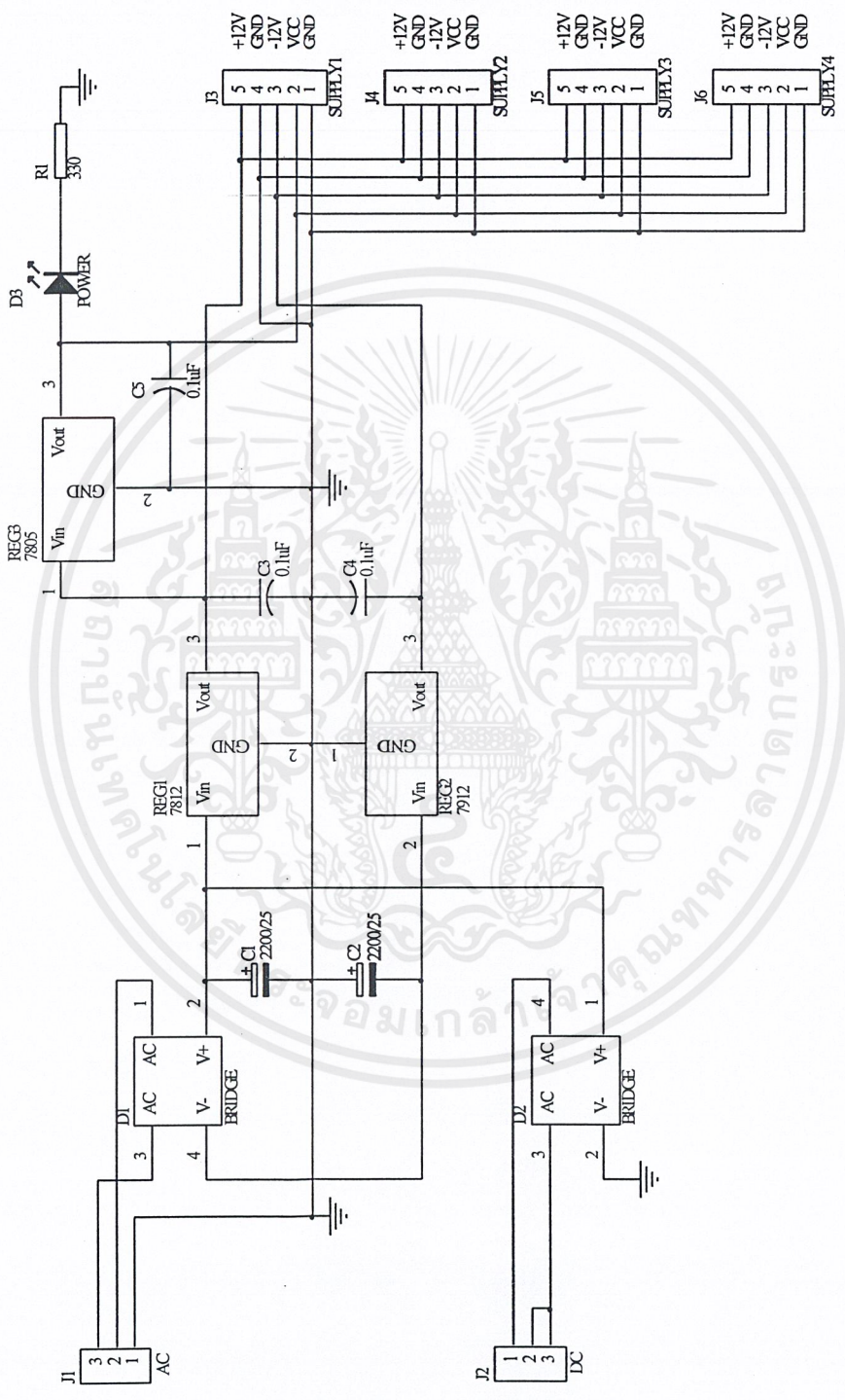
รูปที่ ค.12 วงจรสมบูรณ์ของโมดูลโซลิดสเตทรีเลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



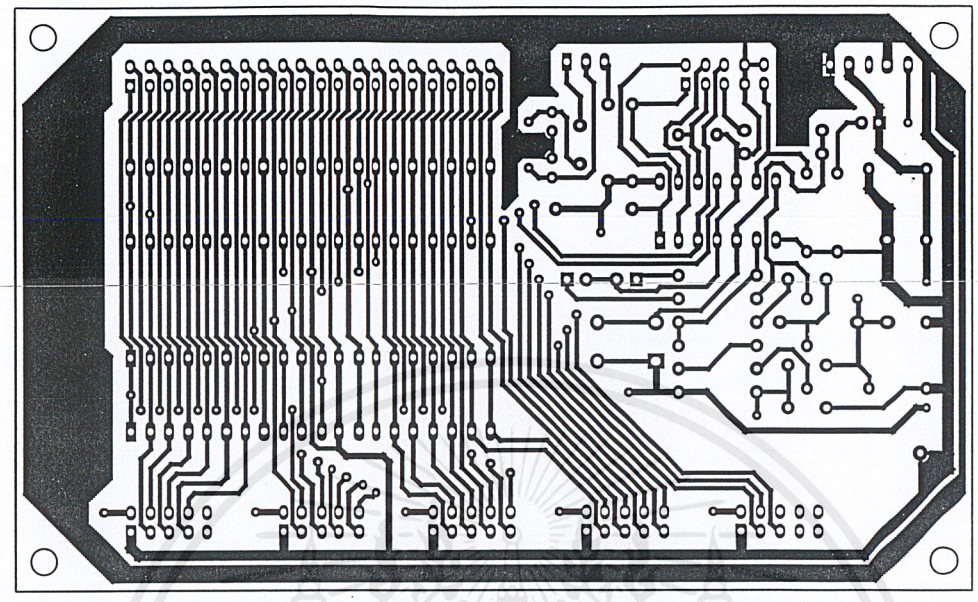
รูปที่ ค.13 วงจรสมบูรณ์ของโมดูลทดลองงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

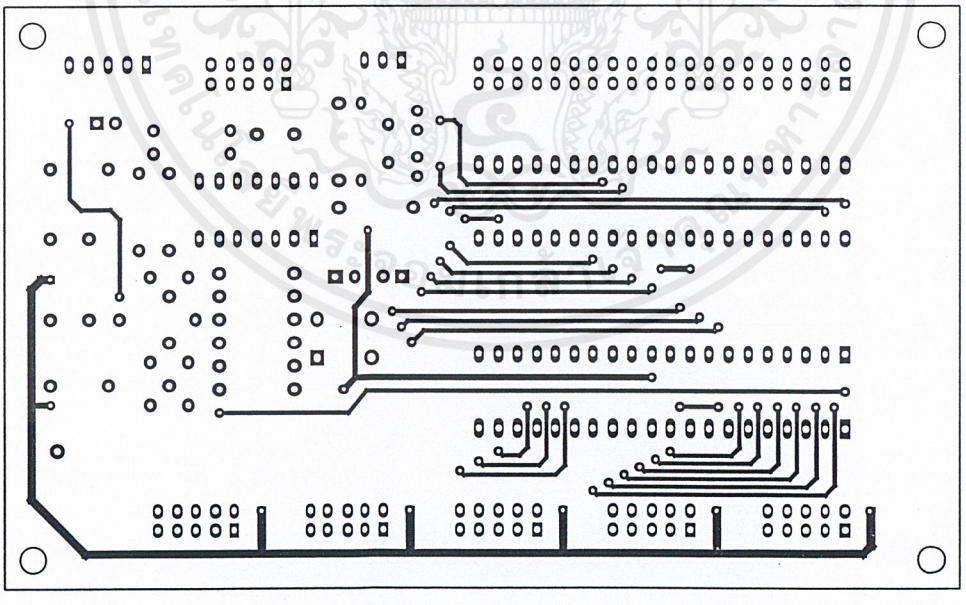


รูปที่ ค.14 วงจรสมบูรณ์ของโมดูลภาคจ่ายไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

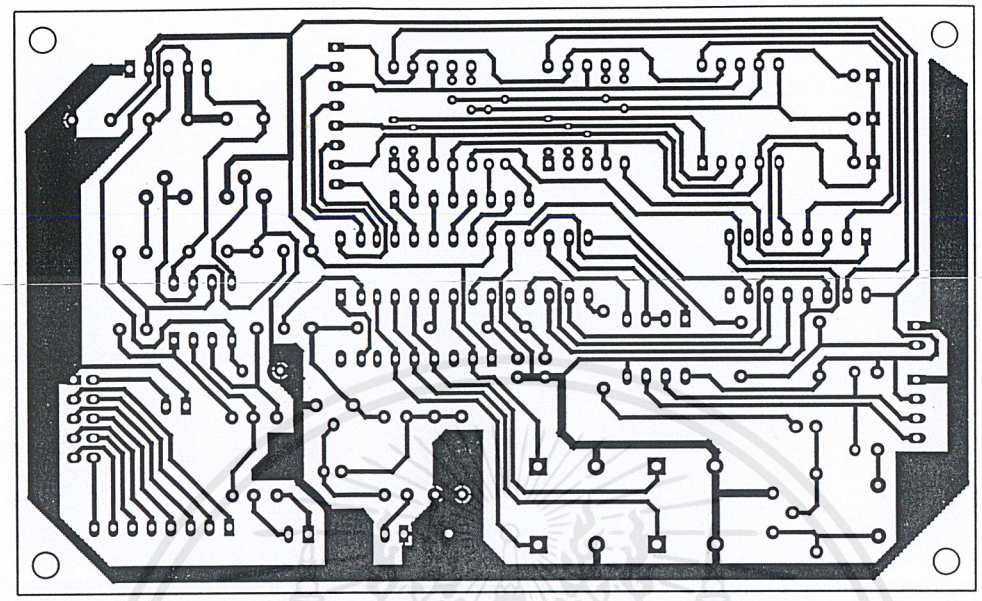


รูปที่ ค.15 ลายวงจรพิมพ์ด้านล่างของโมดูลหลัก PIC-ICSP

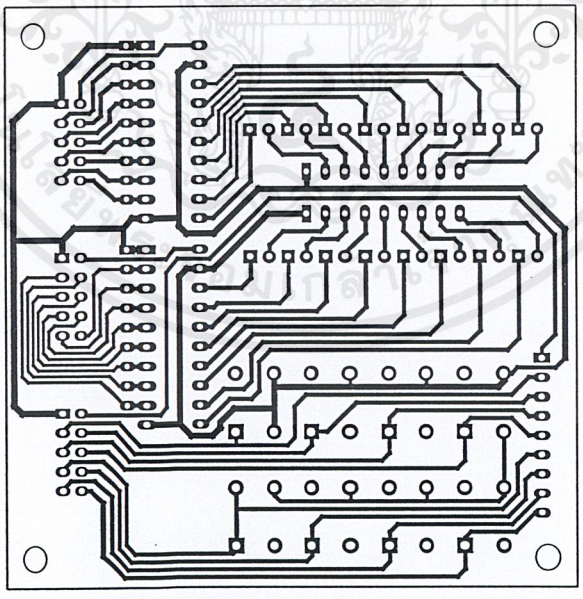


รูปที่ ค.16 ลายวงจรพิมพ์ด้านบนของโมดูลหลัก PIC-ICSP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

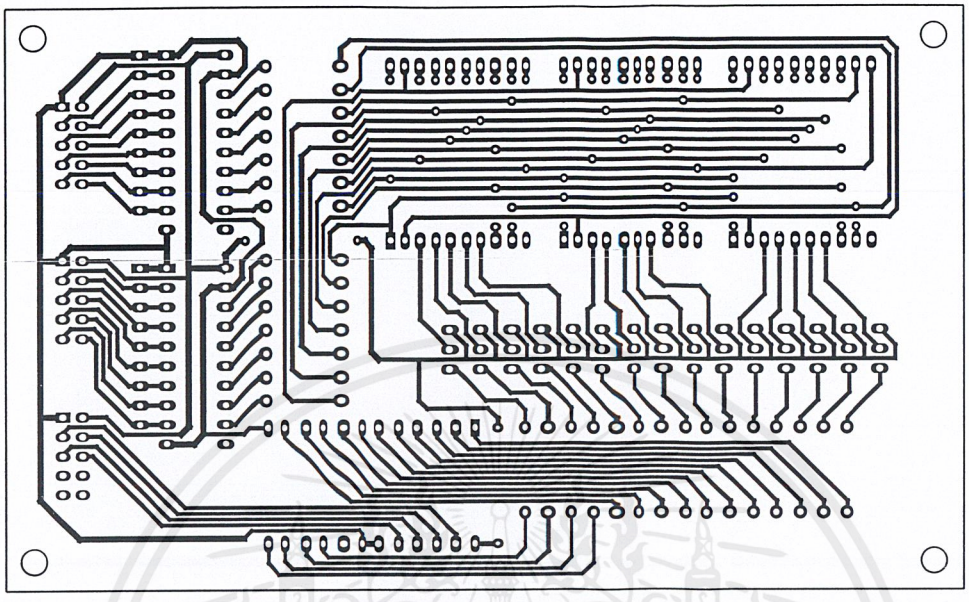


รูปที่ ค.17 ลายวงจรพิมพ์ด้านล่างของโมดูลเครื่องมือวัด

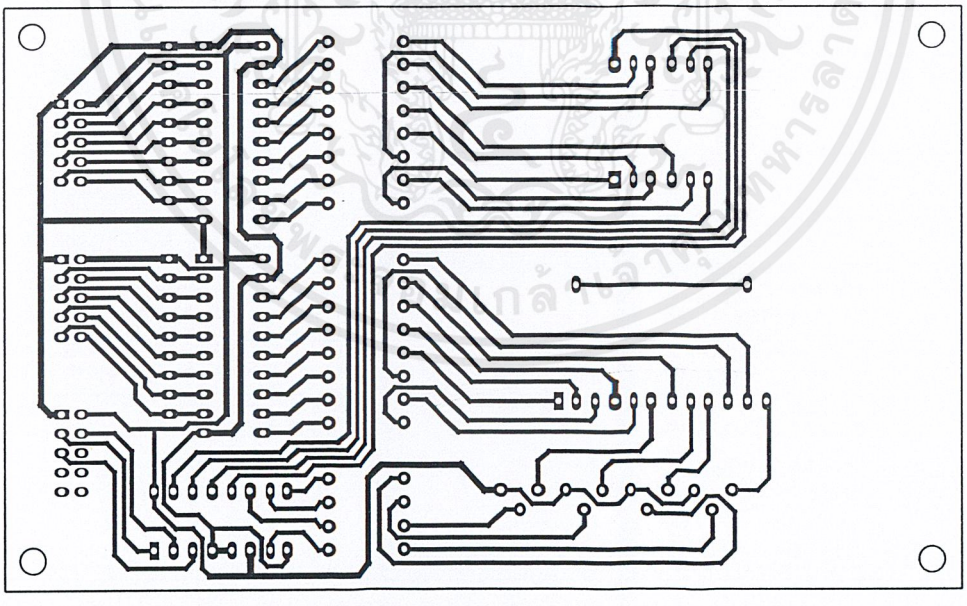


รูปที่ ค.18 ลายวงจรพิมพ์ด้านล่างของโมดูลเบสิกอินพุตเอาต์พุตพื้นฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

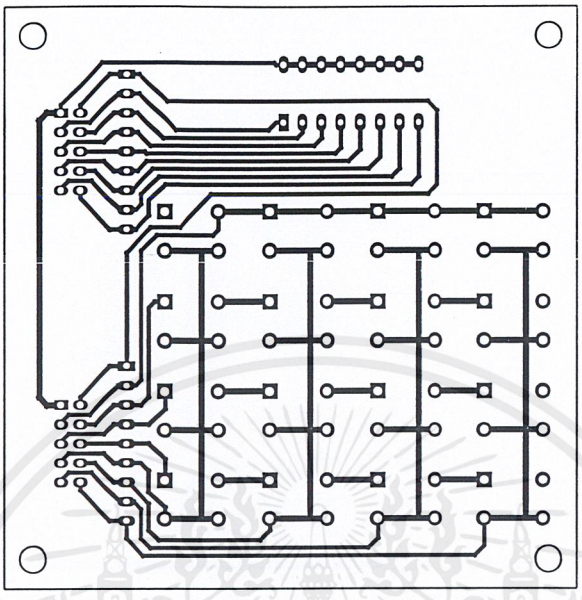


รูปที่ ค.19 ลายวงจรพิมพ์ด้านล่างของโมดูลแสดงผลแอลอีดีดอตเมตริกซ์

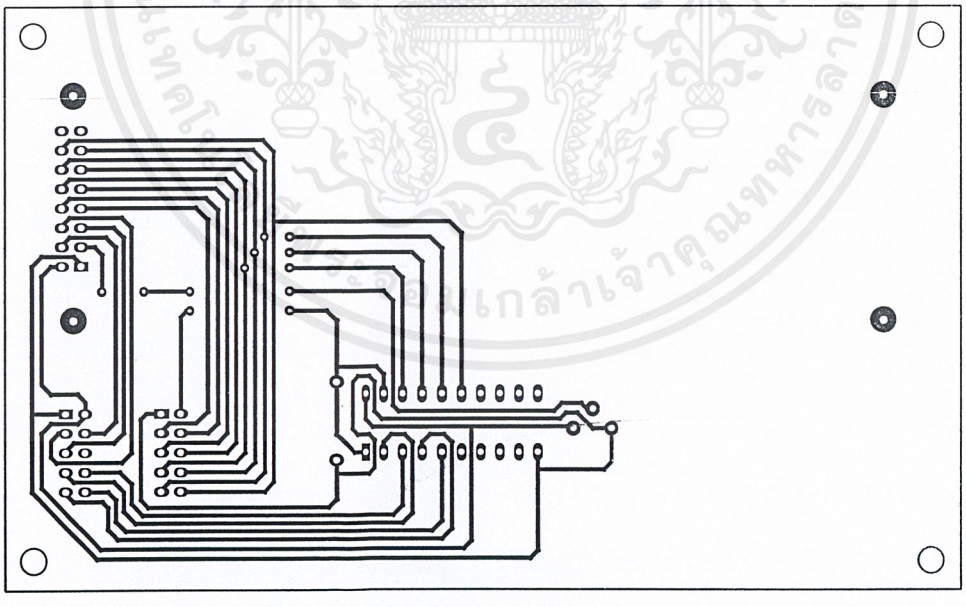


รูปที่ ค.20 ลายวงจรพิมพ์ด้านล่างของโมดูลแสดงผลตัวเลข 7 ส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

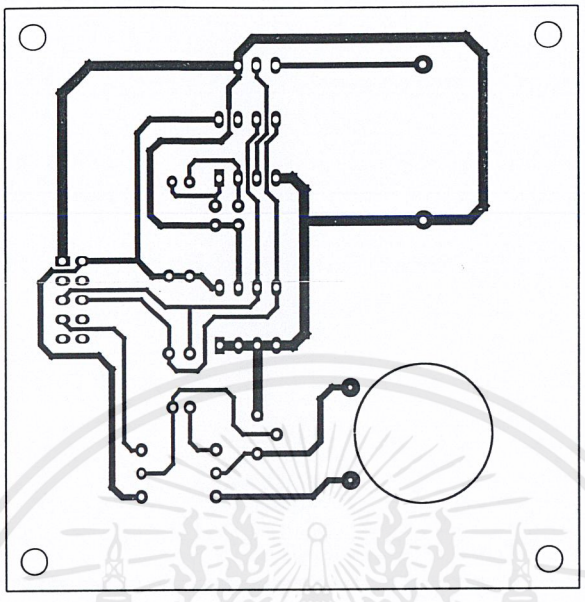


รูปที่ ค.21 ลายวงจรพิมพ์ด้านล่างของโมดูลดิพสวิตช์และเมตริกซ์สวิตช์

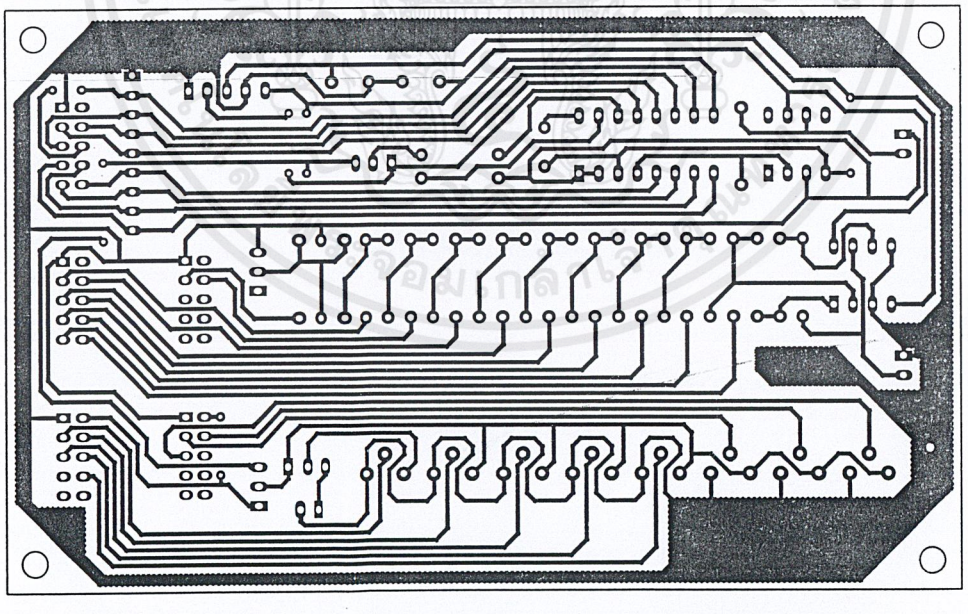


รูปที่ ค.22 ลายวงจรพิมพ์ด้านล่างของโมดูลแสดงผลแบบผลึกเหลว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

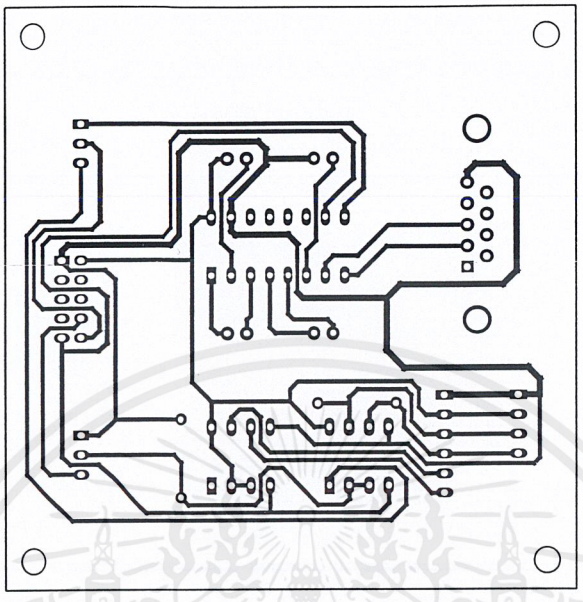


รูปที่ ค.23 ลายวงจรพิมพ์ด้านล่างของโมดูลเชื่อมต่ออุปกรณ์ RTC / EEPROM / SPEAKER

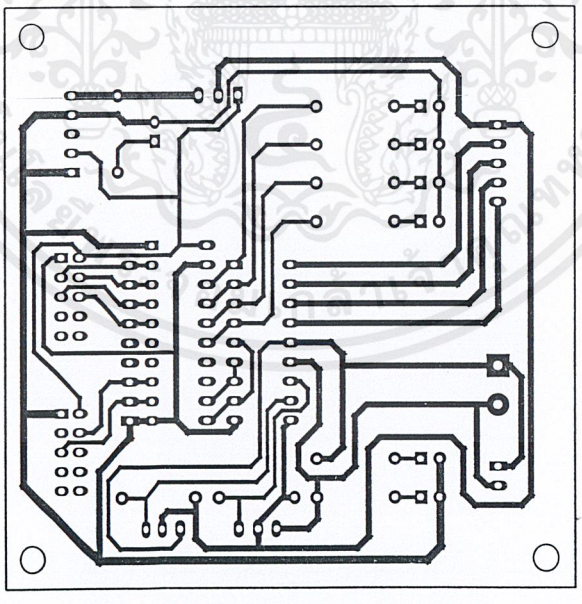


รูปที่ ค.24 ลายวงจรพิมพ์ด้านล่างของโมดูลแปลงสัญญาณแอนะลอกและดิจิตอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

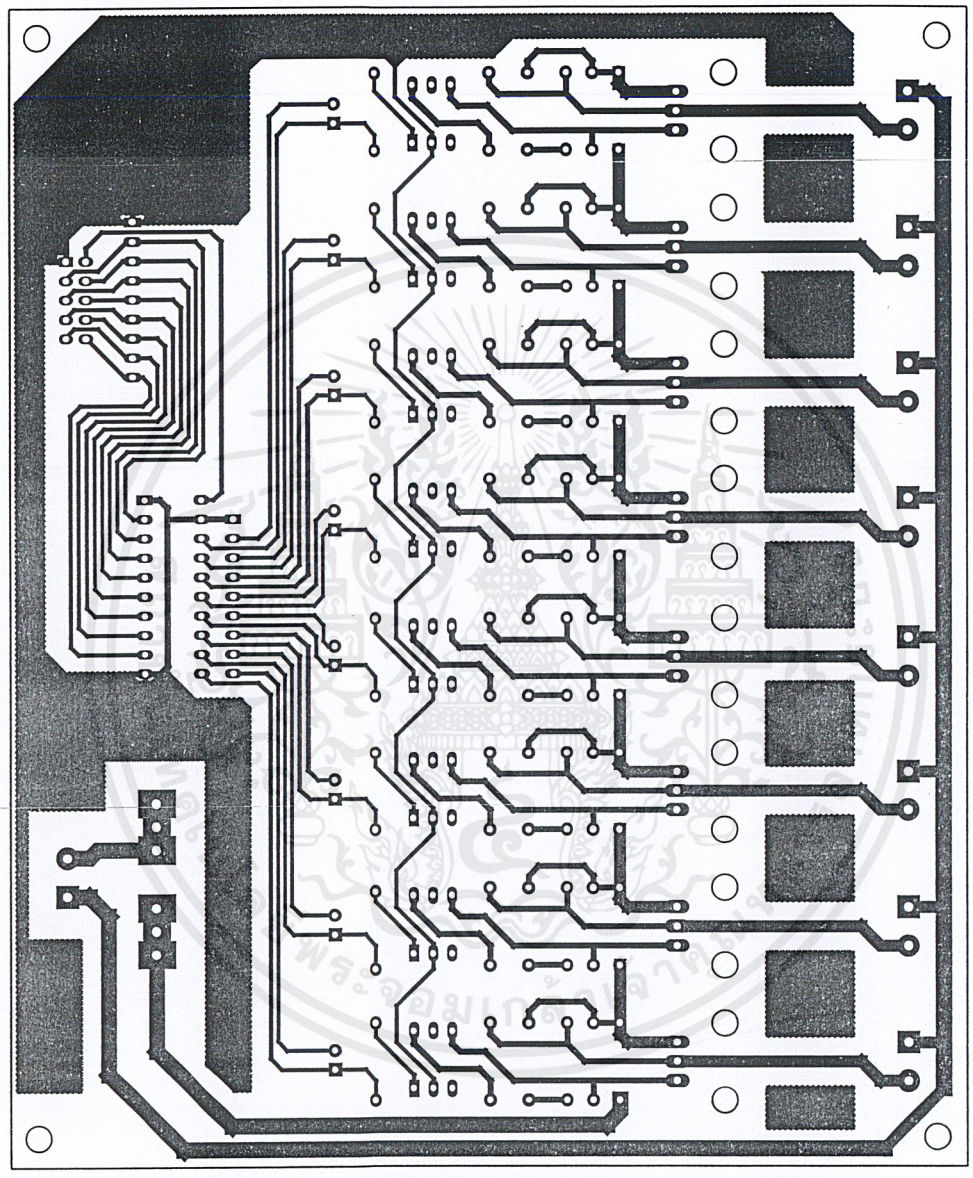


รูปที่ ค.25 ลายวงจรพิมพ์ด้านล่างของโมดูลสื่อสารข้อมูลอนุกรม



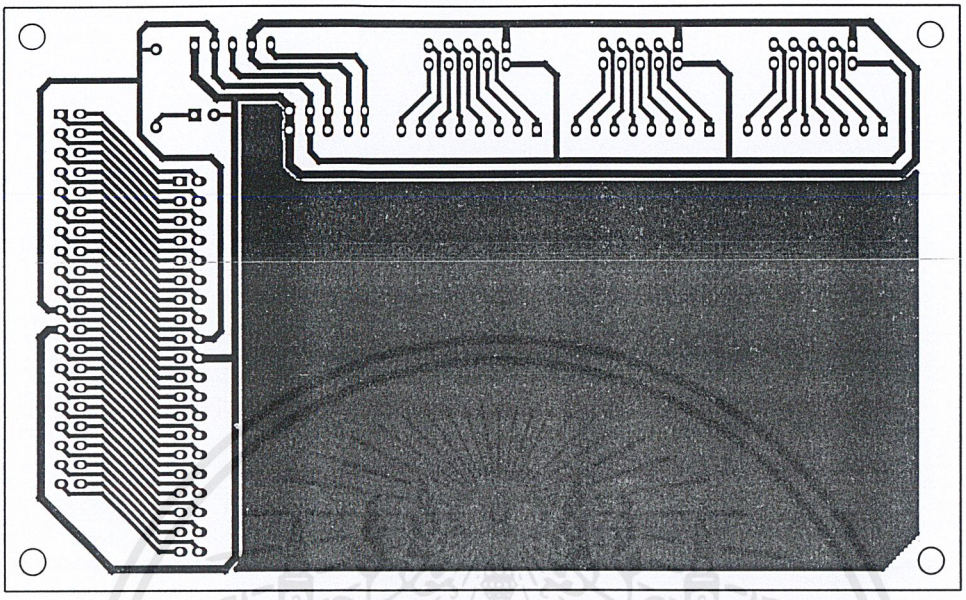
รูปที่ ค.26 ลายวงจรพิมพ์ด้านล่างของโมดูลขับดีซีมอเตอร์และสเตปปิ้งมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

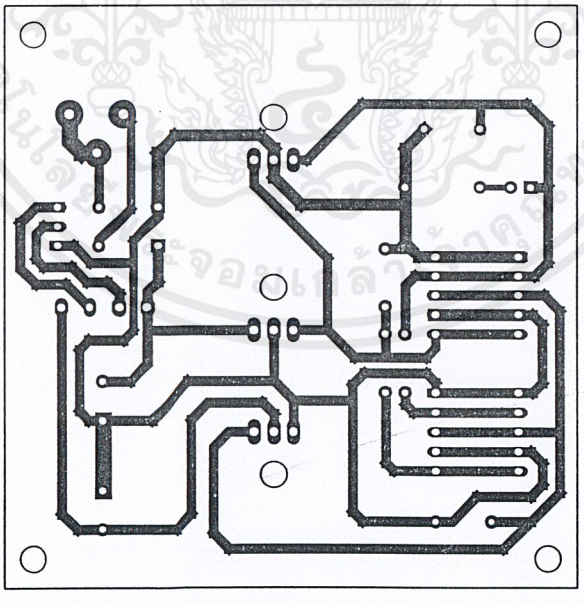


รูปที่ ค.27 ลายวงจรพิมพ์ด้านล่างของโมดูลโซลิดสเตทรีเลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค.28 ลายวงจรพิมพ์ด้านล่างของโมดูลทดลองวงจร



รูปที่ ค.29 ลายวงจรพิมพ์ด้านล่างของโมดูลภาคจ่ายไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง
รายการวัสดุและอุปกรณ์ที่ใช้ในการทำชุดทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายการวัสดุและอุปกรณ์ที่ใช้ในการทำชุดทดลอง

ไอซี (IC)

ลำดับ	รายการ	จำนวน
1.	74LS541	4 ตัว
2.	74LS540	2 ตัว
3.	74LS154	1 ตัว
4.	74LS138	1 ตัว
5.	75176	3 ตัว
6.	74LS125	1 ตัว
7.	ULN2803	1 ตัว
8.	MOC3041	8 ตัว
9.	74LS595	1 ตัว
10.	LF351	1 ตัว
11.	LF353N	1 ตัว
12.	MAX232	1 ตัว
13.	7812	1 ตัว
14.	7912	1 ตัว
15.	7805	1 ตัว
16.	DS1307	1 ตัว
17.	DAC0808	1 ตัว

ตัวเก็บประจุ (Capacitor)

ลำดับ	รายการ	จำนวน
1.	15 pF	2 ตัว
2.	22 pF	2 ตัว
3.	33 pF	2 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.	220 pF	1 ตัว
5.	10 nF	8 ตัว
6.	0.1 uF	9 ตัว
7.	1 uF	1 ตัว
8.	10 uF	4 ตัว
9.	100 uF	1 ตัว
10.	2200 uF	2 ตัว

ตัวต้านทาน (Resistor)

ลำดับ	รายการ	จำนวน
1.	10 โอห์ม	2 ตัว
2.	39 โอห์ม	8 ตัว
3.	56 โอห์ม	8 ตัว
4.	100 โอห์ม	15 ตัว
5.	150 โอห์ม	1 ตัว
6.	220 โอห์ม	18 ตัว
7.	330 โอห์ม	24 ตัว
8.	680 โอห์ม	2 ตัว
9.	820 โอห์ม	1 ตัว
10.	1 กิโลโอห์ม	11 ตัว
11.	2 กิโลโอห์ม	20 ตัว
12.	10 กิโลโอห์ม	20 ตัว
13.	20 กิโลโอห์ม	17 ตัว
14.	33 กิโลโอห์ม	3 ตัว
15.	120 กิโลโอห์ม	2 ตัว
16.	แบบแถว 330 โอห์ม	8 ตัว
17.	แบบแถว 10 กิโลโอห์ม	21 ตัว
18.	ปรับค่าได้ 10 กิโลโอห์ม	3 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

19	ปรับค่าได้ 50 กิโลโอห์ม	1 ตัว

ไดโอดเปล่งแสง (LED)

ลำดับ	รายการ	จำนวน
1.	LED กลมขนาด 3 mm (เขียว)	16 ดวง
2.	LED กลมขนาด 3 mm (แดง)	19 ดวง
3.	LED กลมขนาด 3 mm (เหลือง)	4 ดวง
4.	7-Segment Common Cathode 4 หลักคู่	1 ตัว
5.	7-Segment Common Anode 4 หลักคู่	1 ตัว
6.	7-Segment Common Anode 2 หลักคู่	1 ตัว

ทรานซิสเตอร์ (Transistor)

ลำดับ	รายการ	จำนวน
1.	BC547	4 ตัว
2.	BC557	6 ตัว
3.	BD139	2 ตัว
4.	BD140	17 ตัว

คอนเนคเตอร์ (Connector)

ลำดับ	รายการ	จำนวน
1.	2 ขา	4 ตัว
2.	3 ขา	18 ตัว
3.	4 ขา	2 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.	5 ขา	14 ตัว
5.	6 ขา	1 ตัว

สวิตซ์ (Switch)

ลำดับ	รายการ	จำนวน
1.	Switch กดติดปล่อยดับ	25 ตัว
2.	DIP Switch 8 ตำแหน่ง	1 ตัว

ไดโอด (Diode)

ลำดับ	รายการ	จำนวน
1.	1N4148	2 ตัว
2.	Diode Bridge	2 ตัว

รายการอื่นๆ

ลำดับ	รายการ	จำนวน
1.	ไตรแอคเบอร์ BT136	8 ตัว
2.	คริสตัลความถี่ 4 เมกะเฮิรตซ์	2 ตัว
3.	LCD Character ขนาด 24 ตัวอักษร 2 บรรทัด	1 ตัว
4.	ลำโพงบัซเซอร์	1 ตัว
5.	ลำโพง 1 W	1 ตัว
6.	ดีซีมอเตอร์	1 ตัว
7.	สเตปปิ้งมอเตอร์	1 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก จ

ใบงานการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 1
การใช้งานพอร์ตของ PIC16F876

จุดประสงค์

1. เพื่อให้สามารถเขียนโปรแกรมใช้งานพอร์ต (PORT) เป็นพอร์ตอินพุต (Input) ได้
2. เพื่อให้สามารถเขียนโปรแกรมใช้งานพอร์ตเป็นพอร์ตเอาต์พุต (Output) ได้
3. เพื่อให้สามารถเขียนโปรแกรมควบคุมเอาต์พุตจากอินพุตพอร์ตได้

อุปกรณ์ที่ใช้ในการทดลอง

1. โมดูลหลัก PIC-ICSP
2. โมดูลเบสิกอินพุตเอาต์พุต
3. คอมพิวเตอร์พีซีที่ติดตั้งโปรแกรม MPLAB
4. สายเชื่อมต่อระหว่างโมดูลหลัก PIC-ICSP กับคอมพิวเตอร์พีซี
5. สายเชื่อมต่อพอร์ตระหว่างโมดูล

ทฤษฎีเบื้องต้น

ข้อกำหนดทางฮาร์ดแวร์ (Hardware) และการกำหนดทิศทางของพอร์ตอินพุตเอาต์พุต

ไมโครคอนโทรลเลอร์ PIC16F876 มีพอร์ตสำหรับติดต่อกับอุปกรณ์ภายนอก 3 พอร์ต คือ พอร์ต A มีขนาด 6 บิต พอร์ต B และพอร์ต C มีขนาด 8 บิต ซึ่งกระบวนการทางซอฟท์แวร์ (Software) จะเป็นตัวกำหนดให้แต่ละพอร์ตทำหน้าที่เป็นอินพุตเอาต์พุตได้อย่างอิสระ

ในกรณีที่ใช้งานเป็นพอร์ตเอาต์พุต ขาแต่ละขาสามารถจ่ายกระแสออกได้สูงสุด 25 มิลลิแอมป์ และสามารถรับกระแสสูงสุดต่อขาได้ 20 มิลลิแอมป์ จึงสามารถนำไปขับไดโอดเปล่งแสงได้โดยต่อตัวต้านทานจำกัดกระแสเข้าไป

รีจิสเตอร์ TRISA, TRISB และ TRISC สำหรับกำหนดทิศทางการถ่ายทอดข้อมูลของพอร์ต A, B และ C ตามลำดับ เมื่อต้องการใช้งานเป็นอินพุตจะต้องเขียนข้อมูล “1” ไปยังบิตที่ต้องการกำหนดให้เป็นอินพุต และถ้าหากต้องการใช้งานเป็นเอาต์พุตก็จะต้องเขียนข้อมูล “0” ไปยังบิตที่ต้องการให้เป็นเอาต์พุต เช่น รีจิสเตอร์ TRISA บิต 0 (TRISA, 0) ใช้กำหนดทิศทางของ RA0 หรือบิต 1 ของพอร์ต B (TRISB, 1) ใช้กำหนดทิศทางของ RB1 และบิตอื่นๆ ไล่ไปตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

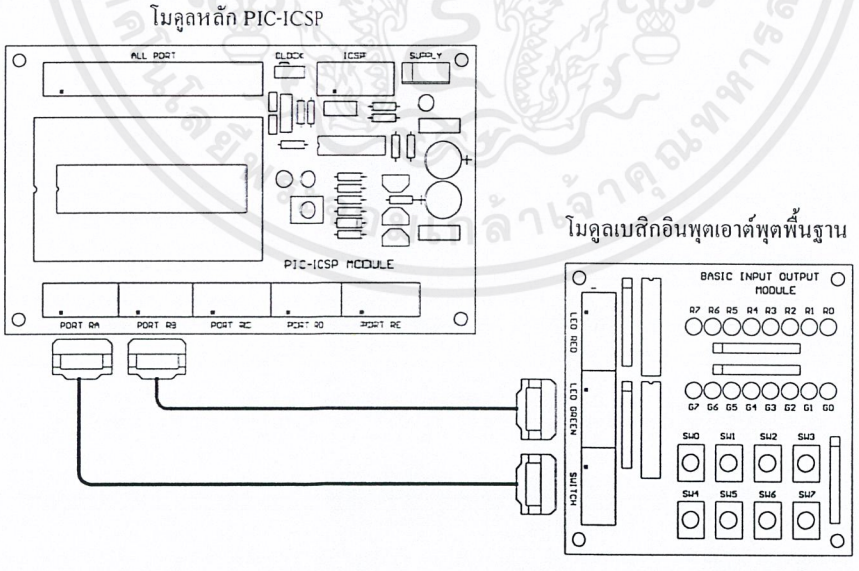
การเขียนโปรแกรมเพื่อกำหนดทิศทางของพอร์ตนั้น จะเริ่มจากการเลือกแบงก์ (Bank) ของรีจิสเตอร์ไฟล์ไปที่แบงก์ 1 โดยการเซตบิต RP0 ในรีจิสเตอร์ STATUS จากนั้นเขียนค่า "0" หรือ "1" ซึ่งเป็นการกำหนดทิศทางว่าเป็นอินพุตหรือเอาต์พุต ไปยังรีจิสเตอร์ TRISA, TRISB หรือ TRISC ตามตำแหน่งบิตที่ต้องการ โดยแอดเดรส (Address) ที่อ้างถึงจะอยู่ที่ตำแหน่ง 0x85, 0x86 และ 0x87 ตามลำดับ

หลังจากการกำหนดทิศทางของพอร์ตแล้ว ก็ต้องทำการเลือกแบงก์ของรีจิสเตอร์ไฟล์ให้กลับมายังแบงก์ 0 เพื่อเขียนหรืออ่านข้อมูลกับพอร์ตจริงๆ โดยผ่านทางรีจิสเตอร์ PORTA, PORTB และ PORTC วิธีการก็คือ เคลียร์บิต RP0 ของรีจิสเตอร์ STATUS

```
bsc     STATUS,RP0      ; เลือกแบงก์ 1
movlw   0x06
movwf   ADCON1          ; พอร์ต A เป็นดิจิตอลอินพุต
movlw   b'11001111'
movwf   TRISA           ; เซต RA0-RA3 เป็นอินพุต ส่วน RA4-RA5 เป็นเอาต์พุต
bcf     STATUS,PR0

clrf    PORTA           ; การเคลียร์พอร์ต A
```

รูปที่ จ.1 ตัวอย่างโปรแกรมย่อยเป็นการอินิเชียลพอร์ต A



รูปที่ จ.2 การเชื่อมต่อระหว่างโมดูลหลัก PIC-ICSP และโมดูลเบสิกอินพุตเอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   List p=16f876
   #include "p16f876.inc"
   __config _WDT_OFF & _PWRTE_ON & _LVP_OFF & _XT_OSC
Count1      equ     0x20
Count2      equ     0x21

            org     0x00
            bsf     STATUS,RP0
            clrf    PORTB           ;พอร์ต B เป็นเอาต์พุต
            bcf     STATUS,RP0      ;เคลียร์ STATUS

Start       movlw   0xff            ;LED ติดทุกดวง
            movwf   PORTB
            call    Delay
            movlw   0x00            ;LED ดับทุกดวง
            movwf   PORTB
            call    Delay
            goto    Start

;------------ หน่วงเวลา ---------------
Delay       movlw   0xff
            movwf   Count1
Loop2       movlw   0xff
            movwf   Count2
Loop1       decfsz  Count2,F
            goto    Loop1
            decfsz  Count1,F
            goto    Loop2
            return

            end
```

รูปที่ จ.3 โปรแกรมการใช้งานพอร์ตของ PIC16F876

```
   List p=16f876
   Include "p16f876.inc"
__config _WDT_OFF & _PWRTE_ON & _LVP_OFF & _XT_OSC
Count1      equ     0x20
Count2      equ     0x21

            org     0x00
            bsf     STATUS,RP0
            movlw   0x06            ;พอร์ต A เป็นดิจิตอลอินพุต
            movwf   ADCON1
            movlw   0xff
            movwf   PORTA
            clrf    PORTB           ;พอร์ต B เป็นเอาต์พุต
            bcf     STATUS,RP0
            clrf    PORTB
SW1         btfsc   PORTA,0         ;ตรวจสอบการกดสวิตช์ SW1
            goto    SW2
            movlw   0x01
            movwf   PORTB
            call    Delay

SW2         btfsc   PORTA,1         ;ตรวจสอบการกดสวิตช์ SW2
            goto    SW3
            movlw   0x02
            movwf   PORTB
            call    Delay
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SW3         btfsc   PORTA,2         ;ตรวจสอบการกดสวิตช์ SW3
            goto    SW4
            movlw   0x04
            movwf   PORTB
            call    Delay

SW4         btfsc   PORTA,3         ;ตรวจสอบการกดสวิตช์ SW4
            goto    SW5
            movlw   0x08
            movwf   PORTB
            call    Delay

SW5         btfsc   PORTA,4         ;ตรวจสอบการกดสวิตช์ SW5
            Goto    SW6
            Movlw   0x10
            Movwf   PORTB
            Call    Delay

SW6         btfsc   PORTA,5         ;ตรวจสอบการกดสวิตช์ SW6
            Goto    SW1
            Movlw   0x20
            Movwf   PORTB
            Call    Delay
            Goto    SW1

;------------ หน่วงเวลา ----------------
Delay       movlw   0xFF
            Movwf   Count1
Loop1       movlw   0xFF
            Movwf   Count2
Loop2       decfsz  Count2,F
            Goto    Loop2
            Decfsz  Count1,F
            Goto    Loop1
            Return

            End
```

รูปที่ จ.4 โปรแกรมการควบคุมเอาต์พุตจากอินพุตพอร์ตของ PIC16F876

ลำดับขั้นตอนการทดลอง

1. เปิดโปรแกรม MPLAB ขึ้นมาใช้งานเพื่อเขียนโปรแกรม
2. ไปที่เมนู OPTION เลือก Development Mode กำหนดโหมดเป็น MPLAB-SIM Simulator
3. เลือก Process เป็นรุ่น PIC16F876 จากนั้นกดปุ่ม Apply แล้วกด OK
4. ไปที่เมนู Project เลือก New Project ตั้งชื่อ Project Name แล้วกด OK
5. ในเมนู Edit Project ให้ “Remove” Project Files ออกให้หมดถ้ามีไฟล์อยู่จากนั้นกด OK
6. ไปที่เมนู File เลือก New Source เพื่อเขียนโปรแกรมดังรูปที่ จ.3 ลงไป
7. จากนั้น save file เป็นนามสกุล . asm จากนั้นทำการแอสเซมเบลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. ไปที่เมนู Project เลือก Edit Project แล้วกำหนดชื่อไฟล์เป็น ชื่อที่เราบันทึกไว้ในข้อ 6. เพื่อ Add ไปยัง Project Files

9. ไปที่เมนู Project เลือก Make Project เพื่อแอสเซมเบลอร์โปรแกรม (หลังจากแอสเซมเบลอร์จะได้ไฟล์เป็นนามสกุล .hex)

10. ตรวจสอบผลการแอสเซมเบลอร์ว่ามีข้อผิดพลาดหรือไม่ ถ้ามีจัดการแก้ไขข้อผิดพลาดที่เกิดขึ้นให้เรียบร้อย แล้วทำตามขั้นตอนในข้อ 8 อีกครั้งจนไม่มีข้อผิดพลาดอีก

11. เขียนข้อมูลลงบน PIC16F876 โดยใช้โปรแกรม PIC-ICSP Device for Windows

12. ต่อสายเชื่อมต่อระหว่างโมดูลตามรูปที่ จ.2

13. สังเกตผลการทำงานของ LED ทั้ง 8 ดวง

.....

.....

14. ในบรรทัด Start movlw 0xFF ถ้าเปลี่ยนข้อมูลจาก 0xFF เป็น 0x01 จะได้ผลเป็นดังนี้

.....

.....

15. ถ้าต้องการให้ LED ดวงที่ 7 (บิตที่ 7) ติดและดับ จะต้องเปลี่ยนข้อมูลเป็นเท่าใด

.....

16. จากโปรแกรมดังรูปที่ จ.2 ถ้าต้องการให้พอร์ต A เป็นเอาต์พุตโดยให้ LED ติด-ดับทุกดวง จะต้องเขียนโปรแกรมอย่างไร

.....

.....

.....

.....

.....

.....

17. ทำการเขียนโปรแกรมดังรูปที่ จ.4 จากนั้น save file เป็นนามสกุล . asm ในหน้าต่าง edit จากนั้นทำการแอสเซมเบลอร์

18. เขียนข้อมูลลงบน PIC16F876 โดยใช้โปรแกรม PIC ICSP Device for Windows

19. ต่อสายเชื่อมต่อระหว่างโมดูลตามรูปที่ จ.2

20. เมื่อมีการกดสวิตซ์แต่ละตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ จ.1 ผลการติดของ LED จากการกดสวิตช์

สวิตช์	LED1	LED2	LED3	LED4	LED5	LED6
SW1						
SW2						
SW3						
SW4						
SW5						
SW6						

21. เมื่อไม่มีการกดสวิตช์ใดๆ สังเกตรูปแบบการแสดงผลของ LED แต่ละตัว

.....

.....

22. จากโปรแกรมดังรูปที่ จ.4 ถ้าต้องการให้กดสวิตช์ SW1 แล้ว LED ควงที่ 1 และ 2 ติด และกดสวิตช์ SW2 แล้ว LED ติดทุกดวงจะต้องแก้ไขโปรแกรมเป็นดังนี้

.....

.....

สรุปผลการทดลอง

.....

.....

.....

.....

.....

คำถามท้ายการทดลอง

1. จงอธิบายการทำงานของโปรแกรมดังรูปที่ จ.3 และโปรแกรมดังรูปที่ จ.4
2. บอกถึงความสัมพันธ์ระหว่างพอร์ต A, B กับรีจิสเตอร์ TRISA และรีจิสเตอร์TRISB
3. เขียนโปรแกรมแสดงรูปแบบไฟวิ่ง 4 แบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 2

การทำงานของ PIC16F876 กับวงจรกำเนิดสัญญาณนาฬิกา แบบ RC และแบบคริสตอล (Crystal)

จุดประสงค์

1. เพื่อให้สามารถใช้งานไมโครคอนโทรลเลอร์ PIC16F876 ให้มีอุปกรณ์ต่อพ่วงน้อยที่สุด
2. เพื่อให้สามารถปรับความเร็วการทำงานของไมโครคอนโทรลเลอร์ได้อย่างอิสระจากการปรับค่าความต้านทานของวงจรกำเนิดสัญญาณนาฬิกาแบบ RC
3. เพื่อให้สามารถเขียนโปรแกรมเพื่อใช้กับวงจรกำเนิดสัญญาณนาฬิกาแบบ RC และคริสตอลได้

อุปกรณ์ที่ใช้ในการทดลอง

1. โมดูลหลัก PIC-ICSP
2. โมดูลเบสิกอินพุตเอาต์พุต
3. คอมพิวเตอร์พีซีที่ติดตั้งโปรแกรม MPLAB
4. สายเชื่อมต่อระหว่างโมดูลหลัก PIC-ICSP กับคอมพิวเตอร์พีซี
5. สายเชื่อมต่อพอร์ตระหว่างโมดูล

ทฤษฎีเบื้องต้น

ไมโครคอนโทรลเลอร์ PIC16F876 สามารถใช้งานกับวงจรกำเนิดสัญญาณนาฬิกาได้ทั้งหมด 4 ชนิดด้วยกันคือ

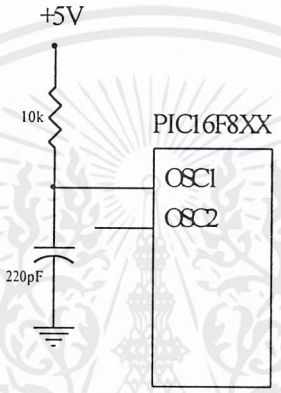
1. RC คือ การต่อตัวต้านทานและตัวเก็บประจุ
2. XT คือ เรโซเนเตอร์หรือคริสตอล
3. HS คือ คริสตอลความเร็วสูง
4. LP คือ คริสตอลกำลังงานต่ำ

ในการทดลองนี้เราจะกล่าวถึงเฉพาะวงจรกำเนิดสัญญาณนาฬิกาแบบ RC และแบบ XT เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรกำเนิดสัญญาณนาฬิกาแบบ RC

วงจรกำเนิดสัญญาณนาฬิกาแบบ RC นั้นจะมีข้อแตกต่างจากแบบอื่นคือเป็นการใช้วงจรสัญญาณนาฬิกาจากภายในแล้วต่อกับตัวต้านทานและตัวเก็บประจุที่ต่อจากภายนอกเพื่อกำหนดค่าความถี่ตามต้องการ ซึ่งจะทำให้ประหยัดอุปกรณ์ในการต่อจากภายนอกไมโครคอนโทรลเลอร์และช่วยให้วงจรมีขนาดเล็กลง ในขณะที่วงจรกำลังทำงานอยู่เราสามารถปรับสัญญาณนาฬิกาได้ 0-4 เมกะเฮิรตซ์ ลักษณะการต่อแบบ RC จะเป็นดังรูปที่ จ.16



รูปที่ จ.5 การต่อวงจรกำเนิดสัญญาณนาฬิกาแบบ RC

วงจรกำเนิดสัญญาณนาฬิกาก็มีข้อเสียคือ ไม่สามารถสร้างสัญญาณนาฬิกาที่มีความเที่ยงตรงสูงๆ ได้เพราะค่าความต้านทานและตัวเก็บประจุจะเปลี่ยนแปลงค่าตามสภาพแวดล้อมทำให้ค่าทั้งสองไม่คงที่ ค่าความต้านทานที่ทำให้เกิดเสถียรภาพที่ดีต้องอยู่ในช่วง 3-100 กิโลโอห์ม ถ้าค่าความต้านทานมากกว่า 100 กิโลโอห์ม สัญญาณนาฬิกาจะมีสัญญาณรบกวนมาก ส่วนค่าตัวเก็บประจุก็เหมือนกันจะต้องมีค่าตั้งแต่ 20 พิโกฟารัดขึ้นไป ถ้าค่าน้อยกว่า 20 พิโกฟารัดจะทำให้เกิดสัญญาณรบกวนได้ง่าย นอกจากนี้ค่าความจุที่เกิดจากลายทองแดง บนแผ่นวงจรและค่าความจุระหว่างขาและตัวถังของตัวเก็บประจุก็อาจส่งผลต่อสัญญาณนาฬิกาได้เช่นกัน วงจรกำเนิดสัญญาณนาฬิกาแบบ RC จะไม่มีสูตรการคำนวณเพื่อเลือกค่าความถี่ที่แน่นอนแต่จะมีค่าโดยประมาณซึ่งกำหนดมาจากผู้ผลิตดังตารางต่อไปนี้

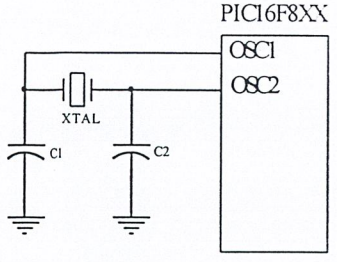
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ จ.2 ตัวอย่างความถี่สัญญาณนาฬิกาของวงจรกำเนิดสัญญาณนาฬิกาแบบ RC

ค่าตัวเก็บประจุ	ค่าความต้านทาน	ค่าความถี่
20pF	3.3kΩ	4.68MHz
20pF	5.1kΩ	3.94 MHz
20pF	10 kΩ	2.34 MHz
20pF	100 kΩ	250.16 kHz
100pF	3.3kΩ	1.49 MHz
100pF	5.1kΩ	1.12 MHz
100pF	10 kΩ	620.31 kHz
100pF	100 kΩ	90.25 kHz
300pF	3.3kΩ	524.24 kHz
300pF	5.1kΩ	415.52 kHz
300pF	10 kΩ	270.33 kHz
300pF	100 kΩ	25.37 kHz

วงจรกำเนิดสัญญาณนาฬิกาแบบ XT

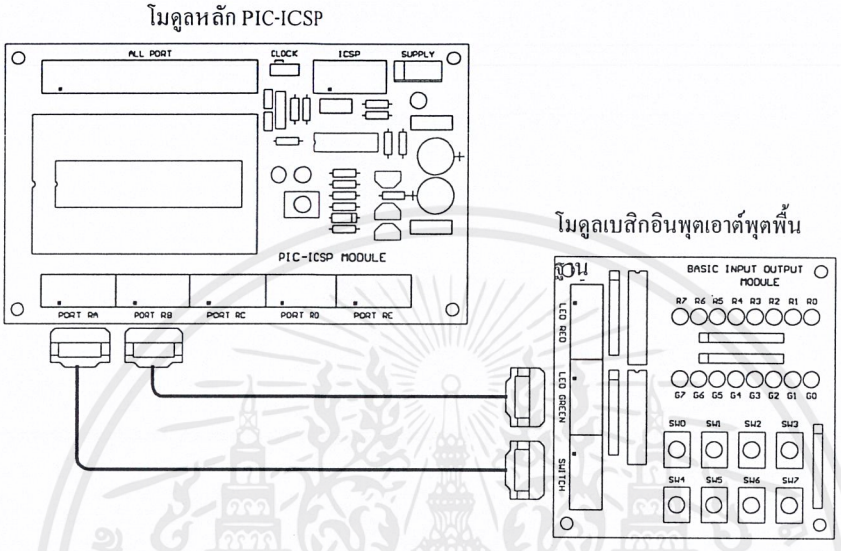
เป็นวงจรกำเนิดสัญญาณนาฬิกาที่ใช้คริสตอลหรือเรโซเนเตอร์ มีความเที่ยงตรงสูงมาก การคำนวณเกี่ยวกับการหน่วงเวลาในการเขียนโปรแกรมจะทำให้แม่นยำมากขึ้นสัญญาณจะป้อนเข้าที่ขา OSC1 ของไมโครคอนโทรลเลอร์ PIC16F87X จะทำให้ได้ความถี่ต่ำลง 4 วงจรกำเนิดสัญญาณนาฬิกาโดยใช้คริสตอลแสดงดังรูปที่ จ.6



รูปที่ จ.6 การต่อวงจรกำเนิดสัญญาณนาฬิกาแบบ XT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการต่อวงจรกำเนิดสัญญาณนาฬิกาแบบ RC ให้กับไมโครคอนโทรลเลอร์ PIC16F876 นั้นห้ามนำสัญญาณนาฬิกาจากภายนอกต่อที่ OSC1 เด็ดขาดเนื่องจากอาจทำให้ไมโครคอลโทรลเลอร์เกิดความเสียหายได้



รูปที่ จ.7 การเชื่อมต่อระหว่างโมดูลหลัก PIC-ICSP และโมดูลเบสิกอินพุตเอาต์พุต

```
   List      p=16F84
   #include "p16f876.inc"
__config _WDT_OFF & _PWRTE_ON & _LVP_OFF & _RC_OSC
COUNT1      equ     0x20
COUNT2      equ     0x21
            org     0x00
Start       bsf     STATUS,RP0
            clrf    PORTB
            bcf     STATUS,RP0
            clrf    PORTB
            movlw   0x01
            movwf   PORTB
            bcf     STATUS,C        ; เคลียร์แฟลกทด
Loop        call    Delay
            rlf     PORTB,f
            goto    Loop
;---------- DELAY TIME ----------
Delay       movlw   0x20
            movwf   COUNT1
Delay1      movlw   0xFF
            movwf   COUNT2
Delay2      decfsz  COUNT2,f
            goto    Delay2
            decfsz  COUNT1,f
            goto    Delay1
            return
            end
```

รูปที่ จ.8 โปรแกรมใช้ในการทดลองการกำเนิดสัญญาณนาฬิกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับขั้นตอนการทดลอง

1. เขียนโปรแกรมดังรูปที่ จ.8 และทำการแอสเซมเบอร์
2. เขียนข้อมูลลงบนตัว PIC16F876
3. ต่อสายเชื่อมต่อโมดูลตามรูปที่ จ.7 และทำการเซตจั๊มเปอร์ไปที่โหมด RC ซึ่งเป็นการกำหนดให้วงจรทำงานในโหมด RC โดยให้พอร์ต A ต่ออยู่กับสวิตช์ และพอร์ต B ต่ออยู่กับ LED RED หรือ LED GREEN แต่ในการทดลองนี้จะใช้ LED GREEN
4. สังเกตผลการทำงานของ LED ทั้ง 8 ดวงที่ต่ออยู่กับพอร์ต B

.....

.....

5. เปลี่ยนส่วนของวงจรกำเนิดสัญญาณนาฬิกามาเป็นแบบ XT โดยเปลี่ยนการเซตจั๊มเปอร์
6. เปลี่ยนโปรแกรมรูปที่ จ.8 บรรทัดที่ 3 กำหนดสัญญาณนาฬิกาจาก _RC_OSC เป็น _XT_OSC แล้วทำการแอสเซมเบอร์ใหม่ แล้วเขียนข้อมูลลงบน PIC16F876 อีกครั้ง
7. สังเกตผลการทำงานของไดโอดเปล่งแสงทั้ง 8 ดวงเปรียบเทียบกับผลที่ได้จากการทดลองข้อ 4 จะได้

.....

.....

.....

สรุปผลการทดลอง

.....

.....

.....

.....

.....

คำถามท้ายการทดลอง

1. จงบอกข้อดีในการสร้างวงจรกำเนิดสัญญาณนาฬิกาทั้งแบบ RC และ XT มาพอเข้าใจ
2. จงอธิบายการทำงานของโปรแกรมดังรูปที่ จ.8 พอเข้าใจ
3. จงเขียนโปรแกรมแสดงไฟกระพริบ 8 ดวง โดยใช้วงจรกำเนิดสัญญาณนาฬิกาแบบ XT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 3
การอินเตอร์รัพต์ PIC16F876

จุดประสงค์

1. เพื่อให้สามารถเซตบิตที่สำหรับตอบสนองการอินเตอร์รัพต์ใน รีจิสเตอร์ INCON ได้
2. เพื่อให้สามารถอธิบายหลักการอินเตอร์รัพต์แบบต่างๆ ได้
3. เพื่อให้สามารถเขียนโปรแกรมใช้งานการอินเตอร์รัพต์แบบต่างๆได้

อุปกรณ์ที่ใช้ในการทดลอง

1. โมดูลหลัก PIC-ICSP
2. โมดูลเบสิกอินพุตเอาต์พุต
3. คอมพิวเตอร์พีซีที่ติดตั้งโปรแกรม MPLAB
4. สายเชื่อมต่อระหว่างโมดูลหลัก PIC-ICSP กับคอมพิวเตอร์พีซี
5. สายเชื่อมต่อพอร์ตระหว่างโมดูล

ทฤษฎีเบื้องต้น

การอินเตอร์รัพต์ คือ การขัดจังหวะการทำงานปกติของระบบไมโคร ปกติเมื่อเริ่มจ่ายไฟในระบบไมโครคอนโทรลเลอร์จะเกิดเพาเวอร์ออนรีเซต PIC16F876 จะเริ่มต้นการทำงานที่ตำแหน่ง 0x00 เสมอและทำงานไล่เรียงตามลำดับไปทีละแอดเดรส แต่ถ้าเกิดการอินเตอร์รัพต์เกิดขึ้น PIC16F876 จะประมวลผลและกระทำคำสั่งล่าสุดในขณะที่เกิดการอินเตอร์รัพต์จนเสร็จสิ้น แล้วจึงกระโดดไปทำงานที่ตำแหน่ง 0x04 ดังนั้นโปรแกรมย่อยของการบริการอินเตอร์รัพต์ (Interrupt Service Routine) จะเริ่มต้นที่ตำแหน่ง 0x04 เสมอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF

บิต 0 : RBIF เป็นแฟลกแสดงการเกิดอินเตอร์รัพต์เนื่องจากการเกิดการเปลี่ยนแปลงระดับลอจิกที่บิต 4-7 ของพอร์ต B (Port B Change Flag)

0 = ไม่มีการเปลี่ยนแปลงเกิดขึ้นที่ขา 4-7 ของพอร์ต B

1 = เกิดการเปลี่ยนแปลงขึ้นที่ขาใดขาหนึ่งของบิต 4-7 ของพอร์ต B

จะต้องเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

บิต 1 : INTF เป็นแฟลกแสดงการเกิดอินเตอร์รัพต์เนื่องจากสัญญาณภายนอก (External Interrupt Flag)

0 = ไม่มีการอินเตอร์รัพต์จากภายนอกเกิดขึ้น

1 = เกิดอินเตอร์รัพต์จากภายนอก

จะต้องเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

บิต 2 : TOIE แฟลกแสดงการเกิดการอินเตอร์รัพต์เนื่องจาก TMR0 เกิดโอเวอร์โฟลว์(TMR0 Overflow Interrupt Flag)

0 = TMR0 ไม่เกิดโอเวอร์โฟลว์

1 = TMR0 เกิดโอเวอร์โฟลว์

จะต้องเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

บิต 3 : RBIE เป็นบิตอีนาเบิลอินเตอร์รัพต์เนื่องจากการเปลี่ยนแปลงที่พอร์ต B (Port B Change Interrupt Enable)

0 = ดิสเอเบิล (ไม่ยอมให้เกิดการอินเตอร์รัพต์)

1 = อีนาเบิล (ยอมให้เกิดการอินเตอร์รัพต์)

บิต 4 : INTE เป็นบิตอีนาเบิลอินเตอร์รัพต์เนื่องจากสัญญาณภายนอก(External Interrupt Enable)

0 = ดิสเอเบิล (ไม่ยอมให้เกิดการอินเตอร์รัพต์)

1 = อีนาเบิล (ยอมให้เกิดการอินเตอร์รัพต์)

บิต 5 : TOIE เป็นบิตอีนาเบิลอินเตอร์รัพต์เนื่องจาก TMR0 เกิดโอเวอร์โฟลว์ (TMR0 Overflow Interrupt Enable)

0 = ดิสเอเบิล (ไม่ยอมให้เกิดการอินเตอร์รัพต์)

1 = อีนาเบิล (ยอมให้เกิดการอินเตอร์รัพต์)

บิต 6: EEIE บิตอีนาเบิลอินเตอร์รัพต์เนื่องจากการเขียนข้อมูลลงในหน่วยความจำอีอีพรอมเสร็จสมบูรณ์(EEPROM Write Complete Interrupt Enable)

0 = ดิสเอเบิล (ไม่ยอมให้เกิดการอินเตอร์รัพต์)

1 = อีนาเบิล (ยอมให้เกิดการอินเตอร์รัพต์)

บิต 7 : GIE เป็นบิตอีนาเบิลการอินเตอร์รัพต์ทุกเงื่อนไข(Global Interrupt Enable)

0 = ดิสเอเบิลการอินเตอร์รัพต์ทุกเงื่อนไข

1 = อีนาเบิลการอินเตอร์รัพต์ทุกเงื่อนไข

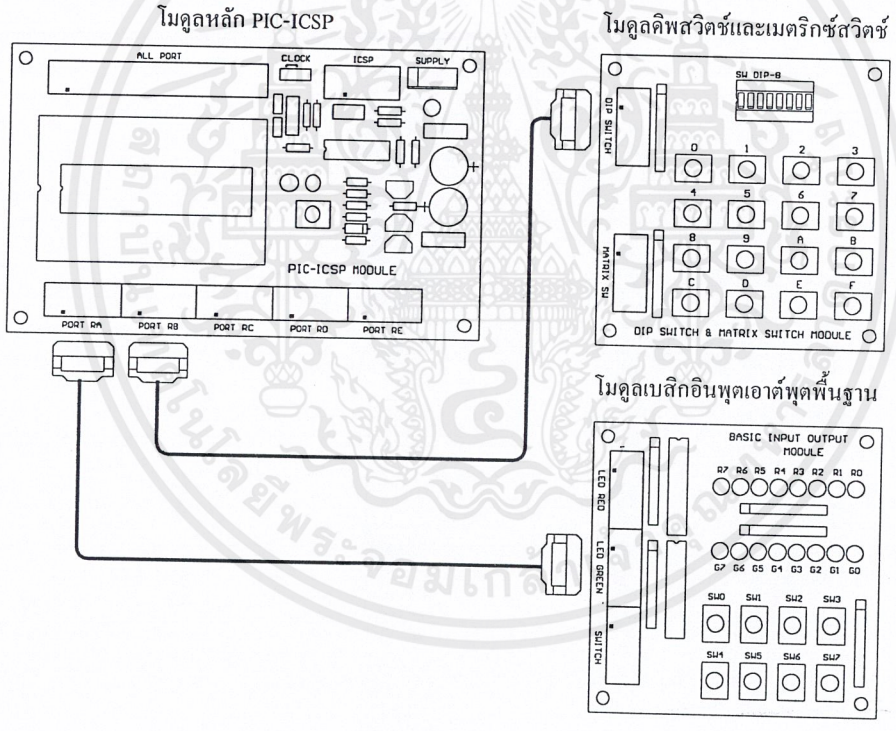
รูปที่ จ.9 ความหมายของบิตต่างๆ ในรีจิสเตอร์ INTCON

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การอีนาเบิลการตอบสนองการอินเตอร์รัพต์

การตอบสนองการอินเตอร์รัพต์สามารถทำได้ 2 ระดับคือ เลือกที่จะตอบสนองหรือไม่ตอบสนองการอินเตอร์รัพต์ในเงื่อนไขต่างๆ และเลือกที่จะตอบสนองหรือไม่ตอบสนองการบริการอินเตอร์รัพต์ในทุกรูปแบบโดยการกำหนดข้อมูลลงในรีจิสเตอร์ควบคุมการอินเตอร์รัพต์หรือ INTCON ซึ่งมีตำแหน่งอยู่ที่ 0x0B ดังรูปที่ จ.9

ถ้าเราต้องการให้ไมโครคอนโทรลเลอร์ PIC16F876 ตอบสนองการอินเตอร์รัพต์ต้องเซตบิตที่ GIE ในรีจิสเตอร์ INTCON ทุกครั้ง จากนั้นจึงมาเลือกว่าต้องการให้ PIC16F876 ตอบสนองการอินเตอร์รัพต์ในรูปแบบใด หากเราไม่เซตบิต GIE ก่อน โดยที่บิต GIE ยังเป็น “0” อยู่ จะทำให้ไมโครคอนโทรลเลอร์ PIC16F876 ไม่สามารถตอบสนองการอินเตอร์รัพต์ในรูปแบบต่างๆได้



รูปที่ จ.10 การเชื่อมต่อระหว่างโมดูล PIC-ICSP และโมดูลเบสิกอินพุตเอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   list p=16f877
   include "p16f877.inc"
__config _CP_OFF & _PWT_ON & _WDT_OFF & _XT_OSC
COUNT1      equ     0x0d
COUNT2      equ     0x0e
COUNT3      equ     0x0f
SAVE_W      equ     0x11
SAVE_S      equ     0x12
            org     0x00
            goto    Start
            org     0x04
            goto    Isr
;------------ Set port -------------
Start       bsf     STATUS,RP0      ; Bank 1
            clrf    PORTA           ; PortA = OUTPUT
            movlw   0xff            ; PortB = INPUT
            movwf   PORTB           ;
            bcf     STATUS,RP0      ; Bank 0
;-------- Enable Interupt ----------
            bcf     INTCON,RBIF
            bsf     INTCON,RBIE
            bsf     INTCON,GIE
;-------------- MAIN -------------
Show_ON     movlw   0xFF
            movwf   PORTA           ; LED OFF
            call    Delay
            clrf    PORTA
            call    Delay
            goto    Show_ON
;--- Interupt Service Routine-----
Isr         movwf   SAVE_W          ; Save W Register
            movf    STATUS,W
            movwf   SAVE_S
            clrf    PORTA
            btfss   PORTB,7
            bsf     PORTA,3
            btfss   PORTB,6
            bsf     PORTA,2
            btfss   PORTB,5
            bsf     PORTA,1
            btfss   PORTB,4
            bsf     PORTA,0
            movlw   0x10            ; Blink = 16
            movwf   COUNT3
Loop        call    Delay
            decfsz  COUNT3,F
            goto    Loop
            goto    End_Isr
End_Isr     movf    SAVE_S,W
            movwf   STATUS
            swapf   SAVE_W,F        ; Restore W
            swapf   SAVE_W,W
            movf    PORTB,W
            bcf     INTCON,RBIF     ; Clear Interupt Flag
            retfie
;--------Delay Subrutine --------;
Delay       movlw   0xaa
            movwf   COUNT1
Dec2        movlw   0xff
            movwf   COUNT2
Dec1        decfsz  COUNT2,F
            goto    Dec1
            decfsz  COUNT1,F
            goto    Dec2
            return
            end
```

รูปที่ จ.11 โปรแกรมการอินเตอร์รัพต์เนื่องจากแปลงลอจิกที่บิต 4 ถึง 7 ของพอร์ต B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    List p=16f876
    include "p16f876.inc"
  __config _WDT_OFF & _PWRTE_ON & _LVP_OFF & _XT_OSC
  COUNT1   equ      0x20
  COUNT2   equ      0x21
  COUNT3   equ      0x22
  COUNT4   equ      0x23
  SAVE_W   equ      0x24
  SAVE_S   equ      0x25
  GIE      equ      7
  INTE     equ      4
  INTF     equ      1

           org      0x00
           goto     Start
           org      0x004
           goto     Inter

  Start    bsf      STATUS,RP0        ; Bank 1
           movlw    0x06
           movwf    ADCON1
           clrf     TRISA             ; PortA = OUTPUT
           movlw    b'00111111'
           movwf    OPTION_
           bcf      STATUS,RP0        ; Bank 0
           clrf     PORTB
  ;------- Enable Interupt --------;
           bcf      INTCON,INTF
           bsf      INTCON,INTE
           bsf      INTCON,GIE
  ;----------- MAIN ---------------;
           clrf     PORTA             ; LED OFF
  Disp_ON  call     Delay
           bsf      PORTA,0           ; RA0 ON
  Left     call     Delay
           bcf      STATUS,C
           rlf      PORTA,F
           btfss    PORTA,3
           goto     Left
  Right    call     Delay
           bcf      STATUS,C
           rrf      PORTA,F
           btfss    PORTA,0
           goto     Right
           goto     Disp_ON

  ;--------- Interupt -------------;
  Inter    bcf      INTCON,INTF
           movwf    SAVE_W
           movf     STATUS,W
           movwf    SAVE_S
           movlw    0x10
           movwf    COUNT4
  Blink    movlw    0x0f
           movwf    PORTA
           call     Delay
           clrf     PORTA
           call     Delay
           decfsz   COUNT4,F
           goto     Blink
           bsf      PORTA,2
  End_Int  movf     SAVE_S,W
           movwf    STATUS
           swapf    SAVE_W,F
           swapf    SAVE_W,W
           retfie
  ;----------Delay Time -----------;
  Delay    movlw    0xaa
           movwf    COUNT1
  Dec1     movlw    0xff
           movwf    COUNT2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Dec2    decfsz   COUNT2,F
        goto     Dec2
        decfsz   COUNT1,F
        goto     Dec1
        return

        end
```

รูปที่ จ.12 การอินเตอร์รัพต์จากการป้อนสัญญาณภายนอกเข้าที่ขา RB0/INT

ลำดับขั้นตอนการทดลอง

1. ต่อสายเชื่อมต่อระหว่างโมดูลตามรูปที่ จ.10
2. เขียนโปรแกรมรูปที่ จ.11 ทำการแอสเซมเบลอร์
3. จากนั้นเขียนข้อมูลลงบน PIC16F876 ทำการรันโปรแกรมบน PIC16F876 โดยการกดสวิตช์รีเซต ผลที่ได้เป็นดังนี้

.....

.....

4. เลื่อนดิพสวิตช์ D0 ขึ้นด้านบนเพื่อป้อนสัญญาณ “0” เข้าที่ขา RB7 ผลที่เกิดขึ้นเป็นดังนี้

.....

.....

5. แก้ไขโปรแกรมดังรูปที่ จ.11 จาก bcf INTCON, RBIF เป็น nop ผลที่เกิดขึ้นเป็นดังนี้

.....

.....

6. จากข้อ 4 เปลี่ยนจากการป้อนสัญญาณ “0” เข้าที่ขา RB7 เป็น RB4 แทน สังเกตผลที่ได้เมื่อเทียบกับข้อ 4

.....

.....

7. จากข้อ 6 เปลี่ยนขาจาก RB4 เป็นขา RB5 สังเกตผลที่ได้เมื่อเทียบกับข้อ 6

.....

.....

8. จากข้อ 6 เปลี่ยนขาจาก RB4 เป็นขา RB6 สังเกตผลที่ได้เมื่อเทียบกับข้อ 4

.....

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. ต่อสายเชื่อมต่อระหว่างโมดูลตามรูปที่ จ.10

10. เขียนโปรแกรมดังรูปที่ จ.12 ทำการแอสเซมเบลอร์

11. จากนั้นเขียนข้อมูลลงบน PIC16F876 ทำการรันโปรแกรมบน PIC16F876 โดยการคลิกปุ่ม RUN ผลที่ได้เป็นดังนี้

.....

.....

12. เปลี่ยนรูปแบบการอินเตอร์รัพต์จากขอบขาขึ้นเป็นขอบขาลง โดยการกำหนดบิต 6 ของรีจิสเตอร์ OPTION เป็น "1" แล้วทำการทดลองอีกครั้ง

.....

.....

สรุปผลการทดลอง

.....

.....

.....

.....

.....

.....

.....

.....

คำถามท้ายการทดลอง

1. บิตใดในรีจิสเตอร์ INTCON ที่อีนาเบิลการเกิดอินเตอร์รัพต์เนื่องจากบิต 4 ถึงบิต 7 ของพอร์ต B เกิดการเปลี่ยนแปลงระดับลอจิก

2. การเลือกขอบขาของสัญญาณอินเตอร์รัพต์จากภายนอกกระทำได้อย่างไร

3. บิต GIE มีความสำคัญอย่างไรต่อการอินเตอร์รัพต์

4. จงเขียนโปรแกรมแสดงการอินเตอร์รัพต์เนื่องจากการเปลี่ยนแปลงลอจิกที่บิต 4 ถึง 7 ของพอร์ต B โดยแสดงไฟวิ่งจากซ้ายไปขวาเมื่อไม่เกิดการอินเตอร์รัพต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 4
การใช้งานไทเมอร์เคาน์เตอร์ภายใน PIC16F876
(Timer/Counter)

จุดประสงค์

1. เพื่อให้สามารถใช้งานไทเมอร์เคาน์เตอร์ใน PIC16F876 ได้
2. เพื่อให้สามารถใช้งานวอตช์ด็อกไทเมอร์ร่วมกับ TMR0 ได้
3. เพื่อให้สามารถเขียนโปรแกรมเพื่อใช้งานไทเมอร์เคาน์เตอร์ได้

อุปกรณ์ที่ใช้ในการทดลอง

1. โมดูลหลัก PIC-ICSP
2. โมดูลเบสิกอินพุตเอาต์พุต
3. คอมพิวเตอร์พีซีที่ติดตั้งโปรแกรม MPLAB
4. สายเชื่อมต่อระหว่างโมดูลหลัก PIC-ICSP กับคอมพิวเตอร์พีซี
5. สายเชื่อมต่อพอร์ตระหว่างโมดูล

ทฤษฎีเบื้องต้น

ไมโครคอนโทรลเลอร์ PIC16F876 มีไทเมอร์เคาน์เตอร์ 3 ตัว คือ TIMER0 ขนาด 8 บิต, TIMER1 ขนาด 16 บิต และ TIMER2 ขนาด 8 บิต ซึ่งภายในไทเมอร์เคาน์เตอร์จะประกอบไปด้วย รีจิสเตอร์ TMR0, TMR1, TMR2, ปรีสเกลเลอร์ (Prescaler) ขนาด 8 บิต และตัวนับหรือเคาน์เตอร์ โดยไทเมอร์เคาน์เตอร์ภายใน PIC16F876 สามารถทำงานได้ทั้งจากสัญญาณนาฬิกาภายในและภายนอก ทั้งยังสามารถเลือกให้ทำงานที่ขอบขาของสัญญาณทั้งขอบขาลงและขอบขาขึ้น

ภายในไทเมอร์เคาน์เตอร์จะมีการนับเฉพาะขอบขาขึ้นเท่านั้น เมื่อค่าจากการนับเกิดโอเวอร์โฟลว์ (Overflow) คือค่าจากการนับเปลี่ยนจาก 0xff เป็น 0x00 จะทำให้เกิดการอินเตอร์รัพต์ขึ้น แต่ต้องขึ้นอยู่กับว่ามีการอีนาเบิลการอินเตอร์รัพต์แบบนี้ที่บิต TOIE ในรีจิสเตอร์ INCON หรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7	6	5	4	3	2	1	0
RBPU	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0

บิต 7 : อีนาเบิลการพูลอัปที่พอร์ต B

0 = อีนาเบิล

1 = ดิสเอเบิล

บิต 6 : เลือกขอบขาของสัญญาณอินเตอร์รัพต์

0 = ขอบขาลง

1 = ขอบขาขึ้น

บิต 5 : เลือกแหล่งกำเนิดของสัญญาณนาฬิกาให้แก่ TMR0

0 = จากภายใน

1 = จากภายนอกผ่านทางขา TOCK1

บิต 4 : เลือกขอบขาของสัญญาณจากภายนอกสำหรับ TMR0

0 = ขอบขาขึ้น

1 = ขอบขาลง

บิต 3 : เลือกการทำงานของปรีสเกลเลอร์

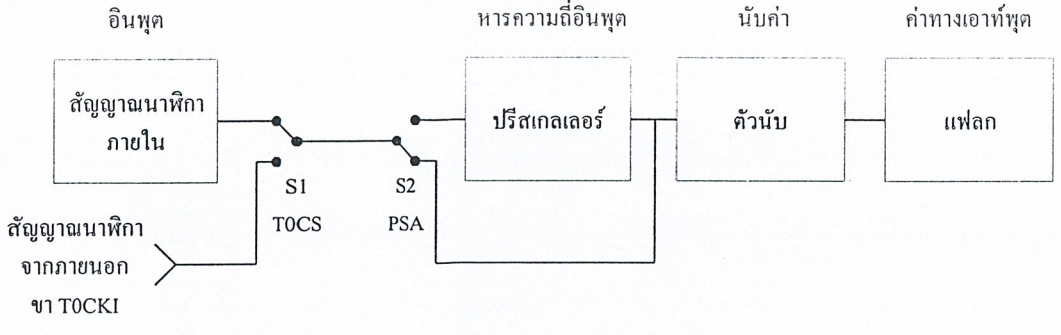
0 = ทำงานกับ TMR0

1 = ทำงานกับ WDT

บิต 2-0 : กำหนดค่าของปรีสเกลเลอร์

PS2	PS1	PS0	ตัวหาร
0	0	0	2
0	0	1	4
0	1	0	8
0	1	1	16
1	0	0	32
1	0	1	64
1	1	0	128
1	1	1	256

รูปที่ จ.13 ความหมายของบิตแต่ละบิตในรีจีสเตอร์ OPTION



รูปที่ จ.14 แผนผังการทำงานของไทเมอร์เคาน์เตอร์ใน PIC16F876

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณนาฬิกาที่ใช้กระตุ้นให้ไทเมอร์เคาน์เตอร์ทำงานมาจาก 2 แหล่งคือ จากวงจรกำเนิดสัญญาณนาฬิกาภายในตัวไมโครคอนโทรลเลอร์เอง และวงจรกำเนิดสัญญาณนาฬิกาจากภายนอก ดังรูปที่ จ.14 เมื่อสวิตช์ S1 ใช้เลือกแหล่งกำเนิดสัญญาณนาฬิกาก็คือ บิต TOCS ในรีจิสเตอร์ OPTION ส่วน S2 ใช้เลือกว่าต้องการให้สัญญาณผ่านการปรีสเกลเลอร์หรือไม่คือ บิต PSA นั่นเอง

การประยุกต์ใช้งาน

การใช้ TMR0 สร้างโปรแกรมหน่วงเวลาเพื่อสร้างสัญญาณสี่เหลี่ยม

การกำเนิดสัญญาณนาฬิกาสี่เหลี่ยมความถี่ 1 เมกะเฮิรตต์ สัญญาณลอจิกสูงจะปรากฏที่เอาต์พุตนานเพียงใดนั้นขึ้นอยู่กับการหน่วงเวลาให้เกิดสัญญาณลอจิกสูงค้างที่เอาต์พุต ซึ่งในที่นี้จะใช้ไทเมอร์เคาน์เตอร์มาทำหน้าที่นี้ จะเริ่มด้วยการเคลียร์ค่าวอตช์ด็อกไทเมอร์ จากนั้นเขียนข้อมูลลงในรีจิสเตอร์ OPTION เพื่อกำหนดให้ปรีสเกลเลอร์ทำงานร่วมกับ TMR0 และเลือกที่จะใช้สัญญาณนาฬิกาจากแหล่งกำเนิดสัญญาณนาฬิกาภายใน PIC16F876 เอง

กำหนดให้ปรีสเกลเลอร์มีอัตราการลดทอนความถี่เท่ากับ 128 นั่นคือ สัญญาณนาฬิกาภายในจะถูกหาร 128 ก่อนส่งเข้าสู่ตัวนับในไทเมอร์เคาน์เตอร์ภายใน PIC16F876 ดังนั้นสัญญาณเอาต์พุตที่ได้จะสามารถคำนวณหาค่าความถี่ได้ดังนี้

ความถี่ = 1 / (ค่าของปรีสเกลเลอร์ × ค่าของ TMR0 × คาบเวลาของสัญญาณนาฬิกาภายใน × 2)

ถ้าค่าปรีสเกลเลอร์= 128, TMR0=4 และสัญญาณนาฬิกามีคาบเวลา 1 ไมโครวินาที ความถี่ที่ได้มีค่า 1024 เมกะเฮิรตต์

ไทเมอร์เคาน์เตอร์กับการอินเตอร์รัพต์

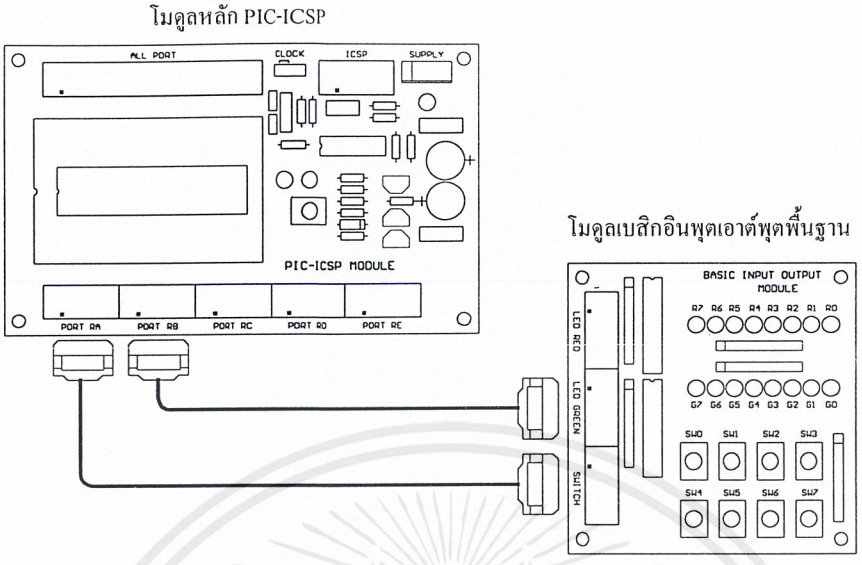
การกำเนิดสัญญาณนาฬิการูปสี่เหลี่ยมโดยใช้การอินเตอร์รัพต์ที่เกิดขึ้นในไทเมอร์เคาน์เตอร์ เมื่อ TMR0 เกิดโอเวอร์โฟลว์ ทำให้เกิดสัญญาณนาฬิการูปสี่เหลี่ยมขึ้น

ไทเมอร์เคาน์เตอร์กับสัญญาณนาฬิกาภายนอก

การกำหนดให้ไทเมอร์เคาน์เตอร์นับสัญญาณนาฬิกาจากภายนอกนั้น เริ่มด้วยการกำหนดให้บิต 5 ของรีจิสเตอร์ OPTION เป็น “1” และเพื่อให้การนับถูกต้องจะต้องกำหนดขอบขาของสัญญาณโดยสามารถกำหนดได้ที่บิต 4 ของรีจิสเตอร์ OPTION

การกำหนดให้ไทเมอร์เคาน์เตอร์นับค่าของสัญญาณนาฬิกาภายนอก โดยกำหนดความถี่ของสัญญาณนาฬิกาภายนอกเท่ากับ 1 เมกะเฮิรตต์ เมื่อมีสัญญาณนาฬิกาเข้ามาครบ 8 ลูก LED ที่ต่ออยู่กับพอร์ต B จะเลื่อนไป 1บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ จ.15 การเชื่อมต่อระหว่างโมดูลหลัก PIC-ICSP และโมดูลเบสิกอินพุตเอาต์พุต

```
   list p=16f876
   include "p16f876.inc"
__config _PWRTE_ON & _WDT_OFF & _LVP_OFF & _XT_OSC
CountDly    equ     0x20

            org     0x00
Initial     bsf     STATUS,RP0
            movlw   b'11010100'     ; prescale หาร 32
            movwf   OPTION_REG
            clrf    TRISB
            bcf     STATUS,RP0
            clrf    PORTB

Start       movlw   0xff
            xorwf   PORTB,F
            call    Delay1S
            goto    Start

Delay1S     movlw   .125            ; for 125 loop
            movwf   CountDly
Del0        clrf    TMR0
Del1        movlw   .125            ; for Timer0
            subwf   TMR0,W
            btfss   STATUS,Z
            goto    Del1
            decfsz  CountDly,F
            goto    Del0
            return
            end
```

รูปที่ จ.16 การใช้ TMR0 ไม่มีอินเตอร์รัพต์ ความถี่ 1 เฮิรตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  list p=16f876
  include "p16f876.inc"
__config _PWRTE_ON & _WDT_OFF & _XT_OSC
CountDly    equ     0x20

            org     0x00
            goto    Initial
            org     .4
            goto    IntRout
Initial     bsf     STATUS,RP0
            movlw   b'11010101'     ; หาร 64
            movwf   OPTION_REG
            clrf    TRISB
            bcf     STATUS,RP0
            clrf    CountDly

Main        movlw   b'10100000'
            movwf   INTCON
            movlw   0xff
            movwf   PORTB
            bsf     INTCON,TOIF
            goto    $

IntRout     bcf     INTCON,TOIF
            movlw   -.125           ; -.125 = 256-125
            movwf   TMR0
            incf    CountDly,F
            movlw   .125
            subwf   CountDly,W
            btfss   STATUS,Z
            retfie
            clrf    CountDly
            comf    PORTB,F
            retfie
            end
```

รูปที่ จ.17 การใช้ TMR0 มีอินเตอร์รัพต์ ความถี่ 0.5 เฮิรตซ์

```
  list p=16f876
  include "p16f876.inc"
__config _PWRTE_ON & _WDT_OFF & _LVP_OFF & _XT_OSC
CountDly    equ     0x20
            org     0x000
            goto    Initial
            org     0x04
            goto    IntServ
Initial     bsf     STATUS,RP0
            clrwdt
            movlw   b'10100001'             ; หาร 4, ขอบขาลง
            movwf   OPTION_REG
            movlw   0x06
            movwf   ADCON1
            movlw   0xff
            movwf   TRISA
            clrf    TRISB
            bcf     STATUS,RP0

Main        clrf    PORTB
            bsf     PORTB,0
            movlw   b'10100000'
            movwf   INTCON
            movlw   0xff
            movwf   TMR0                    ; TMR0 = -1
            bcf     STATUS,C
            goto    $
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
IntServ     bcf     INTCON,T0IF
            movlw   0xff
            movwf   TMR0
            rlf     PORTB,F
            retfie
            end
```

รูปที่ จ.18 การใช้ไทเมอร์เคาน์เตอร์กับสัญญาณนาฬิกาภายนอก

ลำดับขั้นตอนการทดลอง

1. เขียนโปรแกรมดังรูปที่ จ.16 ทำการแอสเซมเบลอร์
2. ต่อสายเชื่อมต่อระหว่างโมดูลตามรูปที่ จ.15
3. เขียนข้อมูลลงบน PIC16F876 ทำการรันโปรแกรมบน PIC16F876 โดยการคลิกปุ่ม RUN ผลที่ได้เป็นดังนี้

.....

.....

4. จากโปรแกรมดังรูปที่ จ.16 ถ้าเราเปลี่ยนจาก movlw b'11010100' เป็น movlw b'11010110' ผลที่ได้เป็นดังนี้

.....

.....

5. เขียนโปรแกรมดังรูปที่ จ.17 ทำการแอสเซมเบลอร์
6. ต่อสายเชื่อมต่อระหว่างโมดูลตามรูปที่ จ.10
7. เขียนข้อมูลลงบน PIC16F876 ทำการรันโปรแกรมบน PIC16F876 โดยการคลิกปุ่ม RUN ผลที่ได้เป็นดังนี้

.....

.....

8. จากโปรแกรมดังรูปที่ จ.17 ถ้าเราเปลี่ยนจาก movlw b'11010101' เป็น movlw b'11010010' ผลที่ได้เป็นดังนี้

.....

.....

9. ถ้าต้องการสร้างสัญญาณที่ความถี่ 1 เมกะเฮิรตซ์ จะต้องกำหนด

ค่าของปรีสเกลเลอร์ = ...

TMR0 = ...

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. เขียนโปรแกรมดังรูปที่ จ.18 ทำการแอสเซมเบลอร์

11. ต่อสายเชื่อมต่อระหว่างโมดูลตามรูปที่ จ.15

12. เขียนข้อมูลลงบน PIC16F876 ทำการรันโปรแกรมบน PIC16F876 โดยการคลิกปุ่ม RUN ผลที่ได้เป็นดังนี้

.....

.....

13. สังเกตการทำงานของ LED

เมื่อกดสวิตช์ที่พอร์ต RA4/TOCKI = ...ครั้ง LED จึงจะเลื่อนบิตไป1 บิต

14. ถ้าต้องการเปลี่ยนค่าของปรีสเกลเลอร์เป็น 8 สังเกตการทำงานของ LED เป็นดังนี้

.....

.....

สรุปผลการทดลอง

.....

.....

.....

.....

.....

.....

.....

.....

คำถามท้ายการทดลอง

1. ในไมโครคอนโทรลเลอร์ PIC16F876 มีไทเมอร์เคาน์เตอร์กี่ตัว ขนาดกี่บิต อะไรบ้าง

2. จงอธิบายการทำงานของโปรแกรมรูปที่ จ.16-จ.18

3. จงอธิบายการนำไทเมอร์เคาน์เตอร์ไปใช้งานได้อย่างไรบ้างและทำอย่างไร

4. จงอธิบายความหมายของปรีสเกลเลอร์พอเข้าใจ

5. จงเขียนโปรแกรมสร้างสัญญาณที่ความถี่ 100 เฮิรตซ์ โดยใช้ไทเมอร์เคาน์เตอร์กับสัญญาณนาฬิกาภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 5

การเชื่อมต่อ PIC16F876 กับจอแสดงผลแบบผลึกเหลว (LCD)

จุดประสงค์

1. เพื่อให้สามารถใช้งาน LCD แบบตัวอักษรหรือแบบอักขระได้
2. เพื่อให้สามารถเขียนโปรแกรมเพื่อควบคุมให้ PIC16F876 แสดงผลออกทาง LCD แบบตัวอักษรหรือแบบอักขระได้
3. เพื่อให้สามารถใช้งาน LCD ในโหมด 4 บิตและ 8 บิตได้

อุปกรณ์ในการทดลอง

1. โมดูลหลัก PIC-ICSP
2. โมดูลแสดงผลแบบผลึกเหลว (LCD)
3. คอมพิวเตอร์พีซีที่ติดตั้งโปรแกรม MPLAB
4. สายเชื่อมต่อระหว่างโมดูลหลัก PIC-ICSP กับคอมพิวเตอร์พีซี
5. สายเชื่อมต่อพอร์ตระหว่างโมดูล

ทฤษฎีเบื้องต้น

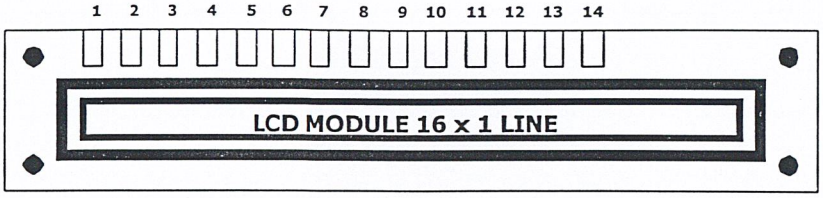
จอแสดงผลแบบผลึกเหลวหรือ LCD (Liquid Crystal Display) แบ่งออกเป็น 3 แบบตามลักษณะการแสดงผลคือ LCD แบบอักขระ (Character LCD Module), LCD แบบกราฟฟิก (Graphic LCD Module) และ LCD แบบเซกเมนต์ (Segment LCD Module)

LCD แบบอักขระเป็น LCD ที่สามารถแสดงตัวอักษร ตัวเลขและเครื่องหมายต่างๆได้โดยสร้างจากจุดเล็กๆ ทางแนวตั้งและแนวนอนหรือ dot matrix โดยทั่วไปจะมี 2 ขนาดคือ 5x7 จุดและ 5x10 จุด

LCD แบบกราฟฟิกมีลักษณะโครงสร้างคล้ายกับ LCD แบบอักขระ สามารถแสดงข้อมูลเป็นทั้งแบบตัวอักขระ ตัวเลข เครื่องหมายและรูปภาพได้ ความละเอียดของภาพก็จะขึ้นอยู่กับความละเอียดของ dot matrix ของ LCD ตัวนั้นๆของขนาด LCD แบบนี้มีอยู่หลายขนาดให้เลือกใช้

LCD แบบเซกเมนต์ (Segment) เป็นแบบเล็กสุดมีลักษณะการแสดงผลเป็นเซกเมนต์คล้ายกับ LED ตัวเลข 7 ส่วนโดยปกติมักจะมีมากกว่า 1 หลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ จ.19 รูปร่างและการจัดขาของจอแสดงผลแบบผลึกเหลวแบบอักขระ

LCD ขนาด 24 ตัวอักษรจำนวน 2 บรรทัด (LCD 24x2)

LCD ที่เราใช้ในการทดลองนี้เป็น LCD อีกตัวที่มีโครงสร้างเป็นมาตรฐานมีขาที่ต่อใช้งาน 14 ขา มีการจัดขาดังรูปที่ จ.19 รายละเอียดของแต่ละขามีดังนี้

Vss (ขา 1) = ต่อกราวด์

Vcc (ขา 2) = ต่อไฟเลี้ยง 5 โวลต์

Vo (ขา 3) = เป็นขาอินพุต รับแรงดันเพื่อรับความเข้มของการแสดงผล

RS (ขา 4) = เป็นขาอินพุต ใช้ในการรับชนิดของข้อมูล ถ้าขานี้เป็น “0” ข้อมูลที่ส่งมาจะเป็นคำสั่งแต่ถ้าเป็น “1” ข้อมูลนี้จะเป็นข้อมูลสำหรับแสดงผล

R/W (ขา 5) = เป็นขาที่ใช้เลือกการอ่านหรือเขียนข้อมูลกับ LCD ถ้าเป็น “0” เป็นการกำหนดให้เขียนข้อมูลแต่ถ้าเป็น “1” จะเป็นการอ่านข้อมูล

E (ขา 6) = เป็นอินาเบิล LCD ให้ทำงาน

D0-D7 (ขา 7-14) = เป็นขาที่ใช้เป็นทางผ่านของข้อมูลระหว่าง LCD กับอุปกรณ์

ขา R/W, RS และขา E จะใช้งานร่วมกันดังตารางการทำงานที่ 10.1

ตารางที่ จ.3 ความสัมพันธ์ของการทำงานร่วมกันของขา RS, R/W และ E

RS	R/W	E	การทำงาน
0	0	↓	เขียนคำสั่ง
0	1	⊓	อ่านสถานะของ LCD
1	0	↓	เขียนข้อมูล
1	1	⊓	อ่านข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งควบคุม LCD

การเขียนคำสั่งลงในตัวควบคุม จะต้องกำหนดให้ RS, R/W และขา E จะต้องเป็น “0” แล้วเขียนคำสั่งตามไป คำสั่งในการควบคุมมี 10 คำสั่งดังนี้

1. คำสั่งเคลียร์ตัวแสดงผล (Clear display) มีข้อมูลเป็นคำสั่ง 0x01 เป็นคำสั่งที่ใช้ในการเขียนข้อมูลช่องว่างหรือ space เข้าไปใน DDRAM ทั้งหมด เมื่อตัวควบคุมเอ็กซิคิ้วต์คำสั่งนี้จะทำการกำหนดแอดเดรสของ DDRAM เป็น 0 เคอร์เซอร์จะกลับไปอยู่ที่ ตำแหน่งซ้ายมือสุดของจอแสดงผล แล้วเซตบิต I/D ให้เป็น “1”

2. คำสั่ง Return Home ต้องกำหนดให้บิต “1” ของข้อมูลเป็น “1” เป็นคำสั่งให้เคอร์เซอร์เคลื่อนที่กลับไปยังตำแหน่งซ้ายสุดของจอแสดงผล แต่ข้อมูลบนตัวแสดงผลไม่เปลี่ยนแปลงนั่นคือข้อมูลของคำสั่งนี้จะเป็น 0x02 หรือ 0x03 ก็ได้

3. คำสั่งเลือกโหมดการป้อนกลับ (Entry mode set) มีรายละเอียดดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
0	0	0	0	0	1	I/D	S

รูปที่ จ.20 คำสั่งเลือกโหมดการป้อนกลับ

บิต S เป็นบิตที่ใช้ในการกำหนดลักษณะของการแสดงผลเมื่อมีการป้อนข้อมูล ถ้าหากบิต S เป็น “1” เมื่อเกิดข้อมูลใหม่บนจอแสดงผล ตัวเคอร์เซอร์จะอยู่กับที่แต่ตัวอักษรข้อมูลเดิมจะถูกดันไปทางซ้าย แต่ถ้าหากบิตนี้เป็น “0” เมื่อเกิดข้อมูลใหม่ตัวเคอร์เซอร์จะเลื่อนไปทางขวามือ

บิต I/D เป็นบิตที่ใช้ในการกำหนดว่าเมื่อเขียนหรืออ่านข้อมูลแล้ว ทำให้แอดเดรสของ DDRAM เพิ่มขึ้นหรือลดลงหนึ่งแอดเดรสโดยถ้าบิตนี้เป็น “1” แอดเดรสของ DDRAM จะเพิ่มขึ้น แต่ถ้าเป็น “0” แอเดรสจะลดลง

4. คำสั่งควบคุมการแสดงผล มีรายละเอียดดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
0	0	0	0	1	D	C	B

รูปที่ จ.21 คำสั่งควบคุมการแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต D ใช้ควบคุมการเปิดปิดจอแสดงผล ถ้าเป็น “0” จะเป็นการปิดจอแสดงผล

บิต C ใช้ควบคุมการแสดงตัวเคอร์เซอร์บนจอแสดงผล ถ้าต้องการให้มีเคอร์เซอร์แสดงผล ต้องกำหนดให้บิตนี้เป็น “1” ถ้ากำหนดให้เป็น “0” จะเป็นการเปิดเคอร์เซอร์หรือไม่แสดงเคอร์เซอร์

บิต B ใช้ควบคุมการกระพริบของเคอร์เซอร์ถ้าบิตเป็น “1” เคอร์เซอร์จะกระพริบ

5. คำสั่งควบคุมการเลื่อนเคอร์เซอร์และข้อมูลตัวอักษร มีรายละเอียดดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
0	**0**	**0**	**1**	**S/C**	**R/L**	*	*

รูปที่ จ.22 การควบคุมการเลื่อนเคอร์เซอร์และข้อมูลตัวอักษร

การควบคุมการเลื่อนเคอร์เซอร์และตัวอักษรบนจอแสดงผล ขึ้นอยู่กับการกำหนดบิต S/C และ R/L สรุปได้ดังนี้

ตารางที่ จ.4 การควบคุมการเลื่อนเคอร์เซอร์บนจอแสดงผล

S/C	R/L	ลักษณะการเลื่อนเคอร์เซอร์	ข้อมูลคำสั่ง
0	0	ไปทางซ้าย	0x10-0x13
0	1	ไปทางขวา	0x14-0x17
1	0	ตัวอักษรใหม่ไปทางซ้าย	0x18-0x1B
1	1	ตัวอักษรใหม่ไปทางขวา	0x1C-0xF

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
0	**0**	**1**	**DL**	**N**	**F**	*	*

รูปที่ จ.23 การกำหนดฟังก์ชั่นการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งกำหนดฟังก์ชั่นการทำงาน ดังรูปที่ จ.23 มีรายละเอียดดังนี้

- บิต DL ใช้กำหนดจำนวนบิตที่ใช้ติดต่อส่งผ่านข้อมูลถ้าบิตนี้เป็น “0” จะเป็นการติดต่อแบบ 4 บิต แต่ถ้าเป็น “1” จะเป็นการติดต่อแบบ 8 บิต

- บิต N ใช้กำหนดจำนวนบรรทัดของการแสดงผล ถ้าเป็น “0” จะแสดงผล 1 บรรทัด ถ้าเป็น “1” จะแสดง 2 บรรทัด หรือในกรณีที่จอแสดงผลสามารถแสดงได้มากกว่า 2 บรรทัด

- บิต F ใช้เลือกความละเอียดของตัวอักษรในการแสดงผล ถ้าบิตนี้เป็น “0” จะเป็นการแสดงผลแบบ 5x7 จุด และถ้าเป็น “1” จะแสดงผลเป็นแบบ 5x10 จุด

6. **คำสั่งเลือกแอดเดรสของ CGRAM** เมื่อต้องการกำหนดแอดเดรสของ CGRAM ต้องกำหนดให้บิต 7 เป็น “0” บิต 6 เป็น “1” ส่วนอีก 6 บิตที่เหลือจะแทนด้วยค่าแอดเดรสของ CGRAM จะต้องทำการกำหนดแอดเดรสด้วยคำสั่งนี้ก่อนที่จะเขียนหรืออ่านข้อมูลให้ CGRAM โดยแอดเดรสจะอยู่ระหว่าง 0x00-0x3F

7. **คำสั่งเลือกแอดเดรสของ DDRAM** ก่อนที่จะทำการอ่านหรือเขียนข้อมูลโดยบิต 7 ต้องเป็น “1” และข้อมูลบิต 7 ที่เหลือ เป็นค่าแอดเดรสของ DDRAM ซึ่งแอดเดรสจะอยู่ระหว่าง 0x8C-0xFF นอกจากนี้จำนวนแอดเดรสยังขึ้นอยู่กับการกำหนดสถานะที่บิต N หากบิต N เป็น “0” แอดเดรสจะอยู่ระหว่าง 0x80-0xCF และถ้าบิต N เป็น “1” แอดเดรสจะมี 2 ช่วงคือ 0x8C-0x87 และ 0xC0-0xC7

8. **คำสั่งอ่าน Busy และแอดเดรส** มีรายละเอียดดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
BF	**A**	**A**	**A**	**A**	**A**	**A**	**A**

รูปที่ จ.24 การอ่าน Busy และแอดเดรส

จะเป็นแฟลกบอกสถานะของตัวควบคุม LCD ว่าพร้อมจะรับข้อมูลอยู่หรือไม่ ถ้าหากบิต BF เป็น “0” แสดงว่าตัวควบคุม LCD พร้อมจะรับข้อมูลหรือคำสั่ง แต่ถ้าเป็น “1” แสดงว่าตัวควบคุม LCD ยังอยู่ในกระบวนการทำงานภายในหรือกำลังประมวลผลข้อมูลอยู่ไม่พร้อมที่จะรับข้อมูลหรือคำสั่ง

เมื่อต้องการอ่านแฟลกต้องกำหนดให้ขา R/W เป็น “1” ด้วย แต่สัญญาณที่ RS ยังต้องเป็น “0” อยู่เพราะข้อมูลนี้ยังเป็นข้อมูลคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จังหวะการทำงานของ LCD

การใช้งาน LCD ต้องเขียนโปรแกรมเพื่อหน่วงเวลาให้ LCD พร้อมที่จะทำงาน โดยเมื่อเริ่มจ่ายแรงดันให้กับ LCD ต้องรอ 5 มิลลิวินาที เพื่อให้ LCD เตรียมความพร้อมหรือ Initial หลังจากนั้นจะต้องกำหนดลอจิกขา RS ของ LCD แล้วต้องหน่วงเป็นเวลาประมาณ 125 ไมโครวินาที เพื่อให้ไมโครคอนโทรลเลอร์ใน LCD แปลความหมายของลอจิกที่ขา RS ว่าข้อมูลต่อไปที่ได้รับเป็นรหัสคำสั่งหรือข้อมูลที่ต้องการแสดงผล จากนั้นจะเป็นการส่งข้อมูลมารอที่บัสข้อมูล D0-D7 (กรณีทำงานในโหมด 8 บิต) ต่อไปจะเป็นการส่งสัญญาณพัลส์ไปที่ขา E เพื่ออินาเบิล LCD ให้รับข้อมูลจากบัส (Bus) ข้อมูลเข้าไปโดยพัลส์ที่ป้อนเข้าที่ขา E ของ LCD ต้องเป็นพัลส์ขอบขาขึ้นจากนั้นทำการหน่วงเวลา 125 ไมโครวินาที

```
Delay5mS    movlw    0x29          ; 40
            movwf    COUNT2
Delay       call     Dealay125uS
            decfsz   COUNT2,F      ; 40 times=5mS
            goto     Delay
            return
```

รูปที่ จ.25 ตัวอย่างโปรแกรมหน่วงเวลา 5 มิลลิวินาที

```
Pulse       bsf      PORTA,E       ; pulse E line
            Nop
            bcf      PORTA,E
            call     Delay125uS    ; Delay 125 uS
            return
```

รูปที่ จ.26 ตัวอย่างโปรแกรมการส่งสัญญาณพัลส์

```
Delay125mS  movlw    0x2a          ; 42x3 cycle
            movwf    COUNT1
Delay       decfsz   COUNT1,F      ; 40 times=5mS
            goto     Repeat        ; not 0
            return
```

รูปที่ จ.27 ตัวอย่างโปรแกรมหน่วงเวลา 125 ไมโครวินาที

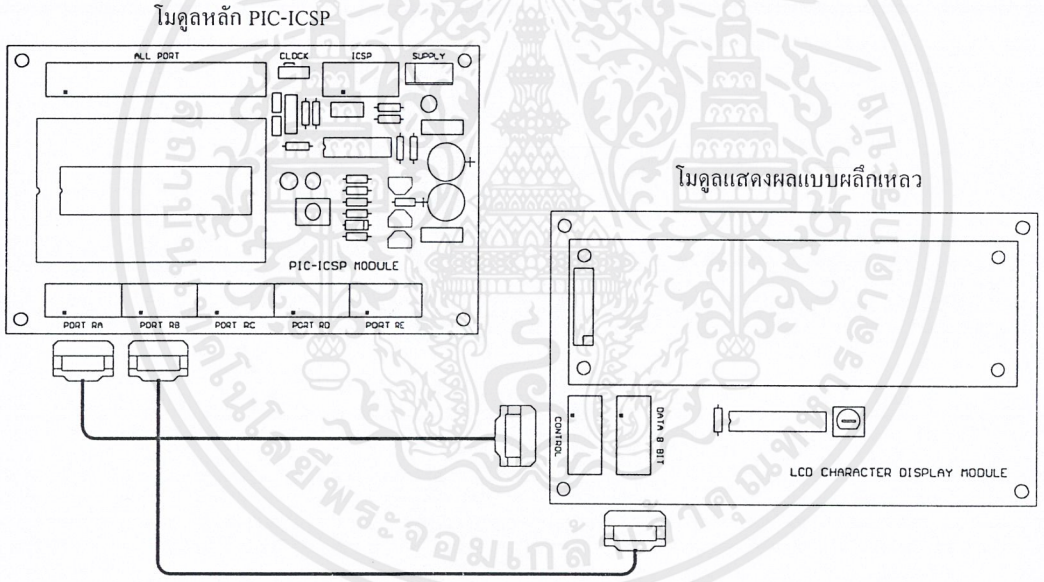
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การต่อ PIC 16F876 กับ LCD

การต่อ PIC 16F876 กับ LCD แบบอักขระ 8 บิต เราจะกำหนดให้พอร์ต A เป็นขาของการควบคุม โดย RA1 ต่ออยู่กับขา E, RA2 ต่ออยู่กับขา RS และ RA3 ต่ออยู่ กับ R/W ส่วนพอร์ต B จะต่ออยู่กับขาข้อมูล D0-D7 ทั้ง 8 เส้น

การใช้งาน LCD แบบ 4 บิต

การต่อขาใช้งาน LCD แบบ 4 บิตนั้นที่ขาข้อมูลเราจะใช้เพียง 4 บิตบนเท่านั้น ส่วนการโปรแกรม ถ้าทำงานในแบบ 8 บิตเมื่อป้อนข้อมูลให้ LCD แล้วต้องป้อนสัญญาณพัลส์ให้แก่ขา E แล้วตามด้วยการหน่วงเวลา 125 ไมโครวินาที แต่ถ้าทำงานแบบ 4 บิตต้องแยกส่งข้อมูลทีละ 4 บิต โดยเริ่มจากส่งข้อมูล 4 บิตบนก่อนและส่งสัญญาณพัลส์ที่ขา E จากนั้นจึงส่งข้อมูล 4 บิตล่างและส่งสัญญาณพัลส์ที่ขา E ตามด้วยการหน่วงเวลา 125 ไมโครวินาที



รูปที่ จ.28 การเชื่อมต่อโมดูลหลัก PIC-ICSP กับโมดูล LCD แบบ 8 บิต

```
            list p=16f876
            #include "p16f876.inc"
            __config _PWRTE_ON & _WDT_OFF & _LVP_OFF & _XT_OSC
Reg1        equ     0x20
Cnt1        equ     0x21
Cnt2        equ     0x22
Count1      equ     0x23
Count2      equ     0x24
Count3      equ     0x25
E           equ     0               ; บรรทัดที่ 10
RS          equ     1
RW          equ     2
LCD_Con     equ     PORTA
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LCD_Data    equ     PORTB

            org     0x00
Initial     bsf     STATUS,RP0
            movlw   0x06
            movwf   ADCON1
            clrf    TRISA           ; บรรทัดที่ 20
            clrf    TRISB
            bcf     STATUS,RP0
            clrf    LCD_Con
            clrf    LCD_Data
;_____________________________________________
Start       call    Delay1
            call    InitialLCD
            call    TitleText
            call    Delay3
            goto    Start           ; บรรทัดที่ 30
;_____________________________________________
InitialLCD  clrf    LCD_Con
            clrf    Reg1
IniLCD1     movf    Reg1,W
            call    Table1
            andlw   0xff
            btfsc   STATUS,Z
            return
            movwf   LCD_Data
            call    Pulse           ; บรรทัดที่ 40
            incf    Reg1,F
            goto    IniLCD1
Table1      addwf   PCL,F
            dt      38, 0c, 06, 01, 00
;_____________________________________________
TitleText   call    Delay1
            call    ClrScreen
            clrf    Reg1
            bsf     LCD_Con,RS      ; บรรทัดที่ 50
            call    Dlay5ms
TT1         movf    Reg1,W
            call    Table2
            andlw   0xff
            btfsc   STATUS,Z
            goto    TT2
            movwf   LCD_Data
            call    Pulse
            call    Delay2          ; บรรทัดที่ 60
            incf    Reg1,F
            goto    TT1
TT2         bcf     LCD_Con,RS
            call    Dlay5ms
            movlw   0xc0
            movwf   LCD_Data
            call    Pulse
            call    Delay1
            clrf    Reg1            ; บรรทัดที่ 70
            bsf     LCD_Con,RS
            call    Dlay5ms
TT3         movf    Reg1,W
            call    Table3
            andlw   0xff
            btfsc   STATUS,Z
            goto    TT4
            movwf   LCD_Data
            call    Pulse           ; บรรทัดที่ 80
            call    Delay2
            incf    Reg1,F
            goto    TT3
TT4         clrf    LCD_Con
            clrf    LCD_Data
            return

Table2      addwf   PCL,F
            dt      "PIC 16F8XX Projects", 00     ; บรรทัดที่ 90
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Table3      addwf    PCL,F
            dt       "ELECTRONICS & COMPUTERS", 00
;_____________________________________________
ClrScreen   clrf     LCD_Con
            call     Dlay125us
            movlw    0x01
            movwf    LCD_Data
            call     Pulse
            return
;_____________________________________________
Pulse       bsf      LCD_Con,E       ; บรรทัดที่ 100
            Call     Dlay125us
            bcf      LCD_Con,E
            call     Dlay125us
            return
;_____________________________________________
Dlay125us   movlw    0x2a
            Movwf    Cnt1
Dlay1       decfsz   Cnt1,F
            goto     Dlay1           ; บรรทัดที่ 110
            return
;_____________________________________________
Dlay5ms     movlw    0x29
            Movwf    Cnt2
Dlay2       call     Dlay125us
            Decfsz   Cnt2,F
            Goto     Dlay2
            Return
;_____________________________________________
Delay1      movlw    0xff            ; บรรทัดที่ 120
            movwf    Count1
Dely1       decfsz   Count1,F
            Goto     Dely1
            Return

Delay2      movlw    0xa0
            Movwf    Count2
Dely2       call     Delay1
            Decfsz   Count2,F
            goto     Dely2           ; บรรทัดที่ 130
            return

Delay3      movlw    0x05
            Movwf    Count3
Dely3       call     Delay2
            Decfsz   Count3,F
            Goto     Dely3
            Return
            end                      ; บรรทัดที่ 140
```

รูปที่ จ.29 โปรแกรมการแสดงผลของ LED แบบ 8 บิต

ลำดับขั้นตอนการทดลอง

1. เขียนโปรแกรมรูปที่ จ.19 ทำการแอสเซมเบลอร์แล้วเขียนข้อมูลลงบน PIC16F876
2. ต่อสายเชื่อมต่อตามรูปที่ จ.28
3. ทำการรันโปรแกรม โดยการคลิกปุ่ม RUN
4. ผลการทำงานของ LCD เป็นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

.....

.....

5. จากโปรแกรมดังรูปที่ จ.29 ถ้าเปลี่ยนการแสดงข้อความเป็น KMITL ทั้ง 2 บรรทัดจะต้องแก้ไขโปรแกรมอย่างไร

.....

.....

.....

.....

6. จากการทดลองข้อ 5 ทำการแอสเซมเบลอร์แล้วเขียนข้อมูลลงบน PIC16F876 อีกครั้ง

7. ทำการรันโปรแกรม โดยการคลิกปุ่ม RUN

8. ผลการทำงานของ LCD เป็นดังนี้

.....

.....

สรุปผลการทดลอง

.....

.....

.....

.....

.....

.....

คำถามท้ายการทดลอง

1. จงอธิบายหลักการทำงานของโปรแกรมดังรูปที่ จ.29

2. จงเขียนโปรแกรมเพื่อควบคุมให้ LCD แสดงข้อความเป็นชื่อของตัวเอง โดยเลื่อนจากซ้ายไปขวาเมื่อครบให้เลื่อนจากขวาไปซ้ายกลับกันโดยใช้ LCD แบบ 4 บิต

3. จงเขียนโปรแกรมเพื่อควบคุมให้ LCD แสดงข้อความ “Test LCD” โดยเลื่อนจากขวาไปซ้ายเมื่อครบแล้วให้แสดงเคอร์เซอร์กระพริบรอที่ท้ายประโยค โดยใช้ LCD แบบ 8 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 6
การใช้งาน PIC16F876 ขับแอลอีดีตัวเลข 7 ส่วน

จุดประสงค์

1. เพื่อให้สามารถใช้ PIC16F876 นำข้อมูลออกไปแสดงผลที่แอลอีดีตัวเลข 7 ส่วนได้
2. เพื่อให้สามารถใช้งานแอลอีดีตัวเลข 7 ส่วน ทั้งในลักษณะแสดงผลแบบหลักเดี่ยวและแบบหลายหลักหรือมัลติเพล็กซ์ (Multiplex) ได้
3. เพื่อให้สามารถเขียนโปรแกรมเพื่อแสดงผลบนแอลอีดีตัวเลข 7 ส่วน ทั้งในลักษณะแสดงผลแบบหลักเดี่ยวและแบบมัลติเพล็กซ์ได้

อุปกรณ์ที่ใช้ในการทดลอง

1. โมดูลแสดงผลแอลอีดีตัวเลข 7 ส่วน
2. โมดูลหลัก PIC-ICSP
3. คอมพิวเตอร์พีซีที่ติดตั้งโปรแกรม MPLAB
4. สายเชื่อมต่อระหว่างโมดูลหลัก PIC-ICSP กับคอมพิวเตอร์พีซี
5. สายเชื่อมต่อพอร์ตระหว่างโมดูล

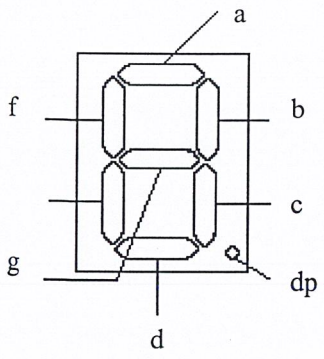
ทฤษฎีเบื้องต้น

ความรู้เบื้องต้นเกี่ยวกับแอลอีดีตัวเลข 7 ส่วน

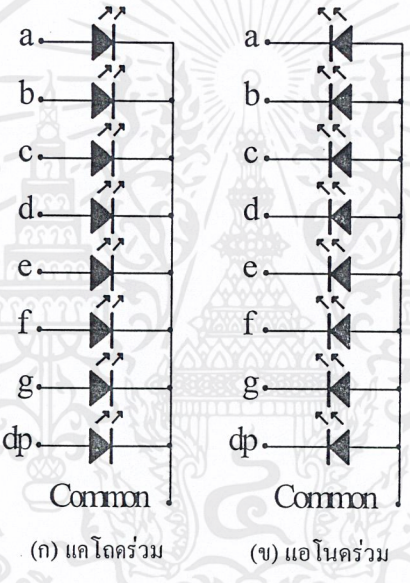
การแสดงผลที่เป็นแอลอีดีตัวเลข 7 ส่วน (LED 7 Segment) แต่ละส่วนหรือเซกเมนต์มีชื่อเรียกแตกต่างกันตามตำแหน่งที่ได้รับการจัดวาง คือ a, b, c, d, e, f และ g ดังรูปที่ จ.30 ส่วน db เป็นแอลอีดีอีก 1 ใช้แสดงตัวเลขที่มีจุดทศนิยม

แอลอีดีตัวเลข 7 ส่วนนี้มีทั้งแบบคาโถดร่วม (common cathode) และแบบอาโนดร่วม (common anode) การขับให้แอลอีดีตัวเลข 7 ส่วน แบบคาโถดร่วมสว่างจะต้องจ่ายไฟลบเข้าที่ขาร่วม แล้วจ่ายไฟบวกเข้าที่ขาอาโนด ซึ่งก็คือขาของแต่ละเซกเมนต์นั่นเองดังรูปที่ จ.31 ก.ในขณะที่แอลอีดีตัวเลข 7 ส่วนแบบอาโนดร่วมจะต้องจ่ายไฟบวกเข้าที่ขาร่วมแล้วจ่ายไฟลบเข้าที่ขาคาโถด ซึ่งเป็นขาของแต่ละเซกเมนต์ ดังรูปที่ จ.31 ข.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ จ.30 รูปร่างและการกำหนดชื่อเซกเมนต์ต่างๆของแอลอีดีตัวเลข 7 ส่วน



รูปที่ จ.31 วงจรภายในของแอลอีดีตัวเลข 7 ส่วนทั้งแบบคาโถดร่วมและอาโนดร่วม

การขับแอลอีดีตัวเลข 7 ส่วนแบบหลักเดียว

การกำหนดให้แอลอีดีตัวเลข 7 ส่วนแสดงข้อมูลเป็นตัวเลขหรือสัญลักษณ์ใดๆ ก็ตาม ต้องมีการกำหนดรูปแบบการแสดงผลของเซกเมนต์ต่างๆ ด้วยข้อมูลแต่ละบิตของไมโครคอนโทรลเลอร์แล้วใช้วิธีการเปิดตารางข้อมูลของการแสดงผลตัวเลขฐานสิบหกของแอลอีดีตัวเลข 7 ส่วน ในตารางที่ จ.5 ซึ่งเป็นตารางข้อมูลของแสดงผลของแอลอีดีตัวเลข 7 ส่วนแบบคาโถดร่วม แต่ถ้าต้องการข้อมูลที่ใช้กับแอลอีดีตัวเลข 7 ส่วนแบบอาโนดร่วมก็จะต้องเป็นข้อมูลเลขตรงข้ามกับแบบคาโถด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ จ.5 ข้อมูลของการแสดงผลตัวเลข 0-F ของแอลอีดีตัวเลข 7 ส่วนแบบคาโถดร่วม

ข้อมูลดิจิตอลเอาต์พุตสำหรับขับตัวเลข 7ส่วน								ค่าเลขฐานสิบหกที่ใช้กับ PIC16F876	ค่าตัวเลขที่แสดงที่ 7 เซกเมนต์
D7	D6	D5	D4	D3	D2	D1	D0		
0	0	1	1	1	1	1	1	0x3F	0
0	0	0	0	0	1	1	0	0x06	1
0	1	0	1	1	0	1	1	0x5B	2
0	1	0	0	1	1	1	1	0x4F	3
0	1	1	0	0	1	1	0	0x66	4
0	1	1	0	1	1	0	1	0x6D	5
0	1	1	1	1	1	0	1	0x7D	6
0	0	0	0	0	1	1	1	0x07	7
0	1	1	1	1	1	1	1	0x7F	8
0	1	1	0	1	1	1	1	0x6F	9
0	1	1	1	0	1	1	1	0x77	A
0	1	1	1	1	1	0	0	0x7C	b
0	0	1	1	1	0	0	1	0x39	C
0	1	0	1	1	1	1	0	0x5E	d
0	1	1	1	1	0	0	1	0x79	E
0	1	1	1	0	0	0	1	0x71	F
1	1	1	1	1	1	1	1	0xFF	88

การขับแอลอีดีตัวเลข 7 ส่วน แบบมัลติเพล็กซ์

ถ้าเราต้องการให้ PIC16F876 ขับแอลอีดีตัวเลข 7 ส่วนได้มากกว่า 1 หลัก จะต้องใช้เทคนิคการแสดงผลที่เรียกว่า การแสดงผลแบบมัลติเพล็กซ์อันเป็นวิธีการขับให้แอลอีดีสว่างทีละหลักด้วยอัตราเร็วที่ตาของมนุษย์ไม่สามารถตรวจจับได้ทัน จึงเหมือนว่าแอลอีดีทุกหลักติดสว่างในเวลาเดียวกัน

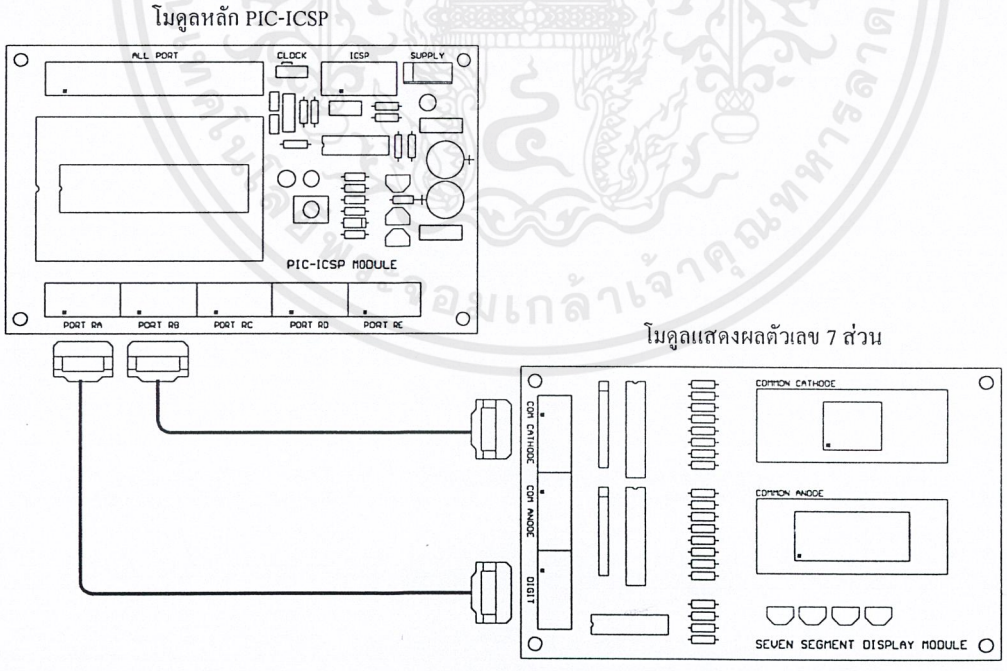
ประโยชน์ของการแสดงผลแบบมัลติเพล็กซ์นี้มีหลายประการดังนี้

1. ช่วยลดพลังงานไฟฟ้าที่ใช้ ทำให้ขนาดของแหล่งจ่ายกำลังงานไฟฟ้าเล็กลง
2. ช่วยประหยัดพอร์ตในการขับแอลอีดีตัวเลข 7 ส่วนหลายๆ หลัก
3. ลดจำนวนตัวต้านทานที่ใช้ในการจำกัดกระแสของแอลอีดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การขับแอลอีดีตัวเลข 7 ส่วนแบบมัลติเพล็กซ์จะทำการต่อขาของแต่ละเซกเมนต์ร่วมกันคือ เซกเมนต์ a ของทุกหลักจะต่อถึงกันไล่เรียงกันไปจนถึงเซกเมนต์ db การควบคุมให้หลักใดติดสว่าง ทำได้โดยการจ่ายไฟเข้าที่ขาร่วมของตัวเลข 7 ส่วนหลักนั้นๆ เช่น ถ้าตัวเลข 7 ส่วนที่ใช้เป็นแบบคาโถดร่วม หากต้องการให้หลักที่ 3 ติดสว่างก็ให้ต่อขาร่วมของหลักที่ 3 ลงกราวด์หรือจ่ายไฟลบให้แล้วส่งข้อมูลที่ต้องการให้ติดเข้ามาทางเซกเมนต์ แอลอีดีตัวเลข 7 ส่วนหลักที่ 3 ก็จะติดสว่างตามข้อมูลที่ต้องการ

กระบวนการเริ่มต้นโดย PIC16F876 ส่งข้อมูลออกไปยังขาพอร์ตที่ต่ออยู่กับเซกเมนต์ a-g และ db (ในกรณีที่ต้องการใช้ขา db) ของแอลอีดีตัวเลข 7 ส่วนก่อน จากนั้นจึงส่งข้อมูล “0” ไปยังขาร่วมของแอลอีดีตัวเลข 7 ส่วนในหลักที่ต้องการให้แสดงผล เช่นถ้าต้องการแสดงตัวเลข 1234 ต้องส่งข้อมูลของเลข 1 ไปก่อนแล้วจึงส่ง “0” ไปยังขาร่วมของหลักที่ 4 จากนั้นจึงส่งข้อมูลเลข 2 แล้วส่ง “0” ไปยังขาร่วมของหลักที่ 3 ทยอยไล่ไปตามลำดับด้วยความเร็วสูง ในอัตราที่ตาของมนุษย์ไม่สามารถสังเกตเห็นความเปลี่ยนแปลงดังกล่าว ภาพที่เห็นจึงกลายเป็นตัวเลข 1234 ติดพร้อมกันทุดหลัก จะเห็นได้ว่าด้วยกระบวนการนี้แอลอีดีตัวเลข 7 ส่วนจะทำงานไม่พร้อมกัน การกินกระแสไฟฟ้าจึงมีค่าสูงสุดเท่ากับแอลอีดีทุกเซกเมนต์ในหนึ่งหลักติดสว่างพร้อมกันเท่านั้น



รูปที่ จ.32 การเชื่อมต่อระหว่างโมดูลหลัก PIC-ICSP และโมดูลแสดงตัวเลข 7 ส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            List p=16f876
            #include "p16f876.inc"
            __config _WDT_OFF & _PWRTE_ON & _LVP_OFF & _XT_OSC
COUNT1      equ     0x20
COUNT2      equ     0x21
REG1        equ     0x23

            org     0x00
            bsf     STATUS,RP0
            clrf    PORTA
            clrf    PORTB
            bcf     STATUS,RP0
            bcf     STATUS,C
            clrf    PORTB
ClrReg      clrf    REG1
Loop        movf    REG1,W
            call    Table1
            bcf     STATUS,Z
            andlw   0xFF
            btfsc   STATUS,Z
            goto    ClrReg1
            movwf   PORTB
            movlw   0x00
            movwf   PORTA
            call    Delay
            incf    REG1,F
            goto    Loop

;------------------------------------------- TABLE DATA
Table2      addwf   PCL,F
            dt      3f,06,5b,4f,66,6d,7d,07,7f,6f,00

;------------------------------------------- DELAY TIME
Delay       movlw   0x20
            Movwf   COUNT1
Delay1      movlw   0xff
            Movwf   COUNT2
Delay2      decfsz  COUNT2,1
            Goto    Delay2
            decfsz  COUNT1,1
            goto    Delay1
            return
            end
```

รูปที่ จ.33 โปรแกรมแสดงผลแอลอีดีตัวเลข 7 ส่วนหลักเดียว

```
1           List p=16f876
2
3           include "p16f876.inc"
4           __config _WDT_OFF & _PWRTE_ON & _LVP_OFF & _XT_OSC
5 COUNT1    equ     0x20
6 COUNT2    equ     0x21
7
8           org     0x00
9           bsf     STATUS,RP0
10          clrf    PORTA
11          clrf    PORTB
12          bcf     STATUS,RP0
13          bcf     STATUS,C
14          clrf    PORTB
15
16          movlw   0x66
17          movwf   PORTB
18          movlw   0x00
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 19          movwf    PORTA
 20          call     Delay
 21
 22          movlw    0x4f
 23          movwf    PORTB
 24          movlw    0x01
 25          movwf    PORTA
 26          call     Delay
 27
 28          movlw    0x5b
 29          movwf    PORTB
 30          movlw    0x02
 31          movwf    PORTA
 32          call     Delay
 33
 34          movlw    0x06
 35          movwf    PORTB
 36          movlw    0x03
 37          movwf    PORTA
 38          call     Delay
 39;-------------------------------- DELAY TIME
 40 Delay    movlw    0x20
 41          movwf    COUNT1
 42 Delay1   movlw    0xff
 43          movwf    COUNT2
 44 Delay2   decfsz   COUNT2,1
 45          goto     Delay2
 46          decfsz   COUNT1,1
 47          goto     Delay1
 48          return
 49
 50          end
```

รูปที่ จ.34 โปรแกรมแสดงผลแอลอีดีตัวเลข 7 ส่วนหลายหลัก

ลำดับขั้นตอนการทดลอง

1. ต่อสายเชื่อมต่อระหว่างโมดูลตามรูปที่ จ.32

2. เขียนโปรแกรมดังรูปที่ จ.33 ทำการแอสเซมเบลอร์ จากนั้นเขียนข้อมูลลงบน PIC16F876

3. ทำการรันโปรแกรมบน PIC16F876 โดยคลิกปุ่ม RUN สังเกตผลที่เกิดขึ้น

.....

.....

4. ถ้าเปลี่ยนข้อมูลในโปรแกรมเป็น dt 6f,7f,07,7d,6d,66,4f,5b,06,3f,00 จากนั้นทำเหมือนขั้นตอนที่ 2 –3 ผลจากการสังเกตเป็นดังนี้

.....

.....

5. จากโปรแกรมดังรูปที่ จ.33 เปลี่ยนข้อมูลจาก movlw 0x00 มาเป็น movlw 0x01 ผลจากการสังเกตเป็นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

.....

.....

6. จากข้อ 5 ถ้าต้องการให้หลักที่ 4 ติด (หลักแรกซ้ายมือ) จะต้องเปลี่ยนข้อมูลเป็นเท่าไร

.....

.....

7. จากโปรแกรมดังรูปที่ จ.33 เปลี่ยนข้อมูลจาก movlw 0x00 มาเป็น movlw 0x04 และเปลี่ยนข้อมูลจาก dt 3f,06,5b,4f,66,6d,7d,07,7f,6f,00 มาเป็น dt b0,f9,a4,cf,99,92,82,f8,80,90,00

.....

.....

8. ทำเหมือนขั้นตอนที่ 2 –3 ผลจากการสังเกตเป็นดังนี้

.....

.....

9. จากข้อ 8 ถ้าเปลี่ยนข้อมูลจาก movlw 0x04 มาเป็น movlw 0x05 ผลจากการสังเกตเป็นดังนี้

.....

.....

10. จากข้อ 9 ถ้าต้องการให้หลักที่ 4 ติด (หลักแรกซ้ายมือ) จะต้องเปลี่ยนข้อมูลเป็นเท่าไร

.....

.....

11. เขียนโปรแกรมที่ 11.5 ทำการแอสเซมเบลอร์ จากนั้นเขียนข้อมูลลงบน PIC16F876

12. ต่อสายเชื่อมต่อระหว่างโมดูลตามรูปที่ จ.32

13. ทำการรันโปรแกรมบน PIC16F876 โดยการคลิกปุ่ม RUN สังเกตผลที่เกิดขึ้น

.....

.....

14. ในบรรทัดที่ 16 ถ้าต้องการเปลี่ยนข้อมูลเป็น 0x77 และในบรรทัดที่ 34 เป็น 0xFF สังเกตผลที่เกิดขึ้น

.....

.....

15. จากโปรแกรมที่ 11.5 ถ้าต้องการให้แสดงผลแบบอาโนดร่วมต้องแก้ไขโปรแกรมอย่างไรบ้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

.....

.....

.....

.....

.....

.....

สรุปผลการทดลอง

.....

.....

.....

.....

.....

.....

.....

.....

.....

คำถามท้ายการทดลอง

1. จงอธิบายความแตกต่างระหว่าง แอลอีดีตัวเลข 7 ส่วนแบบคาโถดร่วมและแบบอาโนดร่วม

2. จงเขียนโปรแกรมนับเลข 0-99 โดยให้แสดงผลในแอลอีดีตัวเลข 7 ส่วน แบบอาโนดร่วม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 7
การเชื่อมต่อ PIC16F876 กับสวิตช์เมตริกซ์ขนาด 4x4

จุดประสงค์

1. เพื่อให้สามารถเชื่อมต่อ PIC16F876 กับสวิตช์เมตริกซ์ (Switch matrix) ขนาด 4x4 ได้
2. เพื่อให้สามารถเขียนโปรแกรมเพื่อรับค่าของสวิตช์ได้

อุปกรณ์ที่ใช้ในการทดลอง

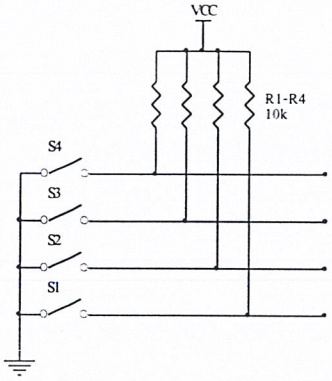
1. โมดูลหลัก PIC-ICSP
2. โมดูลสวิตช์เมตริกซ์ขนาด 4x4
3. คอมพิวเตอร์พีซีที่ติดตั้งโปรแกรม MPLAB
4. สายเชื่อมต่อระหว่างโมดูลหลัก PIC-ICSP กับคอมพิวเตอร์พีซี
5. สายเชื่อมต่อพอร์ตระหว่างโมดูล

ทฤษฎีเบื้องต้น

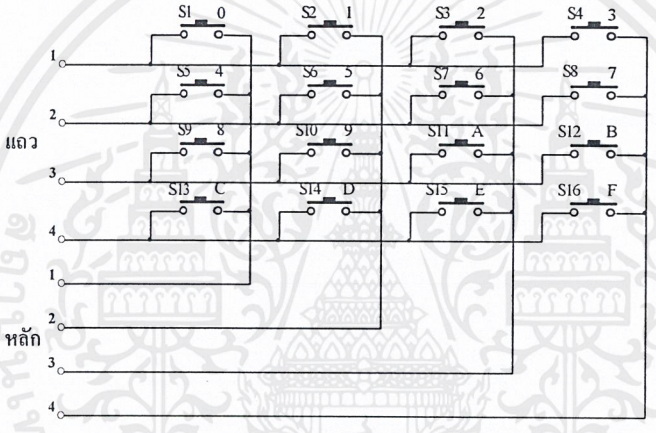
ไมโครคอนโทรลเลอร์สามารถรองรับและเชื่อมต่อใช้งานในการอ่านค่าและการกดสวิตช์ได้ วงจรของการกดสวิตช์มีการต่ออยู่ 2 แบบใหญ่ๆ คือ ต่อเข้ากับไฟเลี้ยงหรือต่อลงกราวด์โดยตรง เมื่อสวิตช์ตัวใดต่อวงจรสามารถอ่านค่าได้โดยตรง ดังในรูปที่ จ.35 วงจรนี้ไม่มีความซับซ้อน สามารถอ่านค่าได้ง่ายและรวดเร็วแต่มีข้อเสียคือ ถ้าจำนวนสวิตช์มีค่ามากๆ จำนวนของสายข้อมูลก็จะมีมากตาม ทำให้ระบบหรือวงจรมีขนาดใหญ่และทำให้สิ้นเปลืองอุปกรณ์ด้วย

วงจรของสวิตช์อีกลักษณะหนึ่งคือ การต่อวงจรแบบเมตริกซ์ ดังรูปที่ จ.36 สวิตช์จะถูกต่อกันในแนวแกนตั้งและแกนนอนจะเรียกแนวตั้งว่า หลักหรือคอลัมน์ (Column) ในขณะที่แนวนอนจะเรียกว่า แถวหรือโรว์ (row) ดังนั้นค่าของสวิตช์จะต้องประกอบด้วย ตำแหน่งในแนวหลักและแถว กระบวนการที่จะทำให้ได้มาซึ่งค่าของสวิตช์มีขั้นตอนที่ซับซ้อนพอสมควร แต่วงจรของสวิตช์แบบนี้มีข้อดีคือสามารถรองรับการเพิ่มของสวิตช์ได้อย่างสะดวกเพียงเพิ่มเติมจำนวนสวิตช์และแก้ไขซอฟแวร์อีกเล็กน้อยเท่านั้น ทำให้วงจรสวิตช์แบบเมตริกซ์เป็นที่นิยมใช้กันมากในระบบควบคุมอัตโนมัติที่มีจำนวนสวิตช์มากกว่า 8 ตัว ในการใช้งานทั่วไปจะเรียกสวิตช์แบบเมตริกซ์นี้ว่า คีย์แพด (Keypad)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ จ.35 วงจรของสวิตช์แบบต่อเข้ากับไฟเลี้ยงและกราวด์

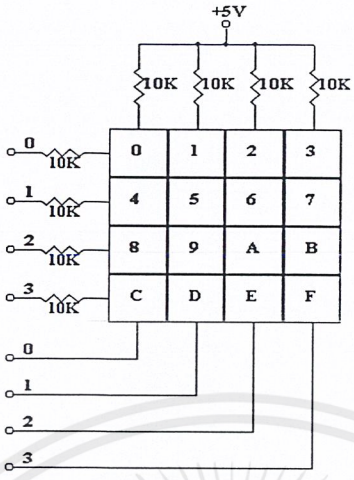


รูปที่ จ.36 การต่อสวิตช์แบบเมตริกซ์

สายสัญญาณในแนวคอลัมน์ของคีย์แพดจะต่อเข้ากับพอร์ตเอาต์พุต ส่วนสายสัญญาณในแนวแถวจะต่อเข้ากับพอร์ตอินพุตไมโครคอนโทรลเลอร์จะส่งสัญญาณ “0” ออกมาทางพอร์ตเอาต์พุต ถ้าหากสวิตช์ถูกกดพอร์ตอินพุตจะอ่านค่า “0” เข้าสู่ไมโครคอนโทรลเลอร์

จากหลักการนี้ถ้าหากคีย์แพดมีจำนวนแถวและหลักมากกว่า 1 เส้นดังรูปที่ จ.37 ขั้นตอนการอ่านค่าสวิตช์ที่ถูกกดเริ่มต้นด้วยไมโครคอนโทรลเลอร์ส่งสัญญาณออกมาทางพอร์ตเอาต์พุตเข้าสู่แต่ละคอลัมน์ของคีย์แพดไล่เรียงกันไปทีละคอลัมน์แล้วอ่านค่าจากสายสัญญาณแต่ละหลักทางพอร์ตอินพุต ถ้าหากค่าของอินพุตที่อ่านเข้ามามีค่าไม่เท่ากับค่าของเอาต์พุตที่ส่งออกไปหรือยังเป็นค่าเดิมแสดงว่ายังไม่มีการกดสวิตช์ เมื่อใดที่ค่าของอินพุตอ่านเข้ามามีค่าเท่ากับค่าของเอาต์พุตที่ส่งออกไปแสดงว่ามีการกดสวิตช์เกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ จ.37 การต่อสวิตช์เมตริกส์ขนาด 4x4

ขั้นตอนการค้นหาตำแหน่งของสวิตช์ที่ถูกกด สายสัญญาณของแถวที่ต่ออยู่กับสวิตช์ที่ถูกกดนั้นจะเป็นลอจิก “0” ไมโครคอนโทรลเลอร์จะนำค่าของเอาต์พุตที่ส่งออกไปและค่าของอินพุตที่อ่านเข้ามาทำการเปิดตารางเพื่อหาค่าของตำแหน่งสวิตช์ด้วยกระบวนการการเขียนโปรแกรม

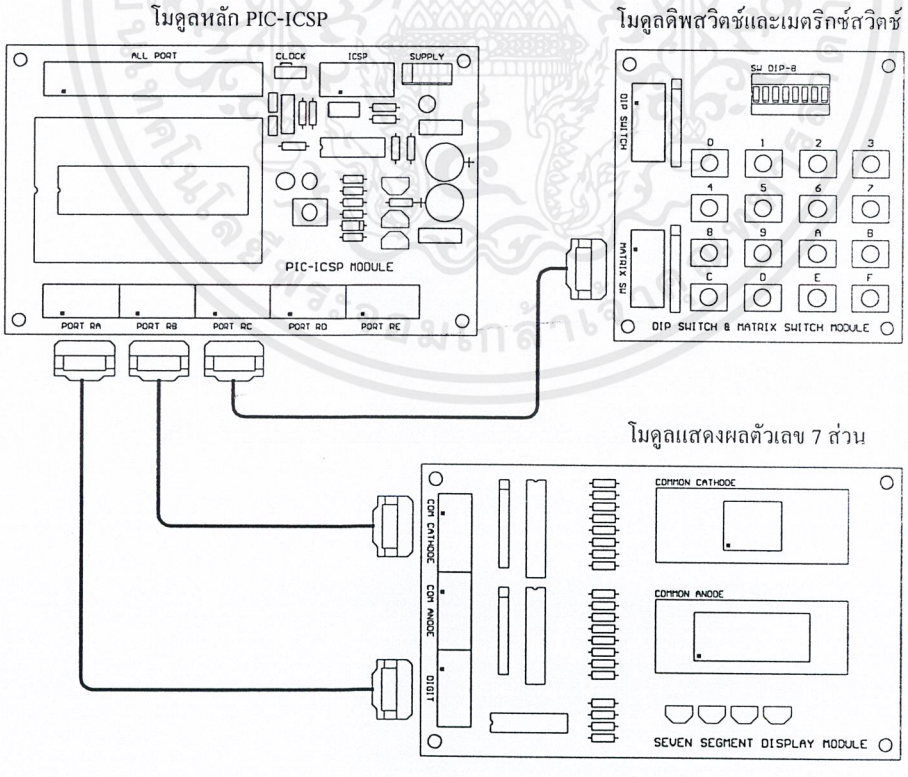
การอ่านค่าสวิตช์แบบเมตริกส์จะต้องส่งค่า “0” ในแถวออกไปก่อนโดยในแถวแรกจะต้องกำหนดค่าตัวบวกไว้เท่ากับ “0” จากนั้นอ่านค่าในแนวหลักกลับมา ถ้าไม่มีการกดคีย์ใดๆในแถวนั้นค่าที่อ่านได้จะเป็นค่าเดิมคือค่าลอจิก 1 อันเป็นผลมาจากค่าความต้านทานที่ต่อพูลอัปเอาไว้ จากนั้นก็ทำการอ่านค่าในแถวถัดไปพร้อมทั้งเพิ่มค่าตัวบวกเข้าไปด้วยจะทำเช่นนี้ไปจนครบทุกแถวแล้วกลับมาเริ่มต้นที่แถวแรกใหม่จนกว่าจะมีการกดคีย์

เมื่อมีการกดคีย์เกิดขึ้นไมโครคอนโทรลเลอร์จะนำค่าของคอลัมน์ที่มีการกดคีย์มีค่าอยู่ระหว่าง 4 หลักไปบวกกับค่าของแถวนั้น เช่น มีการกดคีย์ 5 เกิดขึ้นดังนั้นถ้าสแกนคีย์ไปในแถวที่ 1 จะพบว่ามีการกดคีย์ในคอลัมน์ที่ 1 ค่าที่อ่านได้จะมีค่าเท่ากับ 4+1 = 5 ดังแสดงในตารางที่ จ.6 แสดงการค้นหาตำแหน่งของสวิตช์ที่ถูกกด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ จ.6 สรุปการค้นหาตำแหน่งของสวิตช์ที่ถูกกด

รหัสของคีย์	ตัวนับคอลัมน์	ตัวบวก	ข้อมูลของคีย์
0 = 1110	0	0	0
1 = 1110	1	0	1
2 = 1110	2	0	2
3 = 1110	3	0	3
4 = 1101	0	4	4
5 = 1101	1	4	5
6 = 1101	2	4	6
7 = 1101	3	4	7
8 = 1011	0	8	8
9 = 1011	1	8	9
A = 1011	2	8	A(10)
B = 1011	3	8	B(11)
C = 0111	0	12	C(12)
D= 0111	1	12	D(13)
E = 0111	2	12	E(14)
F = 0111	3	12	F (15)



รูปที่ จ.38 การเชื่อมต่อโมดูล PIC-ICSP กับโมดูลสวิตช์เมตริกส์แสดงผลที่ LED เลข 7 ส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  List p=16f876
  #Include "p16f876.inc"
__config _WDT_OFF & _PWRTE_ON & _LVP_OFF & _XT_OSC
COUNT       equ     0x21
COUNTD      equ     0x22
            org     0x000
            bsf     STATUS,RP0
            movlw   0x06
            movwf   ADCON1
            movlw   0xff
            movwf   PORTA
            clrf    PORTC
            movlw   b'11111111'
            movwf   PORTB
            bcf     STATUS,RP0
            movlw   0xFF
            movwf   PORTB

Start       bcf     PORTB,0
            call    Key
            bsf     PORTB,0
            bcf     PORTB,1
            call    Key
            bsf     PORTB,1
            bcf     PORTB,2
            call    Key
            bsf     PORTB,2
            bcf     PORTB,3
            call    Key
            bsf     PORTB,3
            goto    Start

Key         movlw   0x00
            movwf   COUNT
            btfss   PORTB,4
            call    Display
            incf    COUNT,F
            btfss   PORTB,5
            goto    Display
            incf    COUNT,F
            btfss   PORTB,6
            goto    Display
            incf    COUNT,F
            btfss   PORTB,7
            goto    Display
            return

Display     clrf    PORTA
            movf    COUNT,W
            call    DATA1
            movwf   PORTC
            call    Delay
            movlw   0x01
            movwf   PORTA
            return
DATA1       addwf   PCL,F
            dt      3f, 06, 5b, 4f, 66, 6d, 7d, 07
            dt      7f, 6f, 77, 7c, 39, 5e, 79, 71 ,00

Delay       movlw   0x02
            movwf   COUNTD
Del1        decfsz  COUNTD,F
            goto    Del1
            return
            end
```

รูปที่ จ.39 โปรแกรมการอ่านค่าคีย์แพดโดยใช้หลักการสแกนคีย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับขั้นตอนการทดลอง

1. เขียนโปรแกรมดังรูปที่ จ.39 ทำการแอสเซมเบลอร์
2. ต่อสายเชื่อมต่อระหว่างโมดูลตามรูปที่ จ.38
3. เขียนข้อมูลลงบน PIC16F876 ทำการรันโปรแกรมบน PIC16F876 โดยการคลิกปุ่ม RUN
4. เมื่อกดสวิตช์แต่ละตัว ผลที่ได้คือ

ตารางที่ จ.7 ผลการติดของไดโอดเปล่งแสงตัวเลข 7 ส่วนจากการกดสวิตช์

กดสวิตช์	แสดงผลที่ตัวเลข 7 ส่วน
0	.....
1	.....
2	.....
3	.....
4	.....
5	.....
6	.....
7	.....
8	.....
9	.....
A	.....
B	.....
C	.....
D	.....
E	.....
F	.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลการทดลอง

.....

.....

.....

.....

.....

.....

.....

.....

คำถามท้ายการทดลอง

1. จงอธิบายหลักการทำงานของโปรแกรมดังรูปที่ จ.39
2. จงอธิบายหลักการทำงานของสวิตช์เมตริกส์ขนาด 4X4 มาพอเข้าใจ
3. จงเขียนโปรแกรมรับค่าจากสวิตช์เมตริกส์ 4X4 ที่เข้ามาที่พอร์ต C โดยให้พอร์ต C ทางด้าน 4 บิตล่างเป็นเอาต์พุตและ 4 บิตบนเป็นเอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 8
การเชื่อมต่อ PIC16F876 กับมอเตอร์ไฟฟ้ากระแสตรง (DC Motor)

จุดประสงค์

1. เพื่อให้สามารถเลือกอุปกรณ์ขับกระแสและแรงดันที่เหมาะสมได้
2. เพื่อให้สามารถใช้งาน PIC16F876 และอุปกรณ์ขับกระแสและแรงดันชนิดต่างๆได้
3. เพื่อให้สามารถเขียนโปรแกรมควบคุมมอเตอร์ไฟฟ้ากระแสตรง(DC Motor)ได้

อุปกรณ์ที่ใช้ในการทดลอง

1. โมดูลหลัก PIC-ICSP
6. โมดูลการขับมอเตอร์
7. คอมพิวเตอร์พีซีที่ติดตั้งโปรแกรม MPLAB
8. สายเชื่อมต่อระหว่างโมดูลหลัก PIC-ICSP กับคอมพิวเตอร์พีซี
9. สายเชื่อมต่อพอร์ตระหว่างโมดูล

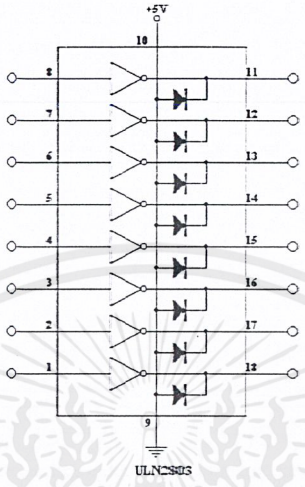
ทฤษฎีเบื้องต้น

มอเตอร์ คือ อุปกรณ์ที่ทำหน้าที่เปลี่ยนพลังงานไฟฟ้าให้เป็นพลังงานกล และมอเตอร์ที่ขับเคลื่อนด้วยกระแสไฟฟ้าดีซีเรียกว่า มอเตอร์ดีซี การควบคุมมอเตอร์เราสามารถควบคุมทิศทางการหมุน หรือควบคุมให้หมุนช้าหรือเร็วก็ได้โดย ควบคุมและผ่านไอซี ULN2803 ซึ่งเป็นตัวช่วยขับกระแสและแรงดันโดยไอซี UNL2803 เป็นไอซีไดรเวอร์ที่มีความสามารถสูง ภายในจะบรรจุอินเวอร์เตอร์เกต (Inverter gate) แบบคอลเล็กเตอร์เปิด (Open collector) 8 ตัว สามารถใช้แรงดันได้สูงสุดบวก 50 โวลต์ กระแสเอาต์พุตสูงสุดในแต่ละขาเท่ากับ 500 มิลลิแอมป์ นอกจากนั้นยังต่อไดโอดป้องกันไว้ทุกๆ เอาต์พุต ทำให้สามารถต่อโหลดที่เป็นขดลวดได้ทันที วงจรภายในและขาต่อใช้งานของไอซีไดรเวอร์เบอร์ UNL2803 เป็นดังรูปที่ จ.40

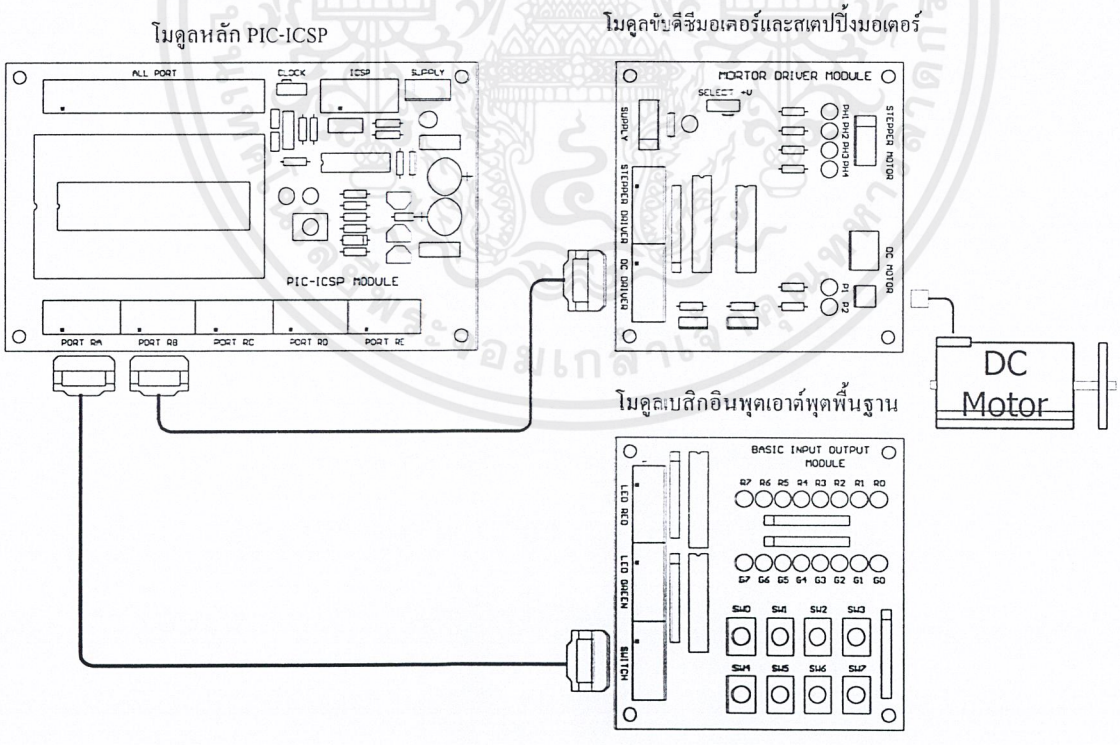
การควบคุมความเร็วการหมุนของมอเตอร์นั้นก็สามารถทำได้โดย จ่ายกระแสในลักษณะของพัลส์และควบคุมความกว้างของพัลส์ในช่วงแรงดัน 5 โวลต์ หากความกว้างของระดับแรงดันมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่ามากค่ากระแสเฉลี่ยก็จะมีค่าสูงขึ้นทำให้มอเตอร์หมุนเร็วขึ้นในทางตรงข้ามถ้าระดับแรงดันมีค่าลดลงค่ากระแสเฉลี่ยก็มีค่าลดลงทำให้มอเตอร์หมุนช้าลงด้วย



รูปที่ จ.40 การจัดขาและวงจรภายในของไอซี ULN2803



รูปที่ จ.41 การเชื่อมโมดูลหลัก PIC-ICSP, โมดูลเบสิกอินพุตเอาต์พุตและโมดูลขับมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    List p=16f876
    #Include "p16f876.inc"
  __config _WDT_OFF & _PWRTE_ON & _LVP_OFF & _XT_OSC
COUNT1      equ     0x21
COUNT2      equ     0x22

            org     0x00
            bsf     STATUS,RP0
            movlw   0x06
            movwf   ADCON1
            movlw   0xFF
            movwf   TRISA
            clrf    TRISB
            bcf     STATUS,RP0
            clr     PORTB
Sw1         btfsc   PORTA,0
            goto    Sw2
            movlw   0x00
            movwf   PORTB
            call    Delay

Sw2         btfsc   PORTA,1
            goto    Sw3
            movlw   0x02
            movwf   PORTB
            call    Delay

Sw3         btfsc   PORTA,2
            goto    Sw4
            movlw   0x04
            movwf   PORTB
            call    Delay

Sw4         btfsc   PORTA,3
            goto    Sw1
            movlw   0x06
            movwf   PORTB
            call    Delay

Delay       movlw   0x20
            movwf   COUNT1
Delay1      movlw   0xFF
            movwf   COUNT2
Delay2      decfsz  COUNT2,1
            goto    Delay2
            decfsz  COUNT1,1
            goto    Delay1
            end
```

รูปที่ จ.42 การขับมอเตอร์ไฟฟ้ากระแสตรง (DC Motor)

ลำดับขั้นตอนการทดลอง

1. เขียนโปรแกรมดังรูปที่ จ.42 ทำการแอสเซมเบลอร์แล้วเขียนข้อมูลลงบน PIC16F876
2. ต่อสายเชื่อมต่อระหว่างโมดูลตามรูปที่ จ.41
3. ทำการรันโปรแกรมบน PIC16F876 โดยคลิกปุ่ม RUN สังเกตผลที่เกิดขึ้น

.....

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เมื่อกดสวิตช์ SW1 ผลที่ได้คือ

.....

5. เมื่อกดสวิตช์ SW2 ผลที่ได้คือ

.....

6. เมื่อกดสวิตช์ SW3 ผลที่ได้คือ

.....

7. เมื่อกดสวิตช์ SW4 ผลที่ได้คือ

.....

สรุปผลการทดลอง

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

คำถามท้ายการทดลอง

1. จงอธิบายโปรแกรมดังรูปที่ จ.40

2. จงเขียนโปรแกรมควบคุมมอเตอร์ไฟฟ้ากระแสตรงโดยมีระดับความเร็วของมอเตอร์ 4 ระดับคือกด SW1 ให้หมุนช้า กด SW2 ให้หมุนเร็ว กด SW3 ให้หมุนเร็วขึ้น และกด SW4 จะหมุนเร็วที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 9
การเชื่อมต่อ PIC16F876 กับวงจรมอดูเลชั่น ทางความกว้างของพัลซ์ (PWM)

จุดประสงค์

1. เพื่อให้สามารถอธิบายหลักการทำงานของการมอดูเลชั่นทางความกว้างของพัลซ์ได้
2. เพื่อให้สามารถใช้ PIC16F876 สร้างวงจรมอดูเลชั่นทางความกว้างของพัลซ์ (PWM) ได้
3. เพื่อให้สามารถประยุกต์ใช้งานวงจรมอดูเลชั่นทางความกว้างของพัลซ์ (PWM) ที่มี PIC16F876 เป็นอุปกรณ์หลักได้

อุปกรณ์ที่ใช้ในการทดลอง

1. โมดูลหลัก PIC-ICSP
2. โมดูลเบสิกอินพุตเอาต์พุต
3. โมดูลขับดีซีมอเตอร์และสเตปปิ้งมอเตอร์
4. คอมพิวเตอร์พีซีที่ติดตั้งโปรแกรม MPLAB
5. สายเชื่อมต่อระหว่างโมดูลหลัก PIC-ICSP กับคอมพิวเตอร์พีซี
6. สายเชื่อมต่อพอร์ตระหว่างโมดูล

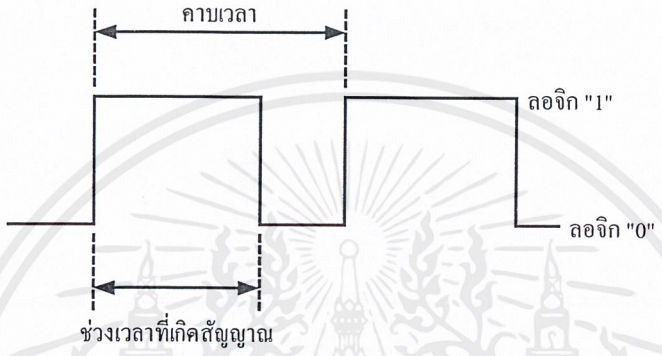
ทฤษฎีเบื้องต้น

การมอดูเลชั่นทางความกว้างของพัลซ์ หรือPWM (Pulse Width modulation)เป็นกระบวนการที่เข้าไปจัดการเกี่ยวกับสัดส่วนความกว้างของพัลซ์หรือที่เรียกว่า ดิวตี้ไซเคิล โดยที่ความถี่ของพัลซ์นั้นไม่เปลี่ยนแปลงดังรูปที่ จ.43 ดิวตี้ไซเคิล คือ ช่วงของพัลซ์ที่เป็นลอจิกสูง สัญญาณที่มีดิวตี้ไซเคิลเท่ากับ 50% หมายความว่า ในหนึ่งรูปสัญญาณมีช่วงสัญญาณที่เป็นลอจิกสูงหรือมีสัญญาณครึ่งหนึ่ง ช่วงที่เป็นลอจิกต่ำหรือไม่มีสัญญาณครึ่งหนึ่งถ้าค่าดิวตี้ไซเคิลมากก็หมายความว่าสัญญาณพัลซ์ลูกนั้นมีช่วงเวลาของสัญญาณมาก นั่นคือเป็นพัลซ์ที่ความกว้างของสัญญาณมากหากว่าค่าของดิวตี้ไซเคิลน้อย พัลซ์ลูกนั้นก็มีช่วงเวลาของสัญญาณน้อย พัลซ์จึงมีลักษณะค่อนข้างแคบ ค่าของดิวตี้ไซเคิลสามารถคำนวณได้จากความสัมพันธ์ดังนี้

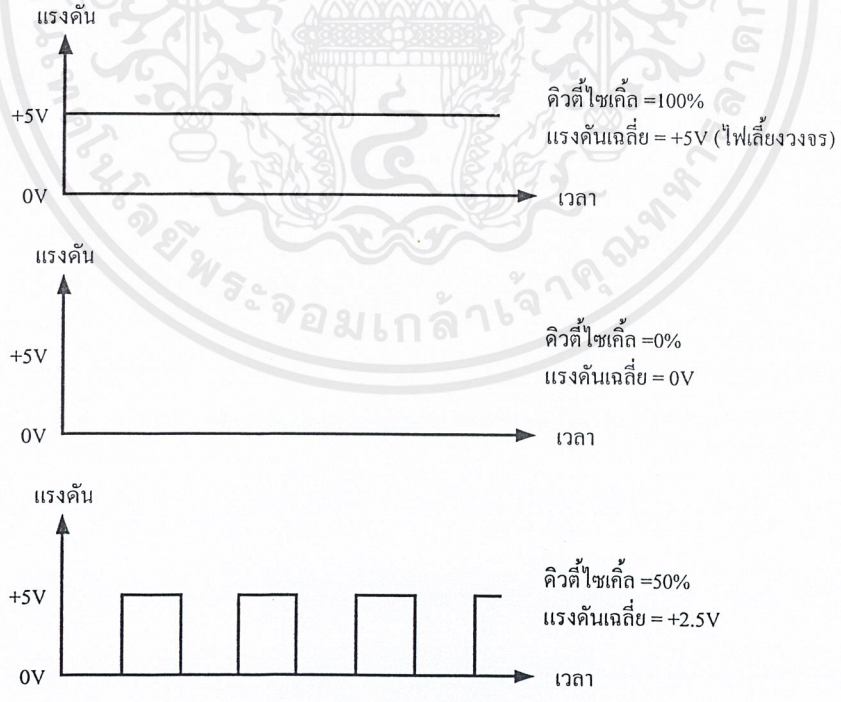
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\text{ดิวตี้ไซเคิล}(D) = \frac{\text{ช่วงเวลาของสัญญาณ}}{\text{คาบเวลาทั้งหมดของสัญญาณ} \times 100\%} \qquad (14.1)$$

ดังนั้นค่าดิวตี้ไซเคิลจะมีค่าตั้งแต่ 0-100%



รูปที่ จ.43 ดิวตี้ไซเคิลของสัญญาณ

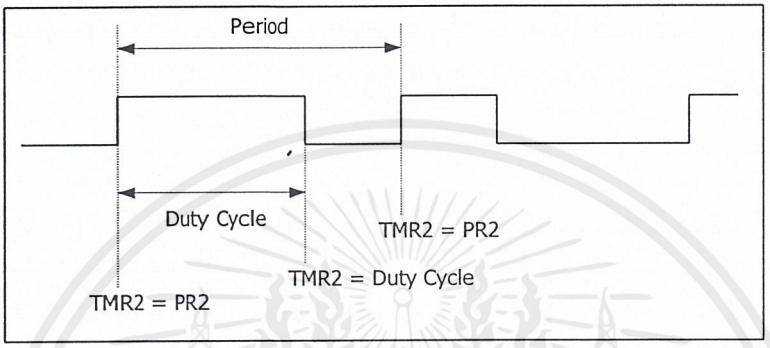


รูปที่ จ.44 ความสัมพันธ์ระหว่างค่าของดิวตี้ไซเคิล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้ PWM Module ในไมโครคอนโทรลเลอร์ PIC16F8XX

ไมโครคอนโทรลเลอร์ PIC16F8XX มีรีจิสเตอร์สำหรับสร้างสัญญาณ PWM อยู่สองตัวคือ CCP1 และ CCP2 (CCP : Capture/Compare/PWM) โดยโปรแกรมให้รีจิสเตอร์ดังกล่าวทำงานอยู่ในโหมด PWM เลือกใช้งานตัวใดก็ได้ จะให้เอาต์พุตสัญญาณ PWM ที่พอร์ต RC1 และ RC2



รูปที่ จ.45 สัญญาณ PWM เอาต์พุต

PWM period คือ คาบเวลา 1 ไซเคิลของ PWM

สามารถที่จะกำหนดค่าเวลานี้ได้โดยการกำหนดค่าในรีจิสเตอร์ PR2 ซึ่งค่าดังกล่าวสามารถคำนวณได้จาก

$$\text{PWM period} = \left[(\text{PR2}) + 1\right] \times 4 \times \text{Tosc} \times (\text{TMR2 prescale value}) \qquad (14.2)$$

(PR2) คือ ค่าในรีจิสเตอร์ขนาด 8 บิต สำหรับกำหนดความกว้างของ PWM period

Tosc คือ คาบเวลาของสัญญาณจากคริสตอล เช่น คริสตอล 4MHz จะมี Tosc = 1/4MHz

(TMR2 prescale value) คือ ค่าการหารความถี่ของ Timer2

ความถี่ของสัญญาณ PWM หาได้จาก

$$\text{PWM frequency} = \frac{1}{\text{PWM period}} \qquad (14.3)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PWM duty cycle คือ คาบเวลาที่สัญญาณ PWM เป็น "1" ใน PWM period

สามารถกำหนดเวลานี้ได้จากการใส่ค่าในรีจิสเตอร์ CCP1L ขนาด 8 บิตนัยความสำคัญด้านสูง และในรีจิสเตอร์ CCP1CON อีก 2 บิตนัยความสำคัญด้านต่ำ คือ บิต 5 และบิต 4 รวมทั้งหมด 10 บิต ค่าดังกล่าวสามารถคำนวนได้จาก

$$\text{PWM duty cycle} = (\text{CCPR1L} : \text{CCP1CON} < 5:4 >) \times \text{Tosc} \times (\text{TM2 prescale value}) \quad (14.5)$$

โดยค่าสูงสุดที่จะใส่ใน (CCPR1L:CCP1CON<5:4>) หาได้จาก $2^{\text{resolution}}$ เมื่อ

$$\text{resolution} = \frac{\log\left[\frac{F_{OSC}}{F_{PWM}}\right]}{\log(2)} \quad (14.6)$$

ตัวอย่าง ถ้าต้องการสร้างสัญญาณ PWM ความถี่ 250 Hz โดยมีค่า PWM duty cycle ที่ 25% โดยสัญญาณนาฬิกาของไมโครคอนโทรลเลอร์ใช้คริสตอลความถี่ 4 MHz

วิธีทำ สามารถคำนวนค่าในรีจิสเตอร์ PR2 และ (CCPR1L:CCP1CON<5:4>) เมื่อใช้ค่า (TMR2 prescale value) = 16 ได้จาก

(14.2) 1/250Hz = (PR2+1) * 4 * (1/4MHz) * 16

จะได้ (PR2) = 249

และ (25%)/(250Hz * 100) = (CCPR1L:CCP1CON<5:4>) * (1/4MHz) * 16

จะได้ (CCPR1L:CCP1CON<5:4>) = 250

ดังนั้นสามารถทำได้โดย

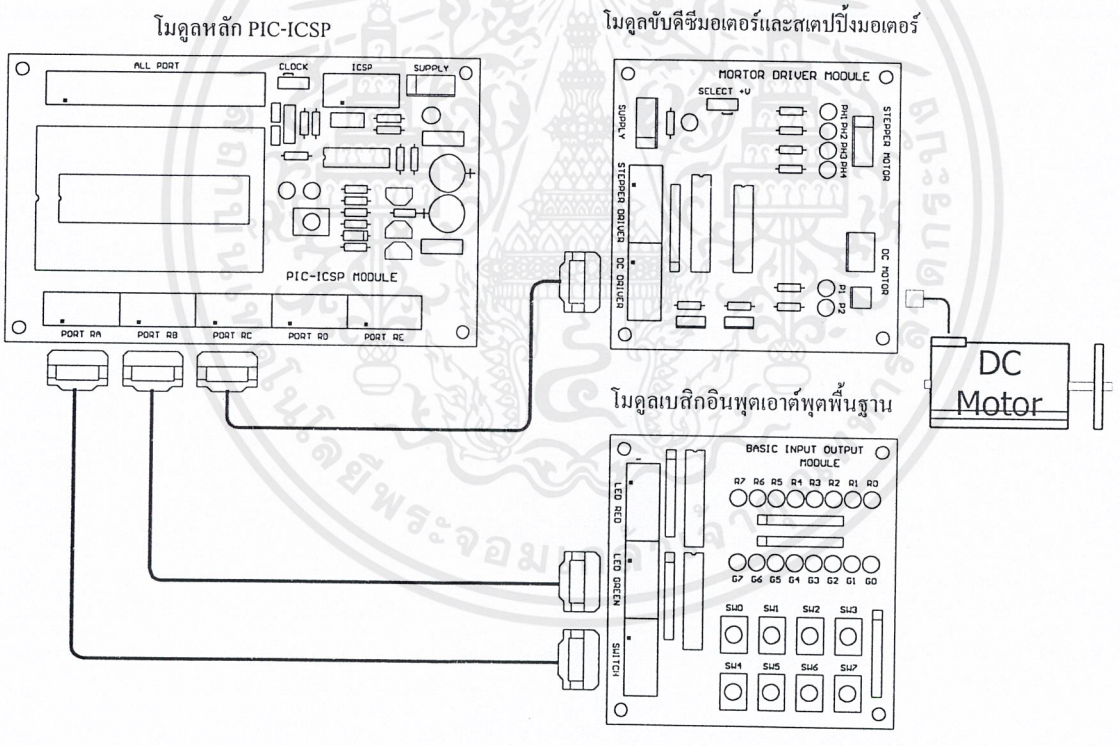
- กำหนดค่าในรีจิสเตอร์ PR2 = 249
- กำหนดค่าในรีจิสเตอร์ CCPR1L = b'00111110'
- กำหนดค่าในรีจิสเตอร์ CCP1CON<5:4> = b'10'
- กำหนดค่า TMR2 prescale = 16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประยุกต์ใช้งาน

ในการทดลองใช้ PWM ที่อยู่ในไมโครคอนโทรลเลอร์ PIC16F876 จะทำโดยการนำมาขับมอเตอร์กระแสตรง ที่สามารถปรับความเร็วได้ ซึ่งการปรับความเร็วจะทำโดยปรับค่าดิวตี้ไซเคิลของสัญญาณ PWM ถ้าหากปรับค่าดิวตี้ไซเคิลมาก จะทำให้มีแรงดันเฉลี่ยของสัญญาณมาก ทำให้มอเตอร์หมุนเร็วขึ้น แต่ถ้าปรับค่าดิวตี้ไซเคิลน้อยแรงดันเฉลี่ยก็น้อย มอเตอร์ก็จะหมุนช้า จนถึงหยุดหมุนเมื่อปรับค่าดิวตี้ไซเคิลเป็น 0 %

การเชื่อมต่อโมดูลขับดีซีมอเตอร์จะเชื่อมต่อกับ PORTC ซึ่งเป็นขาเอาต์พุตของสัญญาณ PWM ส่วนสวิตช์ที่ใช้ปรับความเร็วจะใช้สวิตช์จากโมดูลเบสิกอินพุต และก็ทำการแสดงผลค่าของ CCP1L เพื่อให้สามารถมองเห็นว่าค่ากำหนดดิวตี้ไซเคิลที่เปลี่ยนไปทำให้มอเตอร์เปลี่ยนความเร็วไปด้วย



รูปที่ จ.46 การเชื่อมต่อระหว่างโมดูล PIC-ICSP, โมดูขับดีซีมอเตอร์และโมดูลเบสิกอินพุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            List p=16f877
            #include "p16f877.inc"
            __config _PWRTE_ON & _WDT_OFF & _LVP_OFF & _XT_OSC
tempSw      equ     0x20
            radix   dec

            org     0x00
            bsf     STATUS,RP0
            movlw   0x06
            movwf   ADCON1
            movlw   0xff
            movwf   TRISA
            clrf    TRISB
            movlw   b'11111001'
            movwf   TRISC           ; make drive DC Moter
            bcf     STATUS,RP0
            movlw   249             ; 249:PWM Period = 250Hz
            movwf   PR2
            movlw   125             ; Duty cycle 50 %
            movwf   CCPR1L          ; value of (CCPR1L:CCP1CON<5:4>) = 500
            movlw   b'00001100'
            movwf   CCP1CON         ; CCP1X:CCP1Y = 00, PWM mode
            clrf    TMR2
            movlw   b'00000010'     ; Postscale (no use), Timer2 OFF, Prescaler 16
            movwf   T2CON
Start       bsf     PORTC,1
            bsf     T2CON,TMR2ON
loop        movf    CCPR1L,W
            movwf   PORTB
            call    scank
            andlw   0xff
            btfsc   STATUS,Z
            goto    loop
            sublw   1
            btfss   STATUS,Z
            goto    speedup
speeddown   movf    CCPR1L,W
            sublw   15
            btfsc   STATUS,C
            goto    Continue
            movlw   10
            subwf   CCPR1L,F
            goto    Continue
speedup     movf    CCPR1L,W        ; test CCPR1L > 245 ?
            sublw   245             ;   245 - CCPR1L = ?
            btfss   STATUS,C        ;   test carry
            goto    Continue        ; yes, no action
            movlw   10              ; no, add 10
            addwf   CCPR1L,F
            goto    Continue
;-------------- scan key -------------
scank       movlw   0
            btfss   PORTA,0
            movlw   1
            btfss   PORTA,1
            movlw   2
            subwf   tempSw,F
            movwf   tempSw
            btfsc   STATUS,C
            retlw   0
            return
            end
```

รูปที่ จ.47 โปรแกรมควบคุมความเร็วดีซีมอเตอร์โดยใช้สัญญาณพัลซ์วิธมอดูเลชั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับขั้นตอนการทดลอง

1. เขียนโปรแกรมตามรูปที่ จ.47 ทำการแอสเซมเบลอร์แล้วเขียนข้อมูลลงบน PIC16F876
2. ต่อสายเชื่อมต่อตามรูปที่ จ.46
3. ทำการรันโปรแกรม โดยการคลิกปุ่ม RUN
4. สังเกตการหมุนของมอเตอร์ และ LED P1 และ P2 ในโมดูลขับมอเตอร์

LED ติดหรือไม่ P1..... P2.....

มอเตอร์หมุนทิศทางใด.....

5. กดสวิตช์ SW0 ในโมดูลเบสิกอินพุต 5 ครั้ง สังเกตการเปลี่ยนแปลง

ความเร็วมอเตอร์เปลี่ยนแปลงอย่างไร.....

LED เปลี่ยนแปลงอย่างไร P1..... P2.....

6. กดสวิตช์ SW1 ในโมดูลเบสิกอินพุต 10 ครั้ง สังเกตการเปลี่ยนแปลง

ความเร็วมอเตอร์เปลี่ยนแปลงอย่างไร.....

LED เปลี่ยนแปลงอย่างไร P1..... P2.....

7. จากโปรแกรม การเปลี่ยนแปลงค่าในรีจิสเตอร์ CCPR1L มีผลต่อการเปลี่ยนแปลง

ความเร็วของมอเตอร์อย่างไร จงอธิบาย

.....

.....

.....

สรุปผลการทดลอง

.....

.....

.....

.....

.....

.....

.....

.....

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำถามท้ายการทดลอง

1. จงอธิบายการทำงานของโปรแกรมที่ จ.47

2. จงอธิบายส่วนสำคัญของการเขียนโปรแกรม เพื่อควบคุมให้ PIC16F876 ทำงานเป็นวงจร PWM

3. จงยกตัวอย่างการนำวงจร PWM ที่มี PIC16F876 เป็นอุปกรณ์หลักไปใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 10
การเชื่อมต่อ PIC16F876 กับสเตปปิ้งมอเตอร์ (Stepping Motor)

จุดประสงค์

1. เพื่อให้สามารถเลือกอุปกรณ์ขับกระแสและแรงดันที่เหมาะสมได้
2. เพื่อให้สามารถใช้งาน PIC16F876 และอุปกรณ์ขับกระแสและแรงดันชนิดต่างๆได้
3. เพื่อให้สามารถเขียนโปรแกรมควบคุมการหมุนของสเตปปิ้งมอเตอร์ได้

อุปกรณ์ที่ใช้ในการทดลอง

1. โมดูลหลัก PIC-ICSP
2. โมดูลการขับสเตปปิ้งมอเตอร์
3. คอมพิวเตอร์พีซีที่ติดตั้งโปรแกรม MPLAB
4. สายเชื่อมต่อระหว่างโมดูลหลัก PIC-ICSP กับคอมพิวเตอร์พีซี
5. สายเชื่อมต่อพอร์ตระหว่างโมดูล

ทฤษฎีเบื้องต้น

เมื่อต้องการขับอุปกรณ์ที่ต้องใช้กระแสและแรงดันสูงๆเกินกว่าไมโครคอนโทรลเลอร์ PIC16F876 ขับได้โดยตรงดังนั้น ต้องมีการขับกระแสและแรงดันผ่านตัวอุปกรณ์ตัวหนึ่งที่เรียกว่าอุปกรณ์ไดรเวอร์ (Driver) ซึ่งอุปกรณ์ตัวนี้มีอยู่ 3 แบบดังนี้

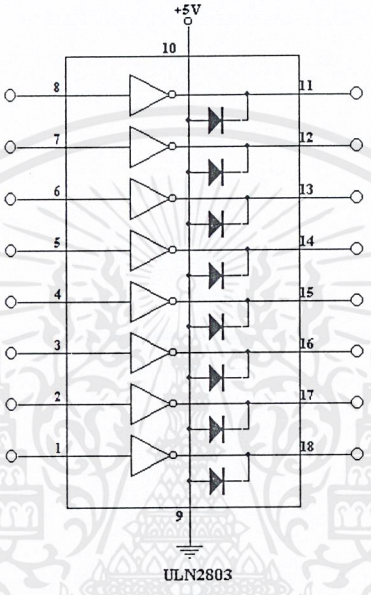
1. ใช้ไอซีขับ
2. ใช้ทรานซิสเตอร์ขับ
3. ใช้อุปกรณ์เชื่อมโยงทางแสง

การใช้ไอซีขับ

การใช้ไอซีขับโหลดที่ต้องการกระแสสูงส่วนใหญ่จะมีวงจรเอาต์พุตเป็นแบบคอลเล็กเตอร์เปิด ทำให้สามารถใช้กับแรงดันตั้งแต่ 30 โวลต์ ขึ้นไปซึ่งขึ้นอยู่กับไอซีแต่ละเบอร์ สำหรับไอซีขับหรือไอซีไดรเวอร์ที่จะอธิบายนี้คือ เบอร์ ULN2803 ซึ่งเป็นไอซีที่เราใช้ขับสเตปปิ้งมอเตอร์ในการทดลองนี้ โดยมีรายละเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ULN2803 เป็นไอซีไดรเวอร์ที่มีความสามารถสูง ภายในจะบรรจุอินเวอร์เตอร์เกตแบบคอลเล็กเตอร์เปิด 8 ตัว สามารถใช้แรงดันได้สูงสุดบวก 50 โวลต์ กระแสเอาต์พุตสูงสุดในแต่ละขาเท่ากับ 500 มิลลิแอมป์ นอกจากนั้นยังต่อไดโอดป้องกันไว้ทุกๆ เอาต์พุต ทำให้สามารถต่อโหลดที่เป็นขดลวดได้ทันที วงจรภายในและขาต่อใช้งานของไอซีไดรเวอร์เบอร์ ULN2803 เป็นดังรูปที่ จ.48



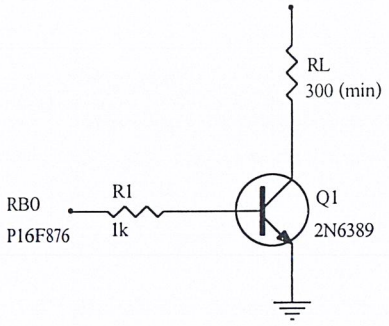
รูปที่ จ.48 การจัดขาและวงจรภายในของไอซี ULN2803

การใช้ทรานซิสเตอร์ขับ

การใช้ทรานซิสเตอร์ขับมีอยู่ 3 แบบ โดยใช้ทรานซิสเตอร์ที่มีความแตกต่างกันของกระแสและแรงดันเพื่อรองรับโหลดที่มีความต้องการกระแสและแรงดันแตกต่างกัน

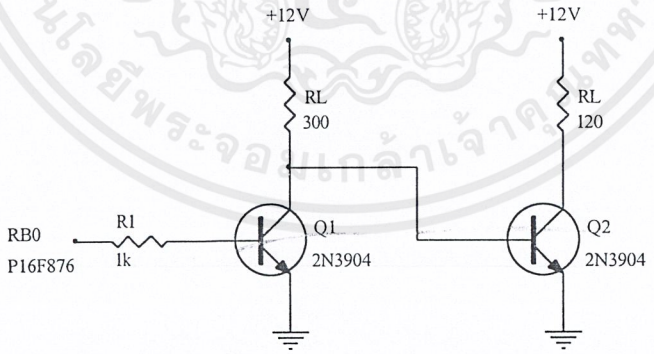
1. การใช้ทรานซิสเตอร์ขับแบบเดียว จะเหมาะกับโหลดที่ต้องการกระแสและแรงดันปานกลางตั้งแต่ 30-200 มิลลิแอมป์ เช่น รีเลย์ (Relay) หรือว่าจะเป็นมอเตอร์ไฟฟ้ากระแสตรงขนาดเล็กการต่อวงจรโดยใช้ทรานซิสเตอร์ขับแบบเดี่ยวเป็นดังรูปที่ จ.49 ซึ่งเป็นการต่อทรานซิสเตอร์เข้ากับพอร์ตเอาต์พุต RB0 ของไมโครคอนโทรลเลอร์โดยมีตัวต้านทาน R1 ทำหน้าที่จำกัดกระแสที่ไหลเข้าขาเบสของทรานซิสเตอร์ Q1 ทรานซิสเตอร์จะทำงานก็ต่อเมื่อที่ขา RB0 มีลอจิกเป็น “1” เมื่อทรานซิสเตอร์ทำงานจะเกิดกระแสไหลผ่าน RL กระแสโหลดสูงสุดเท่ากับ 40 มิลลิแอมป์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ จ.49 วงจรขับโหลดโดยใช้ทรานซิสเตอร์

2. การใช้ทรานซิสเตอร์ขับแบบคาสเคด จะมีอัตราการทนกระแสคอลเล็กเตอร์ที่สูงแต่การขับกระแสแบบนี้จะทำให้ความเร็วในการทำงานช้ากว่าทรานซิสเตอร์ที่มีกำลังงานต่ำๆ ในระบบไมโครคอนโทรลเลอร์ที่มีความเร็วในการทำงานสูงเมื่อต้องนำมาขับโหลดที่มีกำลังปานกลางจึงต้องใช้การขับแบบคาสเคดโดยเราจะให้ไมโครคอนโทรลเลอร์กระตุ้นทรานซิสเตอร์กำลังงานต่ำทำงานเพื่อไปขับให้ทรานซิสเตอร์กำลังงานปานกลางต่อไป ดังรูปที่ จ.50 เมื่อต้องการให้ทรานซิสเตอร์ Q1 ทำงานต้องป้อนลอจิก “1” ออกทางขา RB0 ของไมโครคอนโทรลเลอร์ ทำให้มีกระแสไหลผ่าน RL เข้าขาเบสของทรานซิสเตอร์ Q2 เพื่อกระตุ้นให้ Q2 ทำงานเกิดกระแสไหลผ่าน RL มีค่า 600 มิลลิแอมป์ ดังนั้นก็สามารถนำไปขับสเต็ปปิ้งมอเตอร์ขนาดเล็กได้

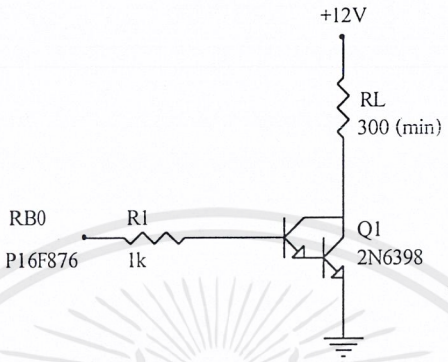


รูปที่ จ.50 วงจรขับโหลดโดยใช้ทรานซิสเตอร์ต่อแบบคาสเคด

3. การใช้ทรานซิสเตอร์แบบดาร์ลิงตัน เพื่อขับกระแสสูงๆ และมีความเร็วในการทำงานสูงด้วยการใช้ทรานซิสเตอร์แบบนี้จะมีทรานซิสเตอร์ 2 ตัวที่ต่อกันแบบดาร์ลิงตันภายในตัวถังเดียวกัน ทำให้ประหยัดอุปกรณ์และวงจรมีขนาดเล็กลง ดังรูปที่ จ.51 ทรานซิสเตอร์ Q1 ซึ่งเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทรานซิสเตอร์แบบดาร์ลิงตันสามารถขับกระแสได้สูงสุด 750 มิลลิแอมป์ โดยการต่อเข้ากับพอร์ตของไมโครคอนโทรลเลอร์ผ่านตัวจำกัดกระแสเพียงตัวเดียวทำให้มีความเร็วในการทำงานสูง นอกจากนั้นยังสามารขับกระแสได้สูงพอสมควร



รูปที่ จ.51 วงจรขับโหลดโดยใช้ทรานซิสเตอร์แบบดาร์ลิงตัน

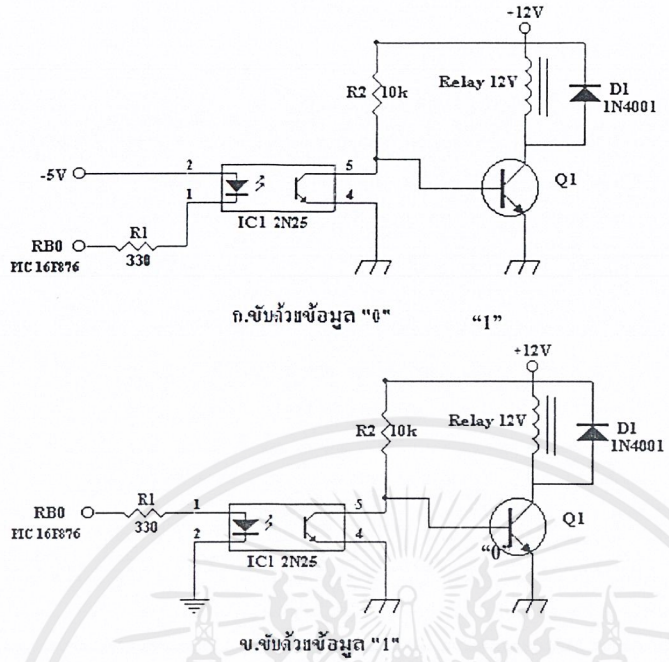
การขับโหลดโดยใช้ออปโต้คัปเปลอร์ (Opto-coupler)

ออปโต้คัปเปลอร์ มีการแบ่งส่วนอินพุตและส่วนเอาต์พุตออกจากกันทางไฟฟ้าการถ่ายทอดสัญญาณระหว่างอินพุตกับเอาต์พุตจะใช้การเชื่อมโยงทางแสงเท่านั้นทำให้กราวด์ของอินพุตและเอาต์พุตไม่เชื่อมต่อกัน จึงทำให้การกำหนดให้ทางอินพุตเป็นวงจรไฟฟ้ากระแสตรง และวงจรทางเอาต์พุตเป็นวงจรไฟฟ้ากระแสสลับ

จากรูปที่ จ.52 เป็นวงจรตัวอย่างของการนำไมโครคอนโทรลเลอร์ขับออปโต้คัปเปลอร์เพื่อทำการขับโหลดรีเลย์กำลังไฟฟ้าสูง จะเห็นได้ว่าการใช้ไมโครคอนโทรลเลอร์กระตุ้นให้ออปโต้คัปเปลอร์ทำงานทำได้ 2 แบบคือ ใช้ลอจิก “0” ดังรูปที่ จ.52 ก. และใช้ลอจิก “1” ดังรูปที่ จ.52 ข.สเตปปิ้งมอเตอร์เป็นมอเตอร์ที่มีลักษณะการทำงานแตกต่างจากมอเตอร์ทั่วไปเพราะจะต้องป้อนสัญญาณเป็นพัลส์ให้กับขดลวดเป็นจังหวะอย่างเหมาะสม และการหมุนของมอเตอร์ชนิดนี้จะหมุนเป็นจังหวะตามพัลส์ที่ป้อนเข้ามาจะไม่หมุนต่อเนื่องเหมือนกับมอเตอร์ธรรมดา ทำให้เราสามารถเลือกตำแหน่งที่ต้องการให้มอเตอร์หยุดหมุนได้ จังหวะการหมุนของสเตปปิ้งมอเตอร์เรียกว่า สเตป (Step) ความละเอียดของมอเตอร์กำหนดเป็นองศาต่อการหมุนไปในหนึ่งสเตป หากจำนวนองศาของมอเตอร์ต่อสเตปมาก หมายความว่า มอเตอร์ตัวนี้มีความละเอียดของการหมุนต่ำ

ขนาดของสเตปปิ้งมอเตอร์มีตั้งแต่ขนาดแรงดันต่ำ 3 โวลต์ ไปจนถึง 24 โวลต์ ส่วนขนาดของกระแสมีตั้งแต่ไม่กี่ 10 มิลลิแอมป์อันเป็นสเตปปิ้งมอเตอร์ตัวเล็กไปจนถึงเป็น 10 แอมป์ ซึ่งขนาดของมอเตอร์จะใหญ่ขึ้นตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



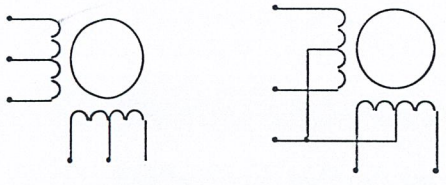
รูปที่ จ.52 วงจรการขับโหลดด้วยออปโต้คัปเปอร์

สเตปปิ้งมอเตอร์แบบยูนิโพล่าร์ (Uni-polar stepping motor)

สเตปปิ้งมอเตอร์ที่นิยมใช้มากที่สุดและหาได้ง่ายคือ สเตปปิ้งมอเตอร์แบบยูนิโพล่าร์มีลักษณะการพันขดลวดของมอเตอร์แสดงดังรูปที่ จ.53

สเตปปิ้งมอเตอร์แบบนี้มีการพันขดลวด 2 ขดบนแต่ละขั้วแม่เหล็กของสเตเตอร์ แต่ละขดลวดแบ่งเป็น 2 เฟส รวมมอเตอร์ทั้งตัวจะมี 4 เฟสคือ เฟส 1,2,3 และ 4 มีการต่อสายออกมาจากขดลวดแต่ละขดเพื่อจ่ายไฟเลี้ยง ทำให้สเตปปิ้งมอเตอร์แบบนี้มีทั้งแบบ 5 สายและ 6 สาย ถ้าเป็นแบบ 6 สายจะเป็นการนำสายไฟเลี้ยงของขดลวดทั้งสองมาต่อรวมกันเป็นสายเดียว

การกระตุ้นและควบคุมการหมุนของสเตปปิ้งมอเตอร์



รูปที่ จ.53 การพันขดลวดของสเตปปิ้งมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกระตุ้นและการควบคุมการหมุนของสเตปปิ้งมอเตอร์ให้เคลื่อนที่ไปในแต่ละสเตปทำได้โดยจ่ายกำลังไฟฟ้าไปยังขดลวดแต่ละขดบนสเตเตอร์ (Stater) ซึ่งต้องป้อนเป็นแบบซีเควนเชียล (Sequential) ในรูปแบบที่ถูกต้องด้วย สามารถแบ่งได้เป็น 3 รูปแบบคือ แบบเวฟ(Wave) , แบบ 2 เฟส (Two phase) และแบบครึ่งสเตป (Half wave)

แบบเวฟ เป็นการกระตุ้นที่มีรูปแบบที่ง่ายที่สุด โดยทำการกระตุ้นขดลวดทีละขดในเวลาหนึ่งไล่เรียงถัดกันไป เช่น เริ่มต้นที่ขดที่ 1, 2, 3 และ 4 แล้ววนกลับมาที่ 1 วนไปเรื่อยๆ หรือเริ่มที่ขดที่ 4 แล้วย้อนกลับไปขดที่ 3, 2 และ 1 วนกลับไปเรื่อยๆ ซึ่งทำให้ทิศทางของการหมุนสวนทางกัน ในการกระตุ้นรูปแบบนี้จึงมีขดลวดเพียงขดเดียวในเวลาหนึ่งที่ถูกกระตุ้นเท่านั้น วงจรกระตุ้นแบบเวฟจึงมีราคาถูกและทำการกระตุ้นได้ง่าย ขั้นตอนการทำงานต่างๆ ดังตารางที่ จ.8

ตารางที่ จ.8 ลำดับการทำงานของสเตปปิ้งมอเตอร์ได้รับการกระตุ้นแบบเวฟ

สเตปที่	เฟสที่ 1	เฟสที่ 2	เฟสที่ 3	เฟสที่ 4
1	ทำงาน	-	-	-
2	-	ทำงาน	-	-
3	-	-	ทำงาน	-
4	-	-	-	ทำงาน

แบบ 2 เฟส เป็นการกระตุ้นซึ่งคล้ายกับแบบเวฟแต่การกระตุ้นแบบนี้จะทำการกระตุ้นโดยจ่ายกำลังไฟฟ้าไปที่ขดลวด 2 ขดที่อยู่ใกล้กันในเวลาเดียวกัน และเรียงถัดกันไปเช่นเดียวกับแบบเวฟ เช่นขดลวดชุดแรกที่ถูกกระตุ้นจะเป็นขดที่ 1 และ 2 ตามด้วยการกระตุ้นขดที่ 2 และ 3 ต่อไปก็เป็นขดที่ 3 และ 4 ถัดไปก็เป็นขดที่ 4 และ 1 แล้วกลับมาที่ขดที่ 1 และ 2 วนไปเรื่อยๆ หรือเริ่มจากที่ขด 1 และ 4 ตามด้วยขดที่ 4 และ3 ต่อไปก็เป็นขดที่ 3 และ 2 ถัดไปก็เป็นขดที่ 2 และ 1 และวนกลับมาที่ 1 และ 4 วนไปเรื่อยๆ แต่การหมุนจะกลับทิศทางกัน การกระตุ้นแบบนี้สามารถเพิ่มแรงบิดได้มากกว่าแบบแรก โรเตอร์ (Roter) จะเคลื่อนที่ด้วยแรงดึงอย่างเต็มแรงด้วย 2 ขดลวดที่ถูกกระตุ้นพร้อมกัน และต่อไปด้วยแรงดึงจากอีก 2 ขดลวดถัดไป สำหรับข้อเสียในแบบนี้คือการกระตุ้นแบบนี้ต้องใช้กำลังไฟฟ้ามากขึ้น ขั้นตอนการทำงานต่างๆแสดงดังตารางที่ จ.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ จ.9 ลำดับการทำงานของสเตปปิ้งมอเตอร์ได้รับการกระตุ้นแบบสองเวฟ

สเตปที่	เฟสที่ 1	เฟสที่ 2	เฟสที่ 3	เฟสที่ 4
1	ทำงาน	ทำงาน	-	-
2	-	ทำงาน	ทำงาน	-
3	-	-	ทำงาน	ทำงาน
4	ทำงาน	-	-	ทำงาน

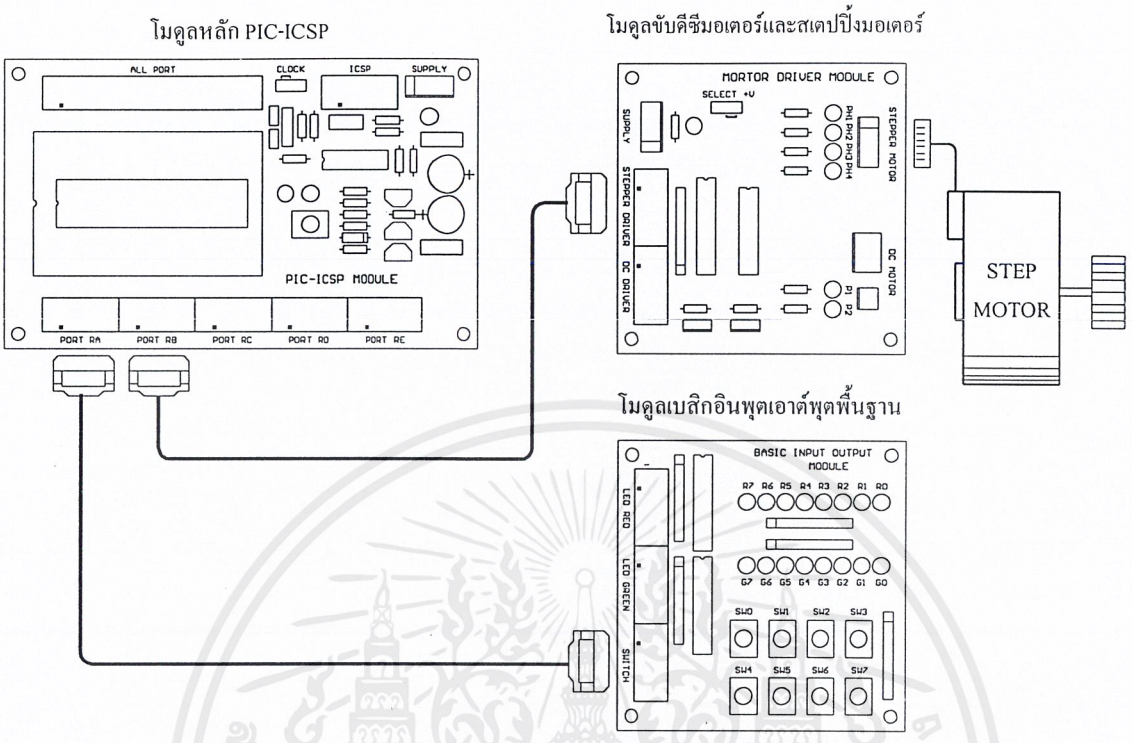
แบบครึ่งสเตป เป็นรูปแบบที่ผสมระหว่างแบบเวฟและแบบ 2 เฟส เพื่อเพิ่มจำนวนของสเตปต่อรอบอีกเท่าตัวหนึ่ง ในระบบนี้จะทำการกระตุ้นที่ขดลวดเรียงกันไปเป็นลำดับโดยเริ่มจากขดลวดที่ 1, 1 และ 2, 2, 2 และ 3, 3, 3 และ 4, 4, 4 และ 1 และวนกลับมายังขดที่ 1 แรงบิดที่ได้จากการกระตุ้นแบบนี้จะเพิ่มมากขึ้นอีก เพราะช่วงสเตปมีระยะสั้นลง แต่ละสเตปเกิดแรงดึงจากขดลวด 2 ขดที่ถูกกระตุ้นพร้อมกัน ความถูกต้องของตำแหน่งมีเพิ่มมากขึ้น แต่ต้องระวังไว้อีกอย่างหนึ่งว่าเมื่อกระตุ้นให้ทำงานในรูปแบบนี้จะต้องทำการหมุนถึง 2 สเตป จึงจะได้เท่ากับระยะเท่ากับ 1 สเตปเต็มของการควบคุมใน 2 แบบแรก สำหรับแหล่งจ่ายกำลังไฟฟ้าต้องใช้ขนาดเท่ากับแบบ 2 เฟสเป็นอย่างน้อยจึงจะเพียงพอ ขั้นตอนการทำงานต่างๆแสดงดังตารางที่ จ.10

ในการทดลองนี้เราจะกล่าวถึงเฉพาะสเตปปิ้งมอเตอร์แบบเวฟ และแบบ 2 เฟส ซึ่งแบบครึ่ง สเตปนั้นก็จะคล้ายๆ กับ 2 แบบแรกรวมกัน ข้อมูลในการป้อนก็จะคล้ายๆกัน

ตารางที่ จ.10 ลำดับการทำงานของสเตปปิ้งมอเตอร์ได้รับการกระตุ้นแบบครึ่งสเตป

สเตปที่	เฟสที่ 1	เฟสที่ 2	เฟสที่ 3	เฟสที่ 4
1	ทำงาน	-	-	-
2	ทำงาน	ทำงาน	-	-
3	-	ทำงาน	-	-
4	-	ทำงาน	ทำงาน	-
5	-	-	ทำงาน	-
6	-	-	ทำงาน	ทำงาน
7	-	-	-	ทำงาน
8	ทำงาน	-	-	ทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ จ.54 การเชื่อมต่อระหว่างโมดูลหลัก PIC-ICSP เข้ากับโมดูลการขับมอเตอร์

```
            List p=16f876
            include "p16f876.inc"
            __config _WDT_OFF & _PWRTE_ON & _LVP_OFF & _XT_OSC
COUNT1      equ     0x20
COUNT2      equ     0x21
DATA1       equ     0x23

            org     0x00
            bsf     STATUS,RP0
            movlw   0x3F
            movwf   PORTA
            movlw   0x00
            movwf   PORTB
            bcf     STATUS,RP0
            bcf     STATUS,C
            clrf    PORTB

Sw1         btfsc   PORTA,0
            goto    Sw2
Again1      movlw   0x01
            movwf   DATA1
Loop1       movf    DATA1,W
            movwf   PORTB
            call    Delay
            btfsc   PORTB,3
            goto    Again1
            bcf     STATUS,C
            rlf     DATA1,F
            btfss   PORTA,1
            goto    Sw2
            goto    Loop1
Sw2         btfsc   PORTA,1
            goto    Sw1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Again2      movlw   0x08
            movwf   DATA1
Loop2       movf    DATA1,W
            movwf   PORTB
            call    Delay
            bcf     STATUS,C
            rrf     DATA1,F
            btfss   PORTA,W
            goto    Sw1
            btfss   STATUS,C
            goto    Loop2
            goto    Again2
;------------------------------------ DELAY TIME
Delay       movlw   0x20
            movwf   COUNT1
Delay1      movlw   0xFF
            movwf   COUNT2
Delay2      decfsz  COUNT2,1
            goto    Delay2
            decfsz  COUNT1,1
            goto    Delay1
            return
            end
```

รูปที่ จ.55 การขับสเตปปิ้งมอเตอร์

```
            List p=16f876
            include "p16f876.inc"
            __config _WDT_OFF & _PWRTE_ON & _LVP_OFF & _XT_OSC
COUNT1      equ     0x20
COUNT2      equ     0x21
DATA1       equ     0x23
            org     0x00
            bsf     STATUS,RP0
            movlw   0x3F
            movwf   PORTA
            movlw   0x00
            movwf   PORTB
            bcf     STATUS,RP0
            bcf     STATUS,C
            clrf    PORTB
Sw1         btfsc   PORTA,0
            goto    Sw2
Again1      clrf    COUNT1,F
Loop1       movf    COUNT1,W
            call    Table1
            bcf     STATUS,Z
            andlw   0xFF
            btfsc   STATUS,Z
            goto    Int01
            goto    Again1
Int01       movwf   PORTB
            incf    COUNT1,F
            goto    Loop1
Sw2         btfsc   PORTA,1
            goto    Sw1
Again2      clrf    COUNT2,F
Loop2       movf    COUNT2,W
            call    Table2
            bcf     STATUS,Z
            andlw   0xFF
            btfsc   STATUS,Z
            goto    Int02
            goto    Again2
Int02       movwf   PORTB
            incf    COUNT2,F
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            goto    Loop2
;------------------------------------ Table Data
Table1      addwf   PCL,F
            dt      03, 06, 0c, 09, 00
Table2      addwf   PCH,F
            dt      03, 06, 0c, 09, 00
;------------------------------------ DELAY TIME
Delay       movlw   0x20
            movwf   COUNT3
Delay1      movlw   0xFF
            movwf   COUNT4
Delay2      decfsz  COUNT4,1
            goto    Delay2
            decfsz  COUNT3,1
            goto    Delay1
            return
            end
```

รูปที่ จ.56 โปรแกรมขับสเตปปิ้งมอเตอร์แบบ 2 เฟส

ลำดับขั้นตอนการทดลอง

1. เขียนโปรแกรมดังรูปที่ จ.55 ทำการแอสเซมเบลอร์ แล้วเขียนข้อมูลลงบน PIC16F876
2. ต่อสายเชื่อมต่อระหว่างโมดูลตามรูปที่ จ.54
3. ทำการรันโปรแกรมบน PIC16F876 โดยการกดสวิตช์รีเซต สังเกตผลที่เกิดขึ้น
4. เมื่อกดสวิตช์ SW1 ผลที่ได้คือ

.....

.....

5. เมื่อกดสวิตช์ SW2 ผลที่ได้คือ

.....

.....

6. เขียนโปรแกรมรูปที่ จ.56 ทำการแอสเซมเบลอร์ จากนั้นเขียนข้อมูลลงบน PIC16F876
7. ต่อสายเชื่อมต่อระหว่างโมดูลตามรูปที่ จ.54
8. ทำการรันโปรแกรมบน PIC16F876 โดยการกดสวิตช์รีเซต สังเกตผลที่เกิดขึ้น
9. เมื่อกดสวิตช์ SW1 ผลที่ได้คือ

.....

.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. เมื่อกดสวิตช์ SW2 ผลที่ได้คือ

.....

.....

11. เมื่อไม่มีการกดสวิตช์ใดๆ

.....

.....

สรุปผลการทดลอง

.....

.....

.....

.....

.....

.....

.....

.....

.....

คำถามท้ายการทดลอง

1. จงอธิบายการทำงานของโปรแกรมรูปที่ จ.48 มาพอเข้าใจ

2. จากโปรแกรมรูปที่ จ.48 เป็นการกระตุ้นและควบคุมการหมุนสเตปปิ้งมอเตอร์แบบเวฟ เพราะเหตุใด

3. จงเขียนโปรแกรมควบคุมเสต็ปมอเตอร์แบบครึ่งสเตปโดยมีระดับความเร็วของมอเตอร์ 4 ระดับคือ กด SW1 ให้หมุนช้า กด SW2 ให้หมุนเร็ว กด SW3 ให้หมุนเร็วขึ้น และกด SW4 จะหมุนเร็วที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 11

การใช้งาน PIC16F876 แปลงสัญญาณแอนะลอกเป็นดิจิตอล

จุดมุ่งหมาย

1. เพื่อให้สามารถอธิบายหลักการแปลงสัญญาณแอนะลอกเป็นดิจิตอลได้
2. เพื่อให้สามารถใช้งาน PIC16F876 ร่วมกับวงจรแปลงสัญญาณแอนะลอกเป็นดิจิตอลในแบบต่างๆ ได้
3. เพื่อให้สามารถใช้งาน PIC16F876 ร่วมกับไอซีแปลงสัญญาณแอนะลอกเป็นดิจิตอลแบบอนุกรมได้

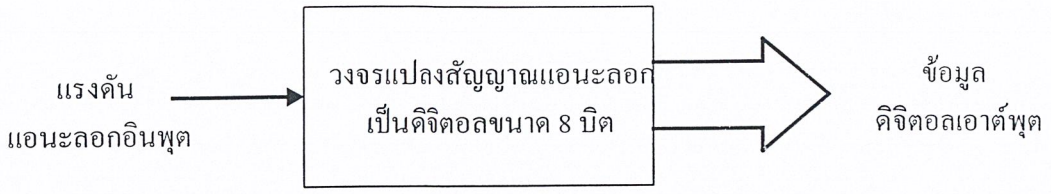
อุปกรณ์ที่ใช้ในการทดลอง

1. โมดูลหลัก PIC-ICSP
2. โมดูลเบสิกอินพุตเอาต์พุต
3. โมดูลแปลงสัญญาณแอนะลอกและดิจิตอล
4. โมดูลแสดงผลตัวเลข 7 ส่วน
5. คอมพิวเตอร์พีซีที่ติดตั้งโปรแกรม MPLAB
6. สายเชื่อมต่อระหว่างโมดูลหลัก PIC-ICSP และคอมพิวเตอร์พีซีสายเชื่อมต่อพอร์ตระหว่างโมดูล

ทฤษฎีเบื้องต้น

กระบวนการแปลงสัญญาณแอนะลอกเป็นดิจิตอลเป้นการแสดงความสัมพันธ์ระหว่างปริมาณของสัญญาณไฟฟ้าที่เป็นสัญญาณแอนะลอกกับข้อมูลตัวเลขที่ใช้แทนสัญญาณดิจิตอลในรูปที่ จ.57 เป็นแผนผังที่แสดงให้เห็นถึงอินพุตและเอาต์พุตของวงจรแปลงสัญญาณแอนะลอกเป็นดิจิตอล ความแม่นยำของการแปลงจะขึ้นอยู่กับจำนวนบิตของข้อมูลดิจิตอล วงจรแปลงสัญญาณแอนะลอกเป็นดิจิตอลหรือวงจร ADC ขนาด n บิต จะเกิดข้อมูลดิจิตอลจำนวน 2^n ข้อมูล เช่น วงจร ADC 3 บิต ก็จะเกิดข้อมูลดิจิตอลทางเอาต์พุตทั้งสิ้น 8 ข้อมูลระยะห่างของข้อมูลแต่ละข้อมูลจะเป็นตัวกำหนดความแม่นยำของการแปลงสัญญาณซึ่งกระบวนการที่ทำหน้าที่ตีความระดับสัญญาณแอนะลอกว่าตรงกับระดับข้อมูลดิจิตอลใด เรียกว่า กระบวนการควอนไทซิ่ง (Quantizing)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ จ.57 แผนผังการทำงานอย่างง่ายของวงจรแปลงสัญญาณแอนะลอกเป็นดิจิตอล

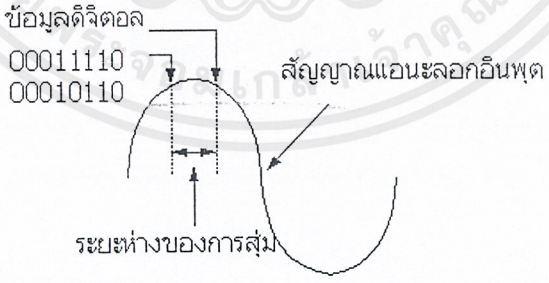
ระยะห่างของระดับข้อมูลดิจิตอลในวงจร ADC สามารถคำนวณได้จากความสัมพันธ์ทางคณิตศาสตร์ดังนี้

$$\text{ระยะห่างของระดับแรงดัน} = \text{VLSB} = V_{Fs} / 2^n \qquad (16.1)$$

เมื่อ V_{Fs} = แรงดันเต็มสเกลหรือแรงดันสูงสุดที่สามารถเกิดขึ้นได้ในวงจร ADC ปกติมีค่าเท่ากับไฟเลี้ยง

การแปลงสัญญาณแอนะลอกเป็นดิจิตอล

การแปลงสัญญาณแอนะลอกเป็นดิจิตอล สัญญาณจะถูกแปลงเป็นจำนวนดิจิตอล โดยการสุ่มหรือแซมปลิ้ง (Sampling) ดังรูปที่ จ.58 ถ้าสมมติมีเอาต์พุต 8 เส้น โดยเอาต์พุตแต่ละเส้นแสดงสถานะทางลอจิกเป็น 0 หรือ 1 จะมีความแตกต่างทางรหัสไบนารี (Binary) ทั้งหมด 2^8 หรือ 256 รหัส



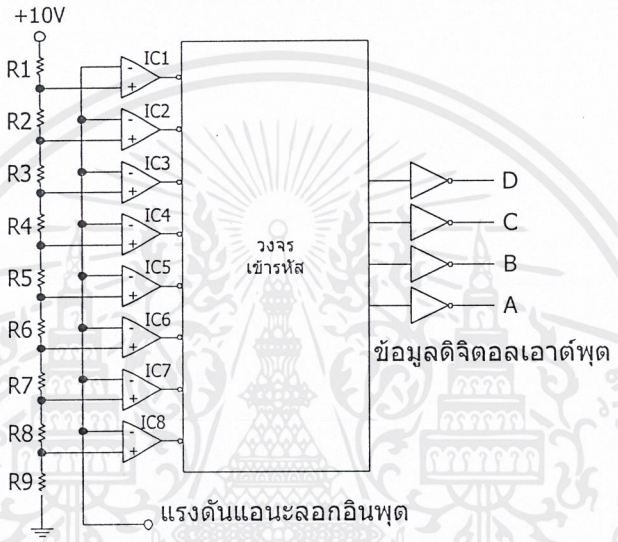
รูปที่ จ.58 การสุ่มสัญญาณแอนะลอกเพื่อกำหนดข้อมูลดิจิตอล

การแปลงสัญญาณแอนะลอกเป็นดิจิตอลแบบต่างๆ มีดังนี้

1. แบบเปรียบเทียบแรงดัน (Voltage Comparator ADC) มีหลักการคือ ใช้การเปรียบเทียบแรงดันระหว่างอินพุตกับแรงดันอ้างอิง จากนั้นวงจรเปรียบเทียบแรงดันจะส่งสัญญาณสูงและต่ำซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แทนผลการเปรียบเทียบเข้าสู่วงจรเข้ารหัส เพื่อกำหนดข้อมูลดิจิตอลทางเอาต์พุตต่อไปดังตัวอย่างตามรูปที่ จ.59 ซึ่งตัวต้านทาน R1-R9 กำหนดแรงดันอ้างอิงให้แก่ออปแอมป์ (Op-amp) IC1-IC8 ซึ่งทำหน้าที่เปรียบเทียบแรงดันอินพุตกับแรงดันอ้างอิง หากแรงดันอินพุตมีค่ามากกว่าแรงดันอ้างอิงที่จุดใดออปแอมป์ตัวนั้นก็จะทำงาน ให้เอาต์พุตเป็นแรงดันต่ำส่งไปยังวงจรเข้ารหัส ในทางตรงข้ามหากแรงดันอินพุตน้อยกว่าแรงดันอ้างอิงออปแอมป์จะไม่ทำงาน



รูปที่ จ.59 วงจรแปลงสัญญาณแอนะลอกเป็นดิจิตอลแบบเปรียบเทียบแรงดัน

2. แบบวงจรนับแรมป์ (Simple Counter-Ramp ADC) จะมีแผนผังแสดงการทำงานดังรูปที่ จ.60 ซึ่งแบ่งเป็น 4 ส่วนคือ

2.1 วงจรกำเนิดสัญญาณนาฬิกาและควบคุมลอจิก (Clock & Control logic) ส่วนนี้จะได้รับสัญญาณให้เริ่มทำงาน แล้วส่งสัญญาณนาฬิกาไปยังวงจรนับ

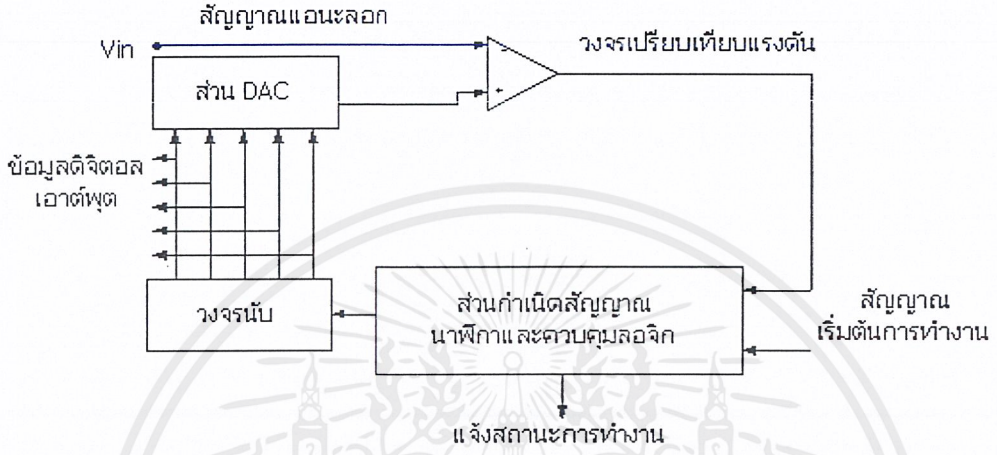
2.2 วงจรนับ (Counter) ส่วนนี้จะทำการนับค่าตามสัญญาณนาฬิกาที่ส่งเข้ามาแล้วส่งต่อออกไปยังวงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอก

2.3 วงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอก (DAC) ในส่วนนี้จะแปลงค่าของการนับจากวงจรนับเป็นแรงดันไฟตรง เพื่อส่งออกไปยังวงจรเปรียบเทียบแรงดัน

2.4 วงจรเปรียบเทียบแรงดัน (Comparator) จะทำหน้าที่เปรียบเทียบแรงดันจากวงจร DACกับแรงดันแอนะลอกที่ส่งเข้ามาทางอินพุตของวงจร (Vin) หากยังไม่เท่ากันวงจรเปรียบเทียบแรงดันจะส่งสัญญาณไปยังวงจรกำเนิดสัญญาณนาฬิกาและส่วนควบคุมลอจิกให้ทำงานต่อไป เพื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระตุ้นให้วงจรนับ และวงจร DAC ทำงานจนกระทั่ง แรงดันจากวงจร DAC มีค่าเท่ากับแรงดันอินพุต วงจรเปรียบเทียบจะทำการส่งสัญญาณควบคุมให้วงจรทั้งหมดหยุดทำงาน ค่าของวงจรนับครั้งสุดท้ายคือผลลัพธ์จากวงจร ADC



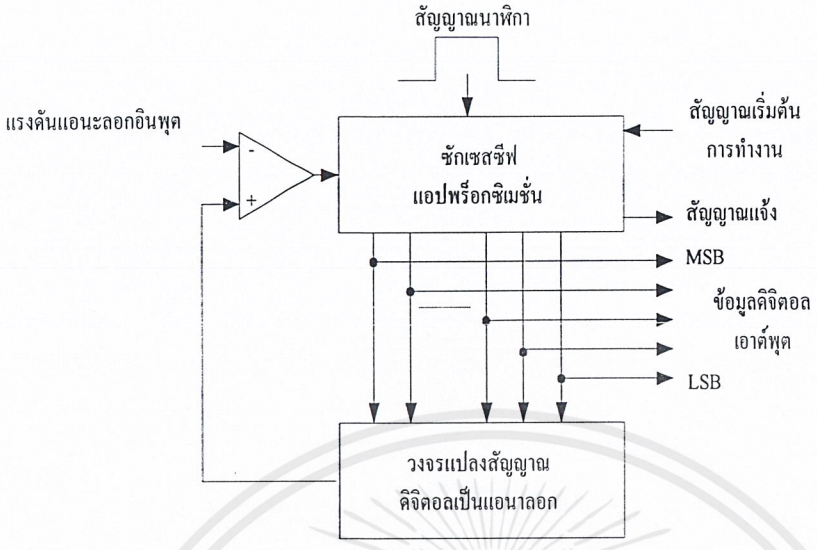
รูปที่ จ.60 แผนผังแสดงการทำงานของวงจร ADC แบบวงจรนับแรมป์

การแปลงสัญญาณแบบนี้มีข้อด้อยคือ มีความเร็วต่ำ เวลาที่ใช้ในการแปลงสัญญาณจะขึ้นอยู่กับปริมาณของแรงดันอินพุต หากแรงดันอินพุตมีค่ามาก เวลาที่ใช้ก็จะมากตาม

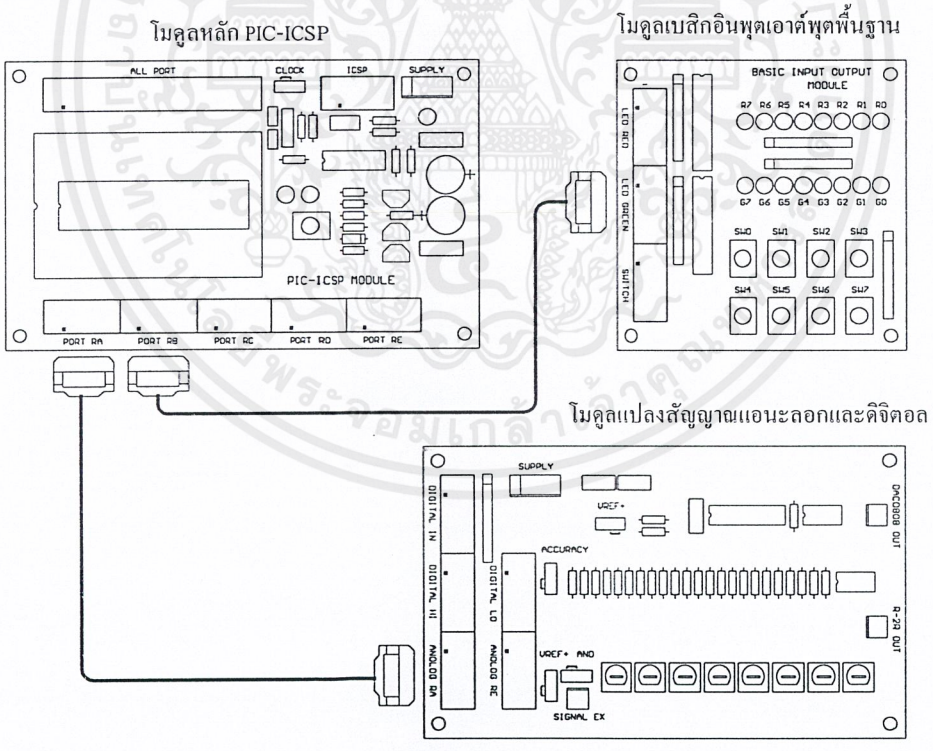
3. แบบซักเซสซีฟแอปพร็อกซิเมชั่น (Successive approximation ADC) หรือการแปลงแบบประมาณค่าใกล้เคียงดังรูปที่ จ.61 ส่วนที่สำคัญจะต้องมีวงจรเปรียบเทียบแรงดัน, สัญญาณนาฬิกา และส่วนควบคุมลอกจิก ซึ่งจะเหมือนกับแบบวงจรนับแรมป์แต่แบบซักเซสซีฟแอปพร็อกซิเมชั่นจะใช้รีจิสเตอร์ฐานสองแทนวงจรนับเลขฐานสอง แต่ละบิตของรีจิสเตอร์จะเซตหรือรีเซตโดยการควบคุมจากวงจรควบคุม

ความเที่ยงตรงของวงจร ADC เป็นการเปรียบเทียบแรงดันแอนะลอกของวงจร ADC กับแรงดันที่ควรจะเกิดขึ้นจริง ค่าเวลาในการแปลงสัญญาณ (Conversion time) เป็นค่าของเวลาทั้งหมดที่วงจร ADC แบบวงจรนับแรมป์และแบบซักเซสซีฟแอปพร็อกซิเมชั่นใช้ในการแปลงสัญญาณแอนะลอกเป็นดิจิตอลจนเสร็จสิ้น พารามิเตอร์ตัวนี้มักจะปรากฏในคุณสมบัติของไอซีที่ทำงานเป็นวงจร ADC เมื่อไอซีทำงานเสร็จสิ้นจะส่งสัญญาณที่เรียกว่า EOC (End of conversion) ซึ่งค่าเวลาในการแปลงสัญญาณของวงจร ADC จะขึ้นอยู่กับจำนวนบิตของวงจร, ค่าความถี่ของสัญญาณนาฬิกาที่ใช้ในการแปลงสัญญาณ และขนาดของสัญญาณแอนะลอกอินพุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

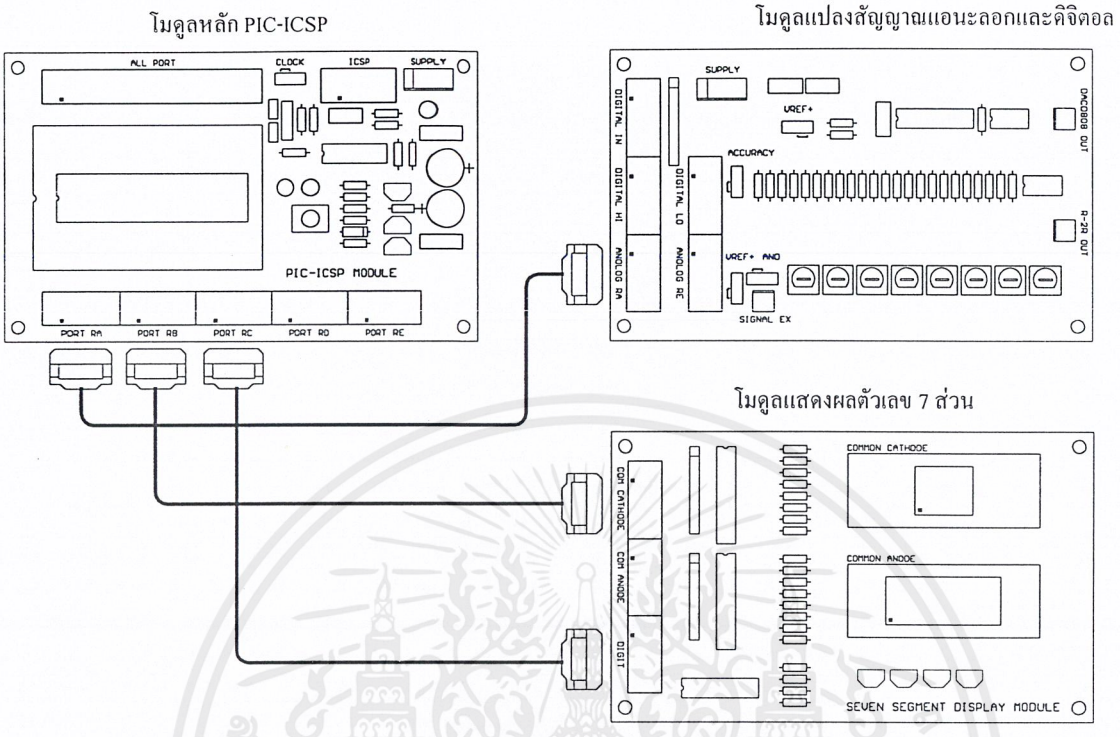


รูปที่ จ.61 แผนผังแสดงการทำงานของวงจร ADC แบบชักเซสซีฟแอปพร็อกซิเมชั่น



รูปที่ จ.62 การเชื่อมต่อโมดูลแปลงสัญญาณแอนะลอกเป็นดิจิตอลแสดงผลออกทาง LED

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ จ.63 การเชื่อมต่อโมดูลแปลงสัญญาณ แสดงผลเป็นตัวเลขที่ LED 7 ส่วน

```
     list p=16f877
     include "p16f877.inc"
   __config _WDT_OFF & _PWRTE_ON & _LVP_OFF & _XT_OSC
              org     0x00
              bsf     STATUS,RP0
              movlw   0x0e
              movwf   ADCON1          ; Left Justified, Analog Bit0
              movlw   0xff
              movwf   TRISA
              clrf    TRISB
              bcf     STATUS,RP0
              movlw   b'01000001'     ; Fosc/8, Channel0, A/D On
              movwf   ADCON0
              clrf    PORTB
Start         btfsc   ADCON0,GO_DONE ; check DONE
              goto    Start
              movf    ADRESH,W        ; read adc result
              movwf   PORTB           ; mov to LED display
              bsf     ADCON0,GO_DONE ; set GO
              goto    Start
              end
```

รูปที่ จ.64 โปรแกรมแปลงสัญญาณแอนะลอกเป็นดิจิตอล แสดงผลทาง LED ขนาด 8 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            list p=16f876
            include "p16f876.inc"
            __config _WDT_OFF & _PWRTE_ON & _LVP_OFF & _XT_OSC
B_ADRESH    equ     0x20
B_ADRESL    equ     0x21
Digit1      equ     0x22
Digit2      equ     0x23
Digit3      equ     0x24
Digit4      equ     0x25
B_Delay1    equ     0x26
Refresh1    equ     0x27
Refresh2    equ     0x28

            org     0x00
IniPort     bsf     STATUS,RP0
            movlw   0x0e
            movwf   ADCON1          ; แอนะลอกบิต 0
movlw       0xff
            movwf   TRISA
            clrf    TRISB
            clrf    TRISC
            bcf     STATUS,RP0
            movlw   b'01000001'     ; สัญญาณนาฬิกา = Fosc/8, เลือกแชนเนล 0 และ A/D เปิด
            movwf   ADCON0
            clrf    PORTB
            clrf    PORTC
Start       call    ADConvt
            call    BinToBCD
            movlw   0x50
            movwf   Refresh1
            movlw   0xff
Refrsh1     movwf   Refresh2
            call    Display
Refrsh2     decfsz  Refresh2,F
            goto    Refrsh2
            decfsz  Refresh1,F
            goto    Refrsh1
            goto    Start

;___________________________________การแปลง A to D
ADConvt     btfsc   ADCON0,2
            goto    ADConvt
            bsf     STATUS,RP0
            movf    ADRESL,W
            bcf     STATUS,RP0
            movwf   B_ADRESL
            movf    ADRESH,W
            movwf   B_ADRESH
            bsf     ADCON0,2
            return

;___________________________________แสดงผลบน LED 7 segment
Display
            clrf    PORTB
            clrf    PORTC
            movlw   Digit1
            movwf   FSR
DP1         movf    INDF,W          ; หลักที่ 1
            call    Table7Seg

            movwf   PORTB
            call    Delay
            clrf    PORTB
            incf    FSR,F
            incf    PORTC,F
            btfss   PORTC,2
            goto    DP1
            clrf    PORTC
            return
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Table7Seg   addwf   PCL,F
            dt      3f, 06, 5b, 4f, 66, 6d, 7d, 07
            dt      7f, 6f, 00, 40
            dt      0bf, 086, 0db, 0cf, 0e6, 0ed, 0fd, 087
            dt      0ff, 0ef, 080
;_____________________________________หน่วงเวลา
Delay       movlw   0x80
            movwf   B_Delay1
Dly0        decfsz  B_Delay1,F
            goto    Dly0
            return
            end
```

รูปที่ จ.65 โปรแกรมแปลงสัญญาณแอนะลอกเป็นดิจิตอล 10 บิต แสดงผลเป็นเลข 0 ถึง 1023

ลำดับขั้นตอนการทดลอง

1. เขียนโปรแกรมดังรูปที่ จ.64 ทำการแอสเซมเบลอร์แล้วเขียนข้อมูลลงบน PIC 16F876
2. ต่อสายเชื่อมต่อตามรูปที่ จ.62
3. ทำการรันโปรแกรม โดยการคลิกปุ่ม RUN
4. ปรับตัวต้านทาน AN0 ไปที่ตำแหน่งกึ่งกลาง
5. สังเกตผลที่ได้เป็นดังนี้

.....

.....

.....

6. ปรับตัวต้านทาน AN0 ไปที่ตำแหน่งซ้ายสุด
7. สังเกตผลที่ได้เป็นดังนี้

.....

.....

8. ปรับตัวต้านทาน AN0 ไปที่ตำแหน่งขวาสุด
9. สังเกตผลที่ได้เป็นดังนี้

.....

.....

10. เขียนโปรแกรมดังรูปที่ จ.65 ทำการแอสเซมเบลอร์แล้วเขียนข้อมูลลงบน PIC 16F876
11. ปรับตัวต้านทาน AN0 ไปที่ตำแหน่งกึ่งกลางเช่นเดียวกับข้อ 4.
12. สังเกตผลที่ได้เมื่อเทียบกับข้อ 4. เป็นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

.....

.....

13. ปรับตัวต้านทาน AN0 ไปที่ตำแหน่งซ้ายสุดเช่นเดียวกับข้อ 6.

14. สังเกตผลที่ได้เมื่อเทียบกับข้อ 6. เป็นดังนี้

.....

.....

15. ปรับตัวต้านทาน AN0 ไปที่ตำแหน่งซ้ายสุดเช่นเดียวกับข้อ 8.

16. สังเกตผลที่ได้เมื่อเทียบกับข้อ 8. เป็นดังนี้

.....

.....

สรุปผลการทดลอง

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

คำถามท้ายการทดลอง

1. จงอธิบายหลักการแปลงสัญญาณแอนะลอกเป็นดิจิตอลทั้ง 3 แบบมาพอเข้าใจ
2. จงอธิบายหลักการทำงานโปรแกรมดังรูปที่ จ.64
3. จงเขียนโปรแกรมเพื่อแปลงสัญญาณแอนะลอกเป็นดิจิตอลแบบ 8 บิต แสดงผลทาง LED 7 segment

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 12
การใช้งาน PIC16F876 กับวงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอก

จุดประสงค์

1. เพื่อให้สามารถอธิบายหลักการแปลงสัญญาณดิจิตอลเป็นแอนะลอกได้
2. เพื่อให้สามารถอธิบายความแตกต่างของวงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอกชนิดต่างๆได้
3. เพื่อให้สามารถใช้งาน PIC16F876 กับวงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอกแบบต่างๆ ได้

อุปกรณ์ที่ใช้ในการทดลอง

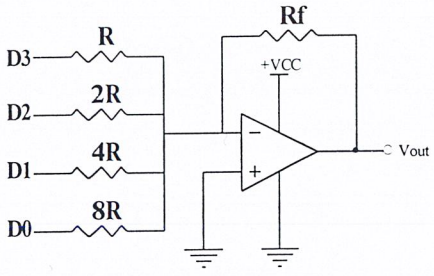
1. โมดูลหลัก PIC-ICSP
2. โมดูลแปลงสัญญาณดิจิตอลและแอนะลอก
3. โมดูลเครื่องมือวัด
4. คอมพิวเตอร์พีซีที่ติดตั้งโปรแกรม MPLAB
5. สายเชื่อมต่อระหว่างโมดูลหลัก PIC-ICSP และคอมพิวเตอร์พีซี
6. สายเชื่อมต่อพอร์ตระหว่างโมดูล

ทฤษฎีเบื้องต้น

การแปลงสัญญาณดิจิตอลเป็นแอนะลอก

ในการนำไมโครคอนโทรลเลอร์ PIC16F876 ไปเชื่อมต่อกับอุปกรณ์ภายนอกเราจะต้องสามารถพัฒนาระบบไมโครคอนโทรลเลอร์ให้สามารถทำงานร่วมกับอุปกรณ์ต่างๆ ได้ทุกรูปแบบและต้องสามารถทำงานกับสัญญาณดิจิตอลและแอนะลอกได้เป็นอย่างดี วงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอกจึงมีความจำเป็นอย่างมากในระบบควบคุม เช่น การแปลงสัญญาณดิจิตอลเป็นแอนะลอกที่สามารถทำความเข้าใจได้ง่ายที่สุดคือวิธีการที่เรียกว่า การกำหนดน้ำหนักตัวต้านทาน (weighted resistor) ดังแสดงตามรูปที่ จ.66 ซึ่งก็คือวงจรรวมสัญญาณ (summing amplifier) นั่นเอง ซึ่งเป็นการแปลงสัญญาณดิจิตอลเป็นแอนะลอกขนาด 4 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ จ.66 วงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอกแบบกำหนดน้ำหนักตัวต้านทาน

ตัวต้านทาน 4 ตัวที่ต่ออนุกรมอินพุตจะมีค่าแตกต่างกันตามสัดส่วนที่กำหนดไว้คือ 1,2,4 และ 8 เช่น ถ้า R = 10 kΩ ตัวต้านทานตัวต่อไปต้องมีค่า 20 kΩ, 40 kΩ และ 80 kΩ เป็นต้น

แรงดันแอนะลอกทางเอาต์พุตสามารถหาได้จากความสัมพันธ์

$$V_{out} = -(R_f / R_i) \times V_{in} \qquad (17.1)$$

โดยที่ R_i คือค่าของตัวต้านทานอินพุต

V_{in} คือแรงดันอินพุทซึ่งมีด้วยกัน 2 ค่า คือ 0 และ 5 โวลต์

วงจร DAC อีกแบบหนึ่งที่ได้รับความนิยมคือแบบ R-2R ladder ทั้งนี้เนื่องจากสามารถหาอุปกรณ์ในการสร้างวงจรง่าย หมายความว่าหากต่อพ่วงวงจรตัวต้านทานเข้าไปอีกหลายสาขาก็จะทำให้กระแสถูกเฉลี่ยมากขึ้น เหมือนกับการสร้างบันไดของกระแสทีละขั้น ด้วยอัตราส่วนของตัวต้านทานที่คงที่คือ R และ 2R จากนั้นนำวงจร R-2R ladder นี้ต่อเข้ากับวงจรแปลงค่ากระแสเป็นแรงดันโดยใช้ออปแอมป์ ดังในรูปที่ จ.67

การคำนวณหาค่า V_{out} เป็นดังนี้

$$V_{out} = -I_{in} \times R_f \qquad (17.2)$$

$$V_{out} = -\left(\frac{D3}{2} + \frac{D2}{4} + \frac{D1}{8} + \frac{D0}{16}\right) \times \frac{V_{in}}{R} R_f \qquad (17.3)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ Vin คือ ค่าแรงดันสูงสุดของอินพุต

Iin คือ ผลรวมของกระแสที่ได้จากวงจร R-2R ladder

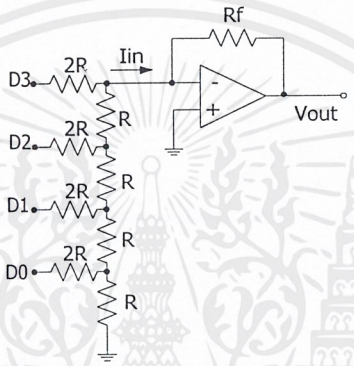
ตัวอย่างการคำนวณ

เมื่อสัญญาณอินพุต D3-D0 เป็น 1111 แรงดันอินพุตมีขนาด 5 โวลต์ Rf = R

Vout = -(0.5+0.25+0.125+0.0625) x 5 V

= -(0.9375) x 5V

= -4.6875 V



รูปที่ จ.67 พื้นฐานของวงจร R-2R ladder

จะเห็นว่า การใช้วงจร DAC แบบ R-2R ladder จะมีข้อดีคือ หาตัวต้านทานได้ง่ายกว่าและกำหนดน้ำหนักตัวต้านทานคือ ใช้ตัวต้านทานเพียง 2 ค่าในขณะที่แบบกำหนดน้ำหนักตัวต้านทานต้องใช้ตัวต้านทานหลายค่า และต้องมีค่าทวีคูณกันไปเรื่อย ๆ ซึ่งอาจทำให้หาตัวต้านทานได้ยากขึ้นตามลำดับ

การใช้งาน PIC16F876 กับวงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอกแบบ R-2R ladder

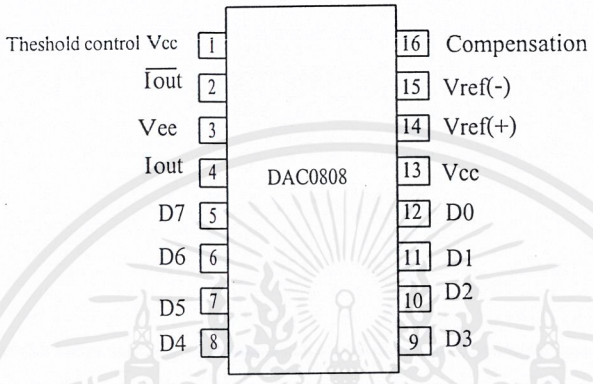
การต่อวงจร DAC แบบ R-2R ladder จะใช้พอร์ต B ทั้ง 8 บิตเป็นขาเอาต์พุตสำหรับส่งสัญญาณดิจิตอลเข้าสู่วงจร R-2R ladderเพื่อทำการแปลงสัญญาณเป็นแอนะลอกทางเอาต์พุต Vout

เนื่องจากข้อมูลที่ส่งออกมาจากไมโครคอนโทรลเลอร์ PIC16F876 มีขนาด 8 บิตจึงทำให้เกิดค่าการเปลี่ยนแปลง 2^8 เท่ากับ 256 ค่า ถ้าค่าแรงดันเอาต์พุตสูงสุดบวก 5 โวลต์ ความละเอียดของวงจร DAC แบบ R-2R ดังรูปที่ จ.67 นี้มีค่าเท่ากับ 5/256 = 0.0195 โวลต์

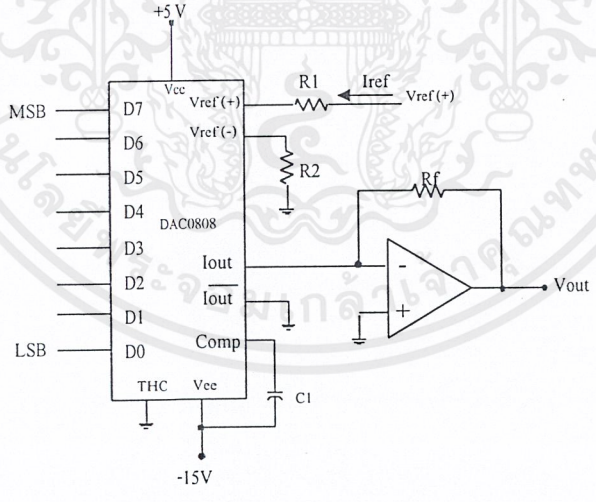
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแปลงสัญญาณดิจิตอลเป็นแอนะลอกโดยใช้ไอซี DAC0808

DAC0808 เป็นไอซีแปลงสัญญาณดิจิตอลเป็นแอนะลอก ซึ่งมีแหล่งกำเหนิดกระแสอ้างอิงอยู่ภายในตัวไอซีดังรูปที่ จ.68 แสดงการจัดขาของ DAC0808 ขา 5 ถึงขา 12 เป็นขารับข้อมูลอินพุตแบบดิจิตอลขนาด 8 บิต ข้อมูลที่ป้อนเข้ามาจะถูกแปลงเป็นกระแส Iout ออกทางขา 4



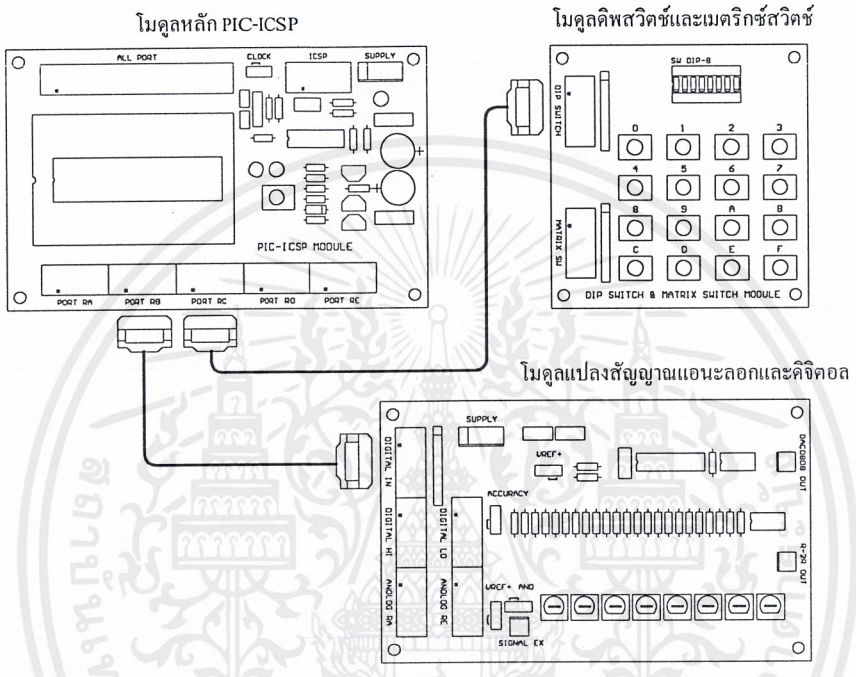
รูปที่ จ.68 การจัดขาของไอซี DAC0808



รูปที่ จ.69 วงจรเพื่อการใช้งานเบื้องต้นของไอซี DAC0808

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ จ.69 แสดงวงจรใช้งานเบื้องต้นของไอซี DAC0808 ที่ต่อเป็นตัวแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะลอก ตัวต้านทาน R1 คือ ตัวต้านทานปรับค่าได้เพื่อกำหนดค่ากระแสอ้างอิงให้มีค่าเท่ากับ 2 มิลลิแอมป์ ตัวต้านทาน R2 มีค่าเท่ากับตัวต้านทาน R1 ต่อเพื่อชดเชยกระแสอินพุตภายในไอซี กระแสเอาต์พุตออกทางขา4 นำไปแปลงเป็นแรงดันด้วยไอซีออปแอมป์จากสูตร



รูปที่ จ.70 การเชื่อมต่อโมดูลหลัก PIC-ICSP เข้ากับโมดูลดิพสวิตช์และโมดูลแปลงสัญญาณ

```
   list p=16f877
   include "p16f877.inc"
   __config _WDT_OFF & _PWRTE_ON & _LVP_OFF & _XT_OSC

            org     0x00
            bsf     STATUS,RP0
            movlw   0xff
            movwf   TRISC
            clrf    TRISB
            bcf     STATUS,RP0

Start       movf    PORTC,W         ; read dip switch
            movwf   PORTB           ; mov digital signal to DAC
            goto    Start

            end
```

รูปที่ จ.71 โปรแกรมการทดลองแปลงสัญญาณดิจิตอลเป็นแอนะลอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับขั้นตอนการทดลอง

1. เขียนโปรแกรมดังรูปที่ จ.71 ทำการแอสเซมเบลอร์แล้วเขียนข้อมูลลงบน PIC16F876
2. ต่อสายเชื่อมต่อตามรูปที่ จ.70
3. ทำการรันโปรแกรม โดยการคลิกปุ่ม RUN
4. ทำการกดดิพสวิตช์ และสังเกตการแสดงผลที่โมดูลเครื่องมือวัด

ตารางที่ จ.11 ผลการกดดิพสวิตช์เมื่อใช้วงจร DAC

ลำดับที่	การกดสวิตช์	แสดงผล
1	00000000	
2	00000011	
3	00000110	
4	00001100	
5	00011000	
6	00110000	
7	01100000	
8	11000000	

5. ทำการทดลองซ้ำอีกครั้ง แต่เปลี่ยนคอนเนคเตอร์ในโมดูลแปลงสัญญาณจากวงจร DAC0808 เป็น R-2R ladder
6. ทำการกดดิพสวิตช์และสังเกตค่าแรงดันในโมดูลเครื่องมือวัดเทียบกับการทดลองข้อ 4.

ตารางที่ จ.12 ผลการกดดิพสวิตช์เมื่อใช้วงจร R-2R ladder

ลำดับที่	การกดสวิตช์	แสดงผล
1	00000000	
2	00000011	
3	00000110	
4	00001100	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ จ.12 (ต่อ) ผลการกดดิพสวิตช์เมื่อใช้วงจร R-2R ladder

5	00011000	
6	00110000	
7	01100000	
8	11000000	

สรุปผลการทดลอง

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

คำถามท้ายการทดลอง

1. จงอธิบายโปรแกรมดังรูปที่ จ.71

2. จงอธิบายหลักการทำงานของการแปลงสัญญาณดิจิตอลเป็นแอนะลอกแบบ R-2R Ladder มาพอเข้าใจ

3. จงอธิบายความแตกต่างของการแปลงสัญญาณดิจิตอลเป็นแอนะลอกแบบที่ใช้ไอซี DAC0808 และแบบ R-2R Ladder มาพอเข้าใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 13
การใช้งาน PIC16F876 เชื่อมต่อกับโซลิดสเตทรีเลย์
(Solid State Relay : SSR)

จุดประสงค์

1. เพื่อให้สามารถใช้งาน PIC16F876 รวมกับอุปกรณ์โซลิดสเตทรีเลย์ได้
2. เพื่อให้สามารถอธิบายข้อดีและข้อเสียของโซลิดสเตทรีเลย์ได้
3. เพื่อให้สามารถเขียนโปรแกรมควบคุมการแสดงผลโดยผ่านอุปกรณ์โซลิดสเตทได้

ทฤษฎีเบื้องต้น

โซลิดสเตทรีเลย์หรือที่เรียกสั้นๆ ว่า SSR คือ อุปกรณ์อิเล็กทรอนิกส์อย่างหนึ่งที่เหมือนกับรีเลย์ไฟฟ้าเครื่องกลโดยปราศจากเครื่องกลที่เคลื่อนไหว โดยพื้นฐานแล้วโซลิดสเตทรีเลย์จะมีขั้วอินพุตและเอาต์พุตอย่างละ 2 ขั้ว ขั้วอินพุตเป็นขั้วสำหรับป้อนสัญญาณควบคุมหรือสัญญาณกระตุ้นเพื่อบังคับให้สวิตช์ทางเอาต์พุตปิดหรือเปิดโดยจะมีการแยกทางไฟฟ้าระหว่างขั้วอินพุตและเอาต์พุตซึ่งจะใช้แสงในการเชื่อมโยงด้วยแสง (Opto coupling)

โซลิดสเตทรีเลย์ส่วนใหญ่จะเป็นชนิดที่สวิตช์ทางด้านเอาต์พุตเป็นแบบขั้วเดียวทางเดียวปกติเปิด (SPST : NO) ทั้งนี้สาเหตุสำคัญเนื่องมาจากราคาต่อหน้าสัมผัสหนึ่งชุดยังคงสูงกว่าของรีเลย์ไฟฟ้ากลมาก เมื่อกล่าวถึงภาวะนำกระแสและภาวะไม่นำกระแสของโซลิดสเตทรีเลย์จะหมายถึง ภาวะที่ด้านอินพุตถูกกระตุ้น และไม่ถูกกระตุ้นจะทำการอ้างถึงรีเลย์แบบปกติเปิดเท่านั้น

คุณสมบัติทางด้านเอาต์พุตของโซลิดสเตทรีเลย์

1. กระแสโหลดสูงสุด (Maximum Load Current) กระแสสูงสุดที่โซลิดสเตทรีเลย์สามารถทนได้ ขึ้นอยู่กับความสามารถในการระบายความร้อน เพราะว่ากระแสที่ทนได้อาจต่ำกว่าค่าที่ระบุก็ได้ ขึ้นอยู่กับแผ่นระบายความร้อนที่เลือกไว้ด้วย

2. กระแสโหลดต่ำสุด (Minimum Load Current) กระแสโหลดต่ำสุดจะต้องไม่ต่ำกว่าที่กำหนดไว้โซลิดสเตทรีเลย์จึงจะทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของโซลิดสเตทรีเลย์แบ่งได้เป็น 2 ภาคใหญ่ๆ คือ

1. ภาคอินพุต จะเป็นภาครับสัญญาณจากการควบคุมมาทำงานโดยเป็นการส่งผ่านสัญญาณไปควบคุมอุปกรณ์ติดต่อซึ่งจะแยกกันในทางไฟฟ้า ซึ่งแบ่งออกเป็น 2 ลักษณะคือ

1.1 ใช้อุปกรณ์ทางออปติคอล (Optical แบบ Photo) เช่น อาจจะใช้ Photo resistor, Photo darlington, Photodiode, Photo SCR หรือ Photo transistor เป็นต้น ตัวอุปกรณ์จุเปลี่ยนสัญญาณไฟฟ้าที่ส่งมาให้อยู่ในรูปของแสงไปควบคุมส่วนเอาต์พุตจะทำให้เป็นการตัดขาดระหว่างอินพุตและเอาต์พุตทางไฟฟ้า

1.2 ใช้อุปกรณ์ทรานฟอเมอร์ (Transformer) โดยจะเป็นการทำงานร่วมกันระหว่างอินพุตและออสซิลเลเตอร์ (Oscillator) โดยมีการออสซิลเลเตอร์ความถี่สูง 50 กิโลเฮิรตซ์ ถึง 10 เมกะเฮิรตซ์ ในการส่งสัญญาณจากอินพุตไปยังเอาต์พุต

2. ภาคเอาต์พุต เป็นส่วนรับสัญญาณจากอินพุตซึ่งอาจจะอยู่ในรูปของแสงมาทำงานโดยต่อวงจรหรือตัดวงจรโดยอุปกรณ์นี้อาจจะเป็นทรานฟอเมอร์, เอสซีอาร์ หรือไตรแอค ก็ได้

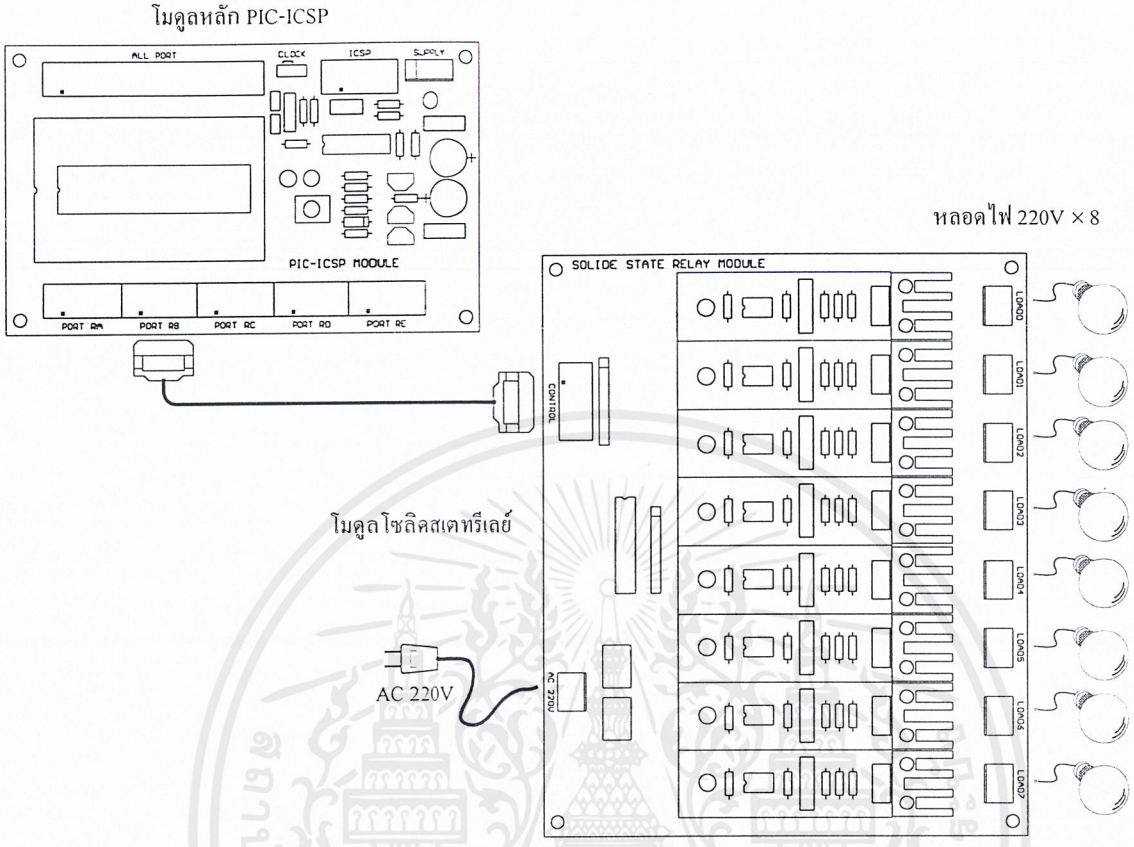
ข้อดีของโซลิดสเตทรีเลย์

1. การทำงานจะต่อวงจรที่แรงดัน (Voltage) ขณะเป็น 0 โวลต์ หรือใกล้จุด 0 โวลต์
2. มีอายุการใช้งานยาวนานมาก
3. ไม่มีเสียงเกิดขึ้นในเวลาทำงาน
4. สามารถต่อเข้ากับระบบไมโครโปรเซสเซอร์ (Microprocessor) ได้โดยง่าย
5. สามารถตัดต่อวงจรได้อย่างรวดเร็ว
6. ไม่มีส่วนเคลื่อนไหวทางเครื่องกลในการทำงาน
7. ไม่เกิดอาการ Bounce ที่หน้าสัมผัส
8. ไม่ก่อให้เกิดสัญญาณรบกวนต่อระบบไฟฟ้า

ข้อเสียของโซลิดสเตทรีเลย์

1. ไม่สามารถใช้งานกับวงจรที่มีแรงดันสูงมากๆ ได้
2. มีราคาแพง
3. ไม่สามารถใช้โซลิดสเตทรีเลย์สำหรับไฟสลับ (AC) หรือไฟตรง (DC) ได้ในตัวเดียวกัน
4. ขณะใช้งานจะเกิดความเร็วขึ้นที่ตัวโซลิดสเตทรีเลย์นั้นๆ จำเป็นต้องใช้อุปกรณ์ระบายความร้อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ จ.72 การเชื่อมต่อโมดูลหลัก PIC-ICSP และโมดูลโซลิดสเตทรีเลย์

```
            List p=16f876
            #Include "p16f876.inc"
            __config _WDT_OFF & _PWRTE_ON & _LVP_OFF & _XT_OSC
Count1      equ     0x20
Count2      equ     0x21

            org     0x00

            bsf     STATUS,RP0
            movlw   0x00            ;พอร์ต B เป็นเอาต์พุต
            movwf   PORTB
            bcf     STATUS,RP0      ;เคลียร์ STATUS

Start       movlw   0xff            ;LED ติดทุกดวง
            movwf   PORTB
            call    Delay
            movlw   0x00            ;LED ดับทุกดวง
            movwf   PORTB
            call    Delay
            goto    Start

;---------------------------------- หน่วงเวลา
Delay       movlw   0Xff
            movwf   Count1
Loop2       movlw   0Xff
            movwf   Count2
Loop1       decfsz  Count2,F
            goto    Loop1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        decfsz   Count1,F
        goto     Loop2
        return
        end
```

รูปที่ จ.73 โปรแกรมใช้ในการทดลองโซลิดสเตทรีเลย์

ลำดับขั้นตอนการการทดลอง

1. เขียนโปรแกรมดังรูปที่ จ.73 ทำการแอสเซมเบลอร์แล้วเขียนข้อมูลลงบน PIC16F876
2. ต่อสายเชื่อมต่อตามรูปที่ จ.72
3. นำหลอดไฟ 8 ดวงมาต่อเข้ากับ LOAD1 ถึง LOAD8 ของโมดูลโซลิดสเตทรีเลย์
4. ทำการรันโปรแกรม โดยคลิกปุ่ม RUN
5. สังเกตหลอดไฟทั้ง 8 ดวง ผลเป็นดังนี้

.....

.....

6. จากโปรแกรมรูปที่ จ.73 ถ้าเปลี่ยนข้อมูลจาก movlw 0xff เป็น movlw 0xaa หลอดไฟทั้ง 8 ดวงจะเกิดการเปลี่ยนแปลงดังนี้

.....

.....

สรุปผลการทดลอง

.....

.....

.....

.....

คำถามท้ายการทดลอง

1. จงบอกข้อดีและข้อเสียของโซลิดสเตทรีเลย์มาพอเข้าใจ
2. จงเขียนโปรแกรมควบคุมให้หลอดไฟวิ่งจากซ้ายไปขวาเมื่อวิ่งครบแล้วให้วิ่งจากขวาไปซ้าย ทำเช่นนี้ไปเรื่อยๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบงานที่ 14
การสื่อสารข้อมูลกับคอมพิวเตอร์ผ่านพอร์ตอนุกรม RS-232

จุดประสงค์

1. เพื่อให้สามารถอธิบายหลักการสื่อสารข้อมูลกับคอมพิวเตอร์แบบอนุกรม RS-232ได้
2. เพื่อให้สามารถใช้งานพอร์ตการสื่อสารแบบอนุกรมของไมโครคอนโทรลเลอร์ PIC16F876 ได้
3. เพื่อให้สามารถเชื่อมต่อไมโครคอนโทรลเลอร์ PIC16F876 เข้ากับกับคอมพิวเตอร์ผ่านพอร์ตอนุกรมมาตรฐาน RS-232C ได้

อุปกรณ์ที่ใช้ในการทดลอง

1. โมดูลหลัก PIC-ICSP
2. โมดูลเบสิกอินพุตเอาต์พุต
3. โมดูลสื่อสารข้อมูลอนุกรม
4. คอมพิวเตอร์พีซีที่ติดตั้งโปรแกรม MPLAB และโปรแกรมสื่อสารข้อมูลอนุกรม MSCOM.EXE
5. สายเชื่อมต่อระหว่างคอมพิวเตอร์พีซีและโมดูลสื่อสารข้อมูลอนุกรม
6. สายเชื่อมต่อพอร์ตระหว่างโมดูล

ทฤษฎีเบื้องต้น

RS-232 นั้นเป็นระบบการส่งข้อมูลในรูปแบบอนุกรมคือ ข้อมูลจะส่งไปได้ทีละบิตโดยจะมีอัตราการส่งข้อมูลเป็นบิตต่อวินาทีหรืออัตราการเปลี่ยนแปลงของสัญญาณใน 1 วินาทีเรียกว่า Baud rate โดยจะมีการส่งบิตเริ่มต้น (Start Bit) มีระดับสัญญาณเป็น 0 และบิตข้อมูล (Data bit) ซึ่งจะมีข้อมูล 7 บิตหรือ 8 บิต และอาจตามด้วยบิตพาริตี้ (Parity bit) อาจจะเป็นแบบคู่ (Even) หรือแบบคี่ (Odd) เพื่อตรวจสอบความถูกต้องของข้อมูล บิตพาริตี้นั้นจะมีหรือไม่มีก็ได้และสุดท้ายจะต้องตามด้วยบิตหยุด (Stop bit) ซึ่งจะมีความกว้างของสัญญาณเป็น 1, 1.5, 2 บิตก็ได้ ซึ่งจะเห็นว่าในหารส่งข้อมูลแบบนี้จึงจำเป็นต้องมีข้อตกลงกันระหว่างการรับและการส่งคือ

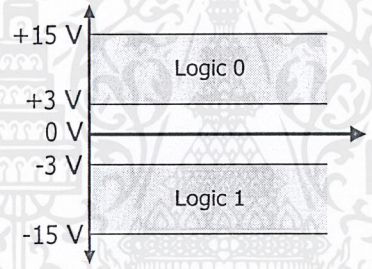
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ความเร็วในการส่ง Boud rate
2. จำนวนข้อมูล
3. มีบิตพาริตี้หรือ ถ้ามีจะเป็นแบบคู่หรือแบบคี่
4. จำนวนบิตหยุด 1, 1.5 หรือ 2 บิต

ในการส่งแบบ RS-232C นี้เราสามารถเลือก Baud rate ได้หลาย Boud rate เช่น 110, 200, 300, 1200, 2400, 4800, 9600 เช่นถ้าเราส่งข้อมูล 8 บิตโดย Boud rate 2400 แล้ว 1 บิตต้องใช้เวลา 1/2400 = 416 ไมโครวินาที

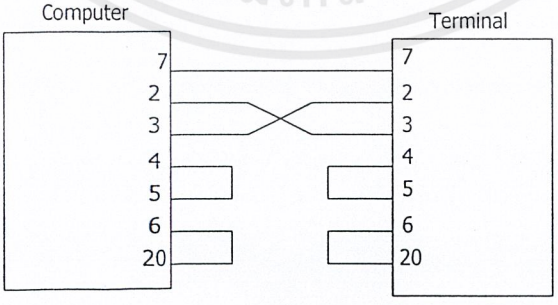
ลักษณะของสัญญาณ RS-232C

เพื่อให้การส่งข้อมูลสามารถไปได้ไกลขึ้นจึงกำหนดให้การส่งสัญญาณมีระดับความแตกต่างเป็นบวก 15 โวลต์ หรืออาจจะเป็นบวก 12 โวลต์ และลบ 12 โวลต์ก็ได้มีระดับสัญญาณลอจิกดังรูปที่ จ.74



รูปที่ จ.74 ระดับสัญญาณลอจิก RS-232C

การกำหนดจุดข้อต่อของ RS-232C



รูปที่ จ.75 การกำหนดจุดต่อของ RS-232C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณของขาต่างๆ ที่ใช้งาน

ขาที่ 2 Transmit data

ขาที่ 3 Receive data เป็นขาที่ได้รับสัญญาณข้อมูลจากเครื่องอื่น

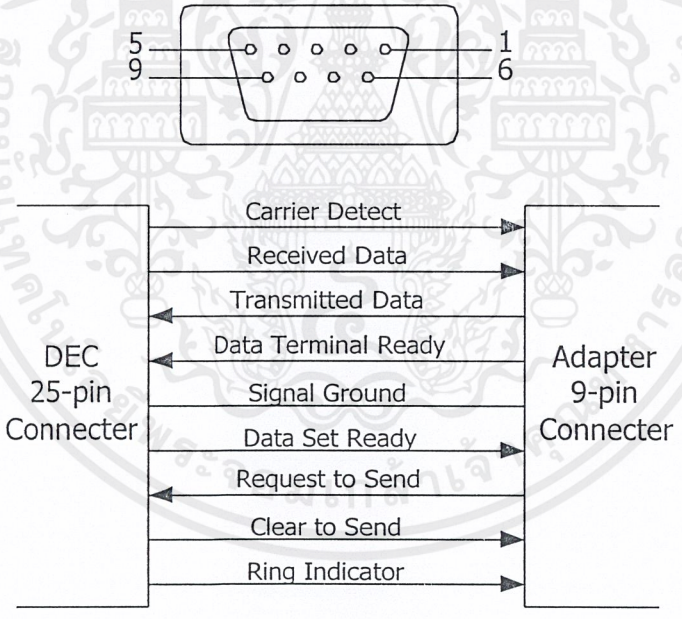
ขาที่ 4 Request to sent เป็นขาที่บอกเครื่องอื่นๆ ว่าตัวเองพร้อมที่จะส่งข้อมูล

ขาที่ 5 Clear to send เป็นขาที่บอกต่อเครื่องอื่นๆ ว่าตัวเองพร้อมที่จะรับข้อมูล

ขาที่ 6 Data set ready เป็นขาที่บอกไมโครคอมพิวเตอร์ว่าโมเด็มต่อเข้ากับสายโทรศัพท์เรียบร้อยแล้วและพร้อมที่จะส่งได้แล้ว

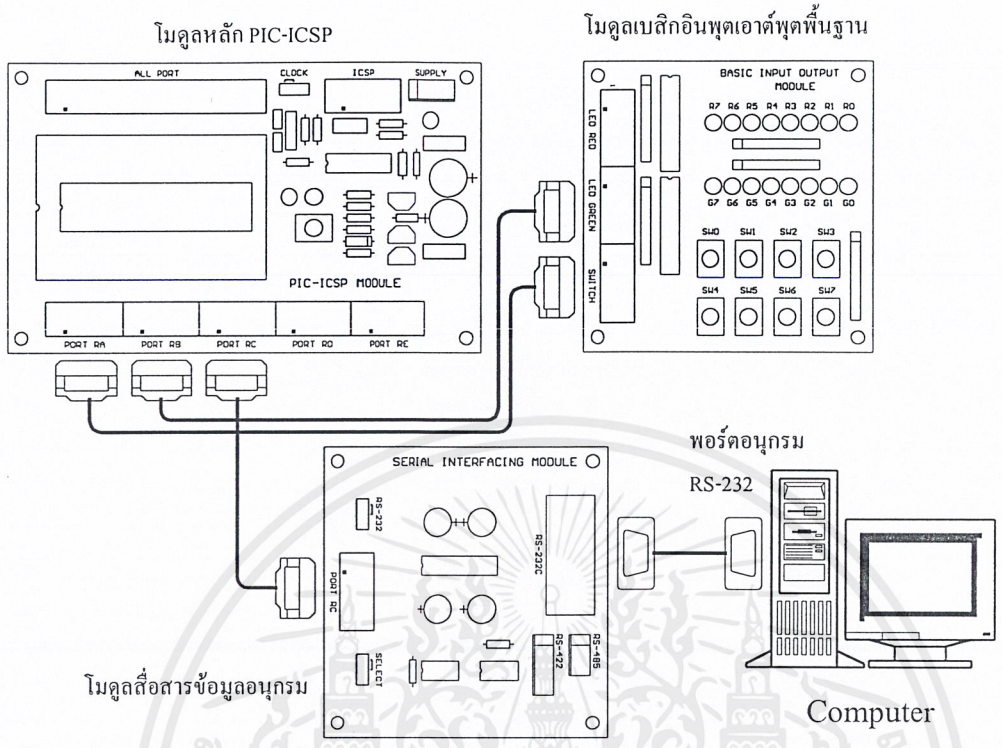
ขาที่ 7 Signal ground เป็นขา GND ของสัญญาณ

ขาที่ 20 Data terminal ready เป็นขาบอกว่าโมเด็มไมโครคอมพิวเตอร์พร้อมที่จะติดต่อด้วย โดยทั่วไปเราสามารถต่อใช้เครื่องคอมพิวเตอร์ติดต่อกันทาง RS-232C ได้ง่ายๆ โดยใช้สัญญาณ Transmit data เพียง 2 เส้น โดยต่อได้ดังรูปที่ จ.76



รูปที่ จ.76 สัญญาณของขาต่างๆ ที่ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ จ.77 การเชื่อมต่อโมดูลสื่อสารข้อมูลกับคอมพิวเตอร์

```
    list p=16f876
    include "p16f876.inc"
  __config _PWRTE_ON & _WDT_OFF & _LVP_OFF & _XT_OSC
offset      equ     0x20
temp        equ     0x21
Dely1       equ     0x22
Dely2       equ     0x23
            ORG     0x00
init        bsf     STATUS,RP0      ; select bank 1
            movlw   0x06
            movwf   ADCON1
            movlw   0xff
            movwf   TRISA
            clrf    TRISB
            movlw   0x19            ; BAUDrate 2400 , (25 decimal)
            movwf   SPBRG
            clrf    TXSTA           ; 8bitsdata,no,1stop
            bcf     STATUS,RP0      ; select bank0
            bsf     RCSTA,SPEN      ; Asynchronous serial port enable
            bsf     RCSTA,CREN      ; continuous receive
            bsf     STATUS,RP0      ; select bank1
            bsf     TXSTA,TXEN      ; Transmit enable
            bcf     STATUS,RP0      ; select bank0
main        call    SendTitle
main2       call    Receive
            call    ScanSW
            goto    main2
SendTitle   clrf    offset          ; load offset of character table
Tit1        movf    offset,W
            call    TAB
            addlw   0x00            ; Character=00?
            btfsc   STATUS,Z        ; Character=00?
            return                  ; Yes,Z=1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                movwf   PORTB
                call    Sender
                call    Delay
                incf    offset,f
                goto    Tit1            ; Send again
;_____________________________________ Sender
Sender          movwf   TXREG
                movlw   TXSTA
                movwf   FSR
wait1           btfss   INDF,1
                goto    wait1
                return                  ; TXREG empty or TRMT=1
;_____________________________________ starttoreceive
     Receive    btfss   PIR1,RCIF       ; Check RCIF bit on PIR1 register (Peripheral  interrupts)
                return                  ; RCREG empty or RCIF=0
                movf    RCREG,W         ; RCREG full or RCIF=1
                movwf   PORTB
                return
;_____________________________________ Scan Switch
ScanSW          movlw   0x00
                btfsc   PORTA,0
                goto    Sc1
                movlw   0x30
Sc1             btfsc   PORTA,1
                goto    Sc2
                movlw   0x31
Sc2             btfsc   PORTA,2
                goto    Sc3
                movlw   0x32
Sc3             btfsc   PORTA,3
                goto    Sc4
                movlw   0x33
Sc4             btfsc   PORTA,4
                goto    Sc5
                movlw   0x34
Sc5             btfsc   PORTA,5
                goto    SEnd
                movlw   0x35
SEnd            addlw   0x00
                btfsc   STATUS,Z
                return
                call    Sender
Scw             movf    PORTA,W
                andlw   0x3f
                sublw   0x3f
                btfss   STATUS,Z
                goto    Scw
                return
;_____________________________________ Tebelofmessage
TAB             addwf   PCL,F           ;Move offset to PC lower
                dt      "ชุ                                ", 0a, 0d
                dt      "PIC16F8XX Experimentation Practice Set", 0a, 0d
                dt      "การสื่อสารข้อมูลอนุกรม RS-232", 0a, 0d
                dt      00
Delay           movlw   0x20
                movwf   Dely1
Delay2          movlw   0xff
                movwf   Dely2
Delay1          decfsz  Dely2,F
                goto    Delay1
                decfsz  Dely1,F
                goto    Delay2
                return
                end
```

รูปที่ จ.78 โปรแกรมรับส่งข้อมูลระหว่างไมโครคอนโทรลเลอร์ PIC16F876 กับคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับขั้นตอนการทดลอง

1. เขียนโปรแกรมรูปที่ จ.78 ทำการแอสเซมเบลอร์แล้วเขียนข้อมูลลงบน PIC16F876
2. ต่อสายเชื่อมต่อตามรูปที่ จ.77
3. เปิดโปรแกรมสื่อสารข้อมูลอนุกรม MSCOM.EXE
4. คลิกเมนู CommPort/Properties
5. เลือกพอร์ตอนุกรมที่ต่อสายสัญญาณข้อมูลอยู่
6. ทำการเลือก Maximum Speed = 2400, Data Bits = 8, Parity = None และ Stop bits = 1 จากนั้นคลิก OK
7. ทำการรันโปรแกรม โดยการคลิกปุ่ม RUN
8. สังเกตที่โปรแกรมสื่อสารข้อมูลอนุกรม MSCOM.EXE

.....

.....

.....

9. ทดลองกดปุ่มตัวอักษรบนคีย์บอร์ด และสังเกตการแสดงผลได้จาก LED บนโมดูลเบสิกอินพุตเอาต์พุต

ตารางที่ จ.13 ผลการกดคีย์ตัวอักษรบนคีย์บอร์ด

คีย์ที่กด	LED แสดงผล (ฐานสิบหก)
A	
K	
d	
h	
P	
e	
Z	

10. ทดลองกดสวิตช์ใดๆ บนโมดูลเบสิกอินพุตเอาต์พุต และสังเกตการแสดงผลบนโปรแกรมสื่อสารข้อมูลอนุกรม MSCOM.EXE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ จ.14 ผลการกดสวิตช์ที่โมดูลเบสิกอินพุตเอาต์พุต

สวิตช์ที่กด	LED แสดงผล (ฐานสิบหก)
SW1	
SW2	
SW3	
SW4	
SW5	

สรุปผลการทดลอง

.....

.....

.....

.....

.....

.....

.....

.....

คำถามท้ายการทดลอง

1. จงอธิบายอธิบายหลักการสื่อสารข้อมูลกับคอมพิวเตอร์ผ่านพอร์ตอนุกรม RS-232 ได้
2. ถ้าต้องการเปลี่ยน Baud rate ในการสื่อสารข้อมูลจะต้องเปลี่ยนค่าในรีจิสเตอร์ใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ฉ
เฉลยใบงานการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เฉลยใบงานที่ 1

การใช้งานพอร์ตของ PIC16F876

ลำดับขั้นตอนการทดลอง

1. เปิดโปรแกรม MPLAB ขึ้นมาใช้งานเพื่อเขียนโปรแกรม
2. ไปที่เมนู OPTION / Development Mode กำหนดโหมดเป็น MPLAB-SIM Simulator
3. เลือก Process เป็นรุ่น PIC 16F876 จากนั้นกดปุ่ม Apply แล้วกด OK
4. ไปที่เมนู Project เลือก New Project ตั้งชื่อ Project Name แล้วกด OK
5. ในเมนู Edit Project ให้ "Remove" Project Files ออกให้หมดถ้ามีไฟล์อยู่จากนั้นกด OK
6. ไปที่เมนู File เลือก New Source เพื่อเขียนโปรแกรมดังรูปที่ จ.3 ลงไป
7. จากนั้น save file เป็นนามสกุล . asm จากนั้นทำการแอสเซมเบลอร์
8. ไปที่เมนู Project เลือก Edit Project แล้วกำหนดชื่อไฟล์เป็น ชื่อที่เราบันทึกไว้ในข้อ 6. เพื่อ Add ไปยัง Project Files
9. ไปที่เมนู Project เลือก Make Project เพื่อแอสเซมเบลอร์โปรแกรมที่เขียนขึ้น (หลังจากแอสเซมเบลอร์จะได้ไฟล์เป็นนามสกุล .hex)
10. ตรวจสอบผลการแอสเซมเบลอร์ว่ามีข้อผิดพลาดหรือไม่ ถ้ามีจัดการแก้ไขข้อผิดพลาดที่เกิดขึ้นให้เรียบร้อย แล้วทำตามขั้นตอนในข้อ 8 อีกครั้งจนไม่มีข้อผิดพลาดอีก
11. เขียนข้อมูลลงบน PIC16F876 โดยใช้โปรแกรม PIC-ICSP Device for Windows
12. ต่อสายเชื่อมต่อระหว่างโมดูลตามรูปที่ จ.2
13. สังเกตผลการทำงานของ LED ทั้ง 8 ดวง

LED ทั้ง 8 ดวงจะกระพริบคือ จะติดดับสลับกันไปเรื่อยๆ

14. ในบรรทัด Start movlw 0xFF ถ้าเปลี่ยนข้อมูลจาก 0xFF เป็น 0x01 จะได้ผลเป็นดังนี้

LED ดวงแรกซ้ายมือสุดจะกระพริบ ส่วนดวงอื่นๆที่เหลือจะดับ

15. ถ้าต้องการให้ LED ดวงที่ 7 (บิตที่ 7) ติดและดับ จะต้องเปลี่ยนข้อมูลเป็นเท่าใด

movlw 0x80

16. จากโปรแกรมดังรูปที่ จ.2 ถ้าต้องการให้พอร์ต A เป็นเอาต์พุตโดยให้ LED ติด-ดับทุกดวง จะต้องเขียนโปรแกรมอย่างไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            org     0x00
            bsf     STATUS,RP0
            movlw   0x00
            movwf   PORTA
            bcf     STATUS,RP0

Start       movlw   0xff
            movwf   PORTA
            call    Delay
            movlw   0x00
            movwf   PORTA
            call    Delay
            goto    Start
;-------------------------  Delay  Time -------------------------
Delay       movlw   0Xff
            movwf   Count1
Loop2       movlw   0Xff
            movwf   Count2
Loop1       decfsz  Count2,F
            goto    Loop1
            decfsz  Count1,F
            goto    Loop2
            return
            end
```

รูปที่ ฉ.1 เฉลยโปรแกรมกำหนดพอร์ต A เป็นเอาต์พุต

17. ทำการเขียนโปรแกรมดังรูปที่ จ.4 จากนั้น save file เป็นนามสกุล . asm ในหน้าต่าง edit จากนั้นทำการแอสเซมเบลอร์

18. เขียนข้อมูลลงบน PIC16F876 โดยใช้โปรแกรม PIC ICSP Device for Windows

19. ต่อสายเชื่อมต่อระหว่างโมดูลตามรูปที่ จ.2

20. เมื่อมีการกดสวิตซ์แต่ละตัว

ตารางที่ ฉ.1 ผลการติดของ LED จากการกดสวิตช์

สวิตช์	LED1	LED2	LED3	LED4	LED5	LED6
SW1	ติด	ดับ	ดับ	ดับ	ดับ	ดับ
SW2	ดับ	ติด	ดับ	ดับ	ดับ	ดับ
SW3	ดับ	ดับ	ติด	ดับ	ดับ	ดับ
SW4	ดับ	ดับ	ดับ	ติด	ดับ	ดับ
SW5	ดับ	ดับ	ดับ	ดับ	ติด	ดับ
SW6	ดับ	ดับ	ดับ	ดับ	ดับ	ติด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

21. เมื่อไม่มีการกดสวิตช์ใดๆ สังเกตรูปแบบการแสดงผลของ LED แต่ละตัว

เมื่อไม่มีการกดสวิตช์ใดๆ LED จะดับทุกดวง

22. จากโปรแกรมดังรูปที่ จ.4 ถ้าต้องการให้กดสวิตช์ SW1 แล้ว LED ดวงที่ 1 และ 2 ติด และกดสวิตช์ SW2 แล้ว LED ติดทุกดวงจะต้องแก้ไขโปรแกรมเป็นดังนี้

```
SW1         btfsc  PORTA,1
            goto   SW2
            movlw  0x03
            movwf  PORTB
            call        Delay
SW2         btfsc  PORTA,2
            goto   SW3
            movlw  0xff
            movwf  PORTB
            call   Delay
```

คำถามท้ายการทดลอง

1. จงอธิบายหลักการทำงานของโปรแกรมในรูปที่ จ.3 และโปรแกรมในรูปที่ จ.4

หลักการทำงานของโปรแกรมในรูปที่ จ.3 คือ กำหนดให้พอร์ต B เป็นเอาต์พุตจากนั้นก็นำข้อมูล FF ให้กับพอร์ต B ซึ่งเป็นการแสดงให้ LED สว่างทั้งหมด 8 ดวงจากนั้นก็หน่วงเวลาให้ไฟสว่างโดยที่ตาเราสามารถมองเห็นได้ หลังจากนั้นส่งข้อมูล 00 ให้กับพอร์ต B ซึ่งเป็นการทำให้หลอด LED ดับทั้ง 8 ดวงและหน่วงเวลาสักครู่ ดังนั้นโปรแกรมนี้จะเป็นการแสดงไฟกระพริบ ติด-ดับสลับกันไปเรื่อยๆ

หลักการทำงานของโปรแกรมในรูปที่ จ.4 คือ กำหนดให้พอร์ต A ต่อยู่กับสวิตช์เพื่อทำการเช็คสวิตช์ ส่วนพอร์ต B เป็นเอาต์พุตแสดงผลจากการกดสวิตช์ เริ่มจากตรวจสอบว่าพอร์ต A บิต 0 มีการกดสวิตช์ SW1 หรือไม่ ถ้ามีการกด (พอร์ต A บิต 0 เป็น 0) LED ที่ต่ออยู่กับพอร์ต B บิต 0 ก็จะสว่างแต่ถ้าไม่มีการกดสวิตช์ SW1 ก็จะไปตรวจสอบสวิตช์ SW2 , SW3, ไปเรื่อยๆ จนถึงสวิตช์ SW6 ไปเรื่อยๆ ถ้ากดสวิตช์ใด LED ดวงนั้นก็จะติด

2. จงบอกถึงความสัมพันธ์ระหว่างพอร์ต A พอร์ต B กับรีจีสเตอร์ TRISA และ TRISB

การที่เราจะกำหนดให้พอร์ต A หรือพอร์ต B เป็นอินพุตหรือเอาต์พุตนั้นเราต้องกำหนดทิศทาง TRISA ให้กับพอร์ต A หรือกำหนดทิศทาง TRIAB ให้กับพอร์ต B โดยข้อมูลแต่ละบิตในรีจีสเตอร์ TRISA บิต 0 จะใช้กำหนดทิศทางของพอร์ต A บิต 0 ไล่เรียงกันไปจนถึงบิต 5 ของ ในทำนองเดียวกันกับรีจีสเตอร์ TRISB บิต 0 ก็กำหนดทิศทางของพอร์ต B บิต 0 ไล่เรียงกันไปจนถึงบิต 7

3. จงเขียนโปรแกรมแสดงรูปแบบไฟวิ่ง 4 แบบ

```
List p=16f876
Include "p16f876.inc"
  config  WDT OFF &  PWRTE ON &  LVP OFF &  XT OSC
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Count1      equ     0x20
Count2      equ     0x21
DATA1       equ     0x22
            org     0x00
            bsf     STATUS,RP0
            movlw   0x06
            movwf   ADCON1
            movlw   0xff
            movwf   PORTA
            clrf    PORTB
            bcf     STATUS,RP0
            clrf    PORTB
SW1         btfsc   PORTA,0
            Goto    SW2
Loop1       movlw   0x01
            Movwf   DATA1
Loop2       mowf    DATA1,W
            Movwf   PORTB
            Call    Delay
            Btfsc   PORTB,7
            Goto    Loop1
            Bcf     STATUS,C
            Rlf     DATA1,F
            Btfss   PORTA,1
            Goto    SW2
            Goto    Loop2
SW2         btfsc   PORTA,1
            Goto    SW3
Loop3       movlw   0x80
            Movwf   DATA1
Loop4       mowf    DATA1,W
            Movwf   PORTB
            Call    Delay
            Bcf     STATUS,C
            Rrf     DATA1,F
            Btfss   PORTA,1
            goto    SW3
            btfss   STATUS,C
            goto    Loop4
SW3         btfsc   PORTA,2
            goto    SW4
            movlw   0xC3
            movwf   PORTB
            call    Delay
            movlw   0x3C
            movwf   PORTB
            call    Delay
            goto    SW3
SW4         btfsc   PORTA,3
            goto    SW1
            movlw   0xf0
            movwf   PORTB
            call    Delay
            movlw   0x0f
            movwf   PORTB
            call    Delay
            goto    SW4

Delay       movlw   0xFF
            movwf   Count1
Loop1       movlw   0xFF
            movwf   Count2
Loop2       decfsz  Count2,F
            goto    Loop2
            decfsz  Count1,F
            goto    Loop1
            return
            end
```

รูปที่ ฉ.2 โปรแกรมแสดงรูปแบบไฟวิ่ง 4 แบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เฉลยใบงานที่ 2

การทำงานของ PIC16F876 กับวงจรกำเนิดสัญญาณนาฬิกา แบบ RC และแบบคริสตอล (Crystal)

ลำดับขั้นตอนการทดลอง

1. เขียนโปรแกรมดังรูปที่ จ.8 และทำการแอสเซมเบอร์
2. เขียนข้อมูลลงบนตัว PIC16F876
3. ต่อสายเชื่อมต่อโมดูลตามรูปที่ จ.7 และทำการเซตจั๊มเปอร์ไปที่โหมด RC ซึ่งเป็นการกำหนดให้วงจรทำงานในโหมด RC โดยให้พอร์ต A ต่ออยู่กับสวิตช์ และพอร์ต B ต่ออยู่กับ LED RED หรือ LED GREEN แต่ในการทดลองนี้จะใช้ LED GREEN
4. สังเกตผลการทำงานของ LED ทั้ง 8 ดวงที่ต่ออยู่กับพอร์ต B

LED จะวิ่งจากขวาไปซ้ายไปเรื่อยๆ

5. เปลี่ยนส่วนของวงจรกำเนิดสัญญาณนาฬิกามาเป็นแบบ XT โดยเปลี่ยนการเซตจั๊มเปอร์
6. เปลี่ยนโปรแกรมรูปที่ จ.8 บรรทัดที่ 3 กำหนดสัญญาณนาฬิกาจาก _RC_OSC เป็น _XT_OSC แล้วทำการแอสเซมเบอร์ใหม่
7. เขียนข้อมูลลงบน PIC16F876 อีกครั้ง
8. สังเกตผลการทำงานของ LED ทั้ง 8 ดวงเปรียบเทียบกับผลที่ได้จากการทดลองข้อ 4 ได้

LED จะวิ่งจากขวาไปซ้ายไปเรื่อยๆ เหมือนกับข้อที่ 4 แต่จะวิ่งเร็วกว่า

คำถามท้ายการทดลอง

1. จงบอกข้อดีในการสร้างวงจรกำเนิดสัญญาณนาฬิกาทั้งแบบ RC และ XT มาพอเข้าใจ

ข้อดีของ RC

- กำหนดค่าความถี่ได้ตั้งแต่ 0-4 กิโลเฮิรตซ์
- ประหยัดอุปกรณ์ในการทดลอง
- วงจรมีขนาดเล็ก

ข้อดีของ XT

- สัญญาณมีความเที่ยงตรงสูง
- การคำนวณการหน่วงเวลาแม่นยำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การคำนวณความถี่แม่นยำ

2. จงอธิบายหลักการทำงานของโปรแกรมรูปที่ จ.8 พอเข้าใจ

หลักการทำงานของโปรแกรมในรูปที่ จ.8 คือ กำหนดให้พอร์ต B เป็นเอาต์พุตแสดงไฟวิ่งจากขวาไปซ้ายไปเรื่อยๆ ซึ่งการสร้างวงจรกำเนิดสัญญาณนาฬิกาทั้งแบบ RC และ XT จะใช้โปรแกรมเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เฉลยใบงานที่ 3

การอินเตอร์รัพต์ PIC16F876

ลำดับขั้นตอนการทดลอง

1. ต่อสายเชื่อมต่อระหว่างโมดูลตามรูปที่ จ.10

2. เขียนโปรแกรมที่ จ.11 ทำการแอสเซมเบลอร์

3. จากนั้นเขียนข้อมูลลงบน PIC16F876 ทำการรันโปรแกรมบน PIC16F876 โดยการกดสวิตช์รีเซต ผลที่ได้เป็นดังนี้

จะเป็นไฟ LED กระพริบ เมื่อเกิดการเปลี่ยนแปลงที่พอร์ต B จะเกิดไฟติดค้างที่พอร์ต A ตามพอร์ตที่เปลี่ยนแปลงสักครู่แล้วที่พอร์ต B กลับมากระพริบเหมือนเดิม

4. เลื่อนดิพสวิตช์ D0 ขึ้นด้านบนเพื่อป้อนสัญญาณ "0" เข้าที่ขา RB7 ผลที่เกิดขึ้นเป็นดังนี้

หลอด LED ที่พอร์ต RA3 จะติดค้างอยู่สักครู่

5. แก้ไขโปรแกรมดังรูปที่ จ.11 จาก bcf INTCON, RBIF เป็น nop ผลที่เกิดขึ้นเป็นดังนี้

เมื่อกดดิพสวิตช์แล้วหลอดไฟจะติดอยู่ครู่หนึ่ง

6. จากข้อ 4 เปลี่ยนจากการป้อนสัญญาณ "0" เข้าที่ขา RB7 เป็น RB4 แทน สังเกตผลที่ได้เมื่อเทียบกับข้อ 4

หลอด LED ที่พอร์ต RA0 จะติดค้างอยู่สักครู่แทน

7. จากข้อ 6 เปลี่ยนขาจาก RB4 เป็นขา RB5 สังเกตผลที่ได้เมื่อเทียบกับข้อ 6

หลอด LED ที่พอร์ต RA1 จะติดค้างอยู่สักครู่แทน

8. จากข้อ 6 เปลี่ยนขาจาก RB4 เป็นขา RB6 สังเกตผลที่ได้เมื่อเทียบกับข้อ 4

หลอด LED ที่พอร์ต RA3 จะติดค้างอยู่สักครู่แทน

9. ต่อสายเชื่อมต่อระหว่างโมดูลตามรูปที่ จ.10

10. เขียนโปรแกรมดังรูปที่ จ.12 ทำการแอสเซมเบลอร์

11. จากนั้นเขียนข้อมูลลงบน PIC16F876 ทำการรันโปรแกรมบน PIC16F876 โดยการคลิกปุ่ม RUN ผลที่ได้เป็นดังนี้ หลอด LED กระพริบ

12. เปลี่ยนรูปแบบการอินเตอร์รัพต์จากขอบขาขึ้นเป็นขอบขาลง โดยการกำหนดบิต 6 ของรีจีสเตอร์ OPTION เป็น "1" แล้วทำการทดลองอีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำถามท้ายการทดลอง

1. บิตใดในรีจีสเตอร์ INTCON ที่อีนาเบิลการเกิดอินเตอร์รัพต์เนื่องจากบิต 4 ถึงบิต 7 ของพอร์ต B เกิดการเปลี่ยนแปลงระดับลอจิก

บิต 3 : RBIE (Port B change interrupt enable bit)

2. การเลือกขอบขาของสัญญาณอินเตอร์รัพต์จากภายนอกกระทำได้อย่างไร

เลือกได้ที่บิต 6 : INTEDG ของรีจีสเตอร์ OPTION

3. บิต GIE มีความสำคัญอย่างไรต่อการอินเตอร์รัพต์

เป็นบิตที่ใช้ควบคุมการ Enable อินเตอร์รัพต์ทั้งหมดที่มีในไมโครคอนโทรลเลอร์

4. จงเขียนโปรแกรมแสดงการอินเตอร์รัพต์เนื่องจากการเปลี่ยนแปลงลอจิกที่บิต 4 ถึง 7 ของพอร์ต B โดยเลื่อนการติดของ LED ทีละหลอดเมื่อเกิดการอินเตอร์รัพต์

```
            List p=16f876
            Include "p16f876.inc"
            __config _WDT_OFF & _PWRTE_ON & _LVP_OFF & _XT_OSC

            org     0x00
            goto    main
            org     0x04
            goto    isr

main        bsf     STATUS,RP0
            movlw   0x06
            movwf   ADCON1
            movlw   0xff
            movwf   PORTB
            clrf    PORTC
            bcf     STATUS,RP0

            movlw   b'10001000'
            movwf   INTCON
            bcf     OPTION_REG,INTEDG
            clrf    PORTC

            bsf     PORTC,0
wait        goto    $

isr         bcf     STATUS,C
            rlf     PORTC,F
            btfsc   STATUS,C
            bsf     PORTC,0
            retfie

            end
```

รูปที่ ฉ.3 เฉลยโปรแกรมแสดงการอินเตอร์รัพต์เนื่องจากระดับลอจิกเปลี่ยนแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เฉลยใบงานที่ 4

การใช้งานไทเมอร์เคาน์เตอร์ภายใน PIC16F87 (Timer/Counter)

ลำดับขั้นตอนการทดลอง

1. เขียนโปรแกรมดังรูปที่ จ.16 ทำการแอสเซมเบลอร์

2. ต่อสายเชื่อมต่อระหว่างโมดูลตามรูปที่ จ.15

3. เขียนข้อมูลลงบน PIC 16F876 ทำการรันโปรแกรมบน PIC16F876 โดยการคลิกปุ่ม RUN ผลที่ได้เป็นดังนี้

LED จะติดเป็นเวลา 500 มิลลิวินาที และจะดับเป็นเวลา 500 มิลลิวินาที สลับกันไปเรื่อยๆ

4. จากโปรแกรมดังรูปที่ จ.16 ถ้าเราเปลี่ยนจาก movlw b'11010100' เป็น movlw b'11010110' ผลที่ได้เป็นดังนี้

LED จะติดเป็นเวลา 2 วินาที และจะดับเป็นเวลา 2 วินาที สลับกันไปเรื่อยๆ

5. เขียนโปรแกรมดังรูปที่ จ.17 ทำการแอสเซมเบลอร์

6. ต่อสายเชื่อมต่อระหว่างโมดูลตามรูปที่ จ.15

7. เขียนข้อมูลลงบน PIC16F876 ทำการรันโปรแกรมบน PIC16F876 โดยการคลิกปุ่ม RUN ผลที่ได้เป็นดังนี้

LED จะติดเป็นเวลา 1วินาที และจะดับเป็นเวลา 1 วินาที สลับกันไปเรื่อยๆ

8. จากโปรแกรมดังรูปที่ จ.17 ถ้าเราเปลี่ยนจาก movlw b'11010101' เป็น movlw b'11010010' ผลที่ได้เป็นดังนี้

LED จะติดเป็นเวลา 25 มิลลิวินาที และจะดับเป็นเวลา 25 มิลลิวินาที สลับกันไปเรื่อยๆ

9. ถ้าต้องการสร้างสัญญาณที่ความถี่ 1 เมกะเฮิรตซ์ จะต้องกำหนด

ค่าของปรีสเกลเลอร์ = หาร 2

TMR0 = 250

10. เขียนโปรแกรมดังรูปที่ จ.18 ทำการแอสเซมเบลอร์

11. ต่อสายเชื่อมต่อระหว่างโมดูลตามรูปที่ จ.15

12. เขียนข้อมูลลงบน PIC16F876 ทำการรันโปรแกรมบน PIC16F876 โดยการคลิกปุ่ม RUN ผลที่ได้เป็นดังนี้

จะรอการกดสวิตช์ 4 ครั้งจึงจะมีการเลื่อนบิตไปทางซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

13. สังเกตการทำงานของ LED

เมื่อกดสวิตช์ที่พอร์ต RA4/TOCKI = 4 ครั้ง LED จึงจะเลื่อนบิตไป1 บิต

14. ถ้าต้องการเปลี่ยนค่าของปรีสเกลเลอร์เป็น 8 สังเกตการทำงานของ LED เป็นดังนี้

จะรอการกดสวิตช์ถึง 8 ครั้งจึงจะมีการเลื่อนบิตไปทางซ้าย

คำถามท้ายการทดลอง

1. ในไมโครคอนโทรลเลอร์ PIC16F876 มีไทเมอร์เคาน์เตอร์กี่ตัว ขนาดกี่บิต อะไรบ้าง

ในไมโครคอนโทรลเลอร์ PIC16F876 มีไทเมอร์เคาน์เตอร์ 3 ตัวคือ ไทเมอร์ 0 มีขนาด 8 บิต, ไทเมอร์ 1 มีขนาด 16 บิต และไทเมอร์ 2 มีขนาด 8 บิต

2. จงอธิบายหลักการทำงานของโปรแกรมในรูปที่ จ.16 - จ.18

หลักการทำงานของโปรแกรมในรูปที่ จ.16 คือ กำหนดให้พอร์ต B เป็นเอาต์พุตแสดงไฟกระพริบติด-ดับสลับกันและกำหนดค่าปรีสเกลเลอร์เท่ากับ 32 ซึ่งการหน่วงเวลานั้นจะใช้ไทเมอร์ 0 สามารถคำนวณความถี่ได้ 1 เฮิรตซ์

หลักการทำงานของโปรแกรมในรูปที่ จ.17 คือ กำหนดให้พอร์ต B เป็นเอาต์พุตแสดงไฟกระพริบติด-ดับสลับกันและกำหนดค่าปรีสเกลเลอร์เท่ากับ 64 ซึ่งการหน่วงเวลานั้นจะใช้ไทเมอร์ 0 สามารถคำนวณความถี่ได้ 0.5 เฮิรตซ์

หลักการทำงานของโปรแกรมในรูปที่ จ.18 คือ กำหนดให้พอร์ต B เป็นเอาต์พุตแสดงไฟกระพริบติด-ดับสลับกันและกำหนดค่าปรีสเกลเลอร์เท่ากับ 4 ขอบขาลง ซึ่งการหน่วงเวลานั้นจะใช้ไทเมอร์เคาน์เตอร์กับสัญญาณนาฬิกาภายนอกและใช้การอินเตอร์รัพต์ สามารถคำนวณความถี่ได้ 0.5 เฮิรตซ์

3. จงอธิบายการนำไทเมอร์เคาน์เตอร์ไปใช้งานได้อย่างไรบ้างและทำอย่างไร

การใช้ไทเมอร์ 0 สร้างโปรแกรมหน่วงเวลาเพื่อสร้างสัญญาณสี่เหลี่ยม สัญญาณลอจิกสูงจะปรากฏที่เอาต์พุตนานเพียงใดนั้นขึ้นอยู่กับการหน่วงเวลาให้เกิดสัญญาณลอจิกสูงค้างที่เอาต์พุต จากนั้นเขียนข้อมูลลงในรีจีสเตอร์ OPTION เพื่อกำหนดให้ปรีสเกลเลอร์ทำงานร่วมกับ TMR0 และเลือกที่จะใช้สัญญาณนาฬิกาจากแหล่งกำเนิดสัญญาณนาฬิกาภายใน PIC16F876 เอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เฉลยใบงานที่ 5

การเชื่อมต่อ PIC16F876 กับจอแสดงผลแบบผลึกเหลว (LCD)

ลำดับขั้นตอนการทดลอง

1. เขียนโปรแกรมที่ 10.1 ทำการแอสเซมเบลอร์แล้วเขียนข้อมูลลงบน PIC16F876
2. ต่อสายเชื่อมต่อตามรูปที่ จ.28
3. ทำการรันโปรแกรม โดยการคลิกปุ่ม RUN
4. ผลการทำงานของ LCD เป็นดังนี้

จะแสดงข้อความ " PIC 16F8XX Projects"

"ELECTRONICS & COMPUTERS"

5. จากโปรแกรมดังรูปที่ จ.29 ถ้าเปลี่ยนการแสดงข้อความเป็น KMITL ทั้ง 2 บรรทัดจะต้องแก้ไขโปรแกรมอย่างไร

ในบรรทัดที่ 75 เปลี่ยนเป็น call Table2

ในบรรทัดที่ 90 เปลี่ยนเป็น dt "KMITL"

ลบบรรทัดที่ 91-92

6. จากการทดลองข้อ 5 ทำการแอสเซมเบลอร์แล้วเขียนข้อมูลลงบน PIC16F876 อีกครั้ง
7. ทำการรันโปรแกรม โดยการคลิกปุ่ม RUN
8. ผลการทำงานของ LCD เป็นดังนี้

แสดงข้อความ "KMITL ทั้ง 2 บรรทัด

คำถามท้ายการทดลอง

1. จงอธิบายหลักการทำงานของโปรแกรมดังรูปที่ จ.29

โปรแกรมแสดงข้อความบน LCD จะทำโดยการส่งข้อมูลตัวอักษรไปแสดงผลที่ละตัวอักษรแล้วทำการหน่วงเวลาเพื่อให้เห็นการติดทีละตัวอักษรของข้อความ แล้วส่งอักษรต่อไปเรื่อยๆ จนหมดข้อความ แล้วเคลีย LCD เพื่อเริ่มต้นแสดงใหม่อีกครั้ง

2. จงเขียนโปรแกรมเพื่อควบคุมให้ LCD แสดงข้อความเป็นชื่อของตัวเอง

ทำได้โดยการเปลี่ยนข้อความในตาราง ให้เป็นข้อความชื่อตามที่ต้องการ แล้วจบด้วยอักขระ 0x00 เพื่อบอกให้รู้ว่าหมดข้อความแล้ว ให้เริ่มต้นใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เฉลยใบงานที่ 6

การใช้งาน PIC16F876 ขับ LED ตัวเลข 7 ส่วน

ลำดับขั้นตอนการทดลอง

1. ต่อสายเชื่อมต่อระหว่างโมดูลตามรูปที่ จ.32

2. เขียนโปรแกรมดังรูปที่ จ.33 ทำการแอสเซมเบลอร์ จากนั้นเขียนข้อมูลลงบน PIC16F876

3. ทำการรันโปรแกรมบน PIC 16F876 โดยการกดสวิตช์รีเซต สังเกตผลที่เกิดขึ้น

<u>จะเป็นการนับเลข 0-9 ที่หลักแรกของ LED ตัวเลข 7 ส่วน แบบคาโถดร่วม</u>

4. ถ้าเปลี่ยนข้อมูลในโปรแกรมเป็น dt 6f, 7f, 07, 7d, 6d, 66, 4f, 5b, 06, 3f, 00 จากนั้นทำเหมือนขั้นตอนที่ 2 –3 ผลจากการสังเกตเป็นดังนี้

<u>จะเป็นการนับเลข 9-0 ที่หลักแรกของ LED ตัวเลข 7 ส่วน แบบคาโถดร่วม</u>

5. จากโปรแกรมดังรูปที่ จ.33 เปลี่ยนข้อมูลจาก movlw 0x00 มาเป็น movlw 0x01 ผลจากการสังเกตเป็นดังนี้

<u>จะเป็นการนับเลข 0-9 ในหลักที่ 2 ของ LED ตัวเลข 7 ส่วน แบบคาโถดร่วม</u>

6. จากข้อ 5 ถ้าต้องการให้หลักที่ 4 ติด (หลักแรกซ้ายมือ) จะต้องเปลี่ยนข้อมูลเป็นเท่าไร

movlw 0x03

movwf PORTA

7. จากโปรแกรมดังรูปที่ จ.33 เปลี่ยนข้อมูลจาก movlw 0x00 มาเป็น movlw 0x04 และเปลี่ยนข้อมูลจาก dt 3f, 06, 5b, 4f, 66, 6d, 7d, 07, 7f, 6f, 00 มาเป็น dt b0, f9, a4, cf, 99, 92, 82, f8, 80, 90, 00

8. ทำเหมือนขั้นตอนที่ 2 –3 ผลจากการสังเกตเป็นดังนี้

<u>จะเป็นการนับเลข 0-9 ที่หลักแรกของ LED ตัวเลข 7 ส่วน แบบอาโนดร่วม</u>

9. จากข้อ 8 ถ้าเปลี่ยนข้อมูลจาก movlw 0x04 มาเป็น movlw 0x05 ผลจากการสังเกตเป็นดังนี้

<u>จะเป็นการนับเลข 0-9 ในหลักที่ 2 ของ LED ตัวเลข 7 ส่วน แบบอาโนดร่วม</u>

10. จากข้อ 9 ถ้าต้องการให้หลักที่ 4 ติด (หลักแรกซ้ายมือ) จะต้องเปลี่ยนข้อมูลเป็นเท่าไร

movlw 0x07

movwf PORTA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. เขียนโปรแกรมดังรูปที่ จ.34 ทำการแอสเซมเบลอร์ จากนั้นเขียนข้อมูลลงบน PIC16F876

12. ต่อสายเชื่อมต่อระหว่างโมดูลตามรูปที่ จ.32

13. ทำการรันโปรแกรมบน PIC16F876 โดยการกดสวิตช์รีเซต สังเกตผลที่เกิดขึ้น

ที่แบบคาโถดร่วมหลักที่ 1 (ซ้ายมือสุด)แสดงตัวเลข 4

หลักที่ 2 แสดงตัวเลข '3'

หลักที่ 3 แสดงตัวเลข '2'

หลักที่ 4 แสดงตัวเลข '1'

14. ในบรรทัดที่ 16 ถ้าต้องการเปลี่ยนข้อมูลเป็น 0x77 และในบรรทัดที่ 34 เป็น 0xFF สังเกตผลที่เกิดขึ้น

หลักที่ 1 จะแสดง 'A'

หลักที่ 4 จะแสดง 'F'

15. จากโปรแกรมที่ 11.5 ถ้าต้องการให้แสดงผลแบบอาโนดร่วมต้องแก้ไขโปรแกรมอย่างไรบ้าง

จาก movlw 0x00 เป็น movlw 0x04

จาก movlw 0x01 เป็น movlw 0x05

จาก movlw 0x02 เป็น movlw 0x06

จาก movlw 0x03 เป็น movlw 0x07

คำถามท้ายการทดลอง

1. จงอธิบายความแตกต่างระหว่าง LED ตัวเลข 7 ส่วนแบบแคโทดร่วมและแอโนดร่วม

LED ตัวเลข 7 ส่วนแบบแคโทดร่วมจะต้องป้อนไฟบวกให้แต่ละขา และป้อนไฟลบเข้าที่ขาร่วม ส่วนแบบแอโนดร่วมก็จะตรงข้ามกันคือ จะต้องป้อนไฟลบให้แต่ละขาและป้อนไฟบวกเข้าที่ขาร่วม

2. จงเขียนวงจรนับเลข 0-9999 โดยให้แสดงผลใน LED ตัวเลข 7 ส่วนแบบแอโนดร่วม

```
      list p=16f876
      include "p16f876.inc"
    __config _PWRTE_ON & _WDT_OFF & _LVP_OFF & _XT_OSC
Digit1      equ      0x20
Digit2      equ      0x21
Digit3      equ      0x22
Digit4      equ      0x23
Column      equ      0x24
CountD      equ      0x25
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Count1      equ     0x26
Buff1       equ     0x28

            org     0x00
            bsf     STATUS,RP0
            movlw   0x06
            movwf   ADCON1
            clrf    TRISA
            clrf    TRISB
            clrf    TRISC
            bcf     STATUS,RP0
MAIN        clrf    PORTA
            clrf    PORTB               ; common cathode
            movlw   0xff
            movwf   PORTC               ; comon anode
            clrf    Digit1
            clrf    Digit2
            clrf    Digit3
            clrf    Digit4
            clrf    Column
Start       movlw   0x10
            movwf   Count1
Lp1         call    SEG0
            decfsz  Count1,F
            goto    Lp1
            incf    Digit1,F
            goto    Start
;_____________________________________ Loop Segment
SEG0        clrf    Column
            movf    Column,W
            movwf   PORTA
SEG11       movf    Digit1,W
            call    SEGMENT
            movwf   Buff1
            sublw   0x00
            btfss   STATUS,Z
            goto    SEG12
            clrf    Digit1
            incf    Digit2,F
            goto    SEG11
SEG12       movf    Buff1,W
            movwf   PORTB
            comf    Buff1,W
            movwf   PORTC
            call    Delay
            movf    Column,W
            addlw   0x04
            movwf   PORTA
            call    Delay
            clrf    PORTB
            movlw   0xff
            movwf   PORTC
            incf    Column,F
            movf    Column,W
            movwf   PORTA
SEG21       movf    Digit2,W
            call    SEGMENT
            movwf   Buff1
            sublw   0x00
            btfss   STATUS,Z
            goto    SEG2
            clrf    Digit2
            incf    Digit3,F
            goto    SEG21
SEG2        movf    Buff1,W
            movwf   PORTB
            comf    Buff1,W
            movwf   PORTC
            call    Delay
            movlw   0x04
            addwf   Column,W
            movwf   PORTA
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            call    Delay
            clrf    PORTB
            movlw   0xff
            movwf   PORTC
            incf    Column,F
            movf    Column,W
            movwf   PORTA
SEG31       movf    Digit3,W
            call    SEGMENT
            movwf   Buff1
            sublw   0x00
            btfss   STATUS,Z
            goto    SEG32
            clrf    Digit3
            incf    Digit4,F
            goto    SEG31
SEG32       movf    Buff1,W
            movwf   PORTB
            comf    Buff1,W
            movwf   PORTC
            call    Delay
            movlw   0x04
            addwf   Column,W
            movwf   PORTA
            call    Delay
            clrf    PORTB
            movlw   0xff
            movwf   PORTC
            incf    Column,F
            movf    Column,W
            movwf   PORTA
SEG41       movf    Digit4,W
            call    SEGMENT
            movwf   Buff1
            sublw   0x00
            btfss   STATUS,Z
            goto    SEG42
            clrf    Digit4
            goto    SEG41
SEG42       movf    Buff1,W
            movwf   PORTB
            comf    Buff1,W
            movwf   PORTC
            call    Delay
            movlw   0x04
            addwf   Column,W
            movwf   PORTA
            call    Delay
            clrf    PORTB
            movlw   0xff
            movwf   PORTC
            return
SEGMENT     addwf   PCL,F
            dt      3f, 06, 5b, 4f, 66, 6d, 7d, 07
            dt      7f, 6f, 00

Delay       movlw   0xff
            movwf   CountD
Del1        decfsz  CountD,F
            goto    Del1
            return

            end
```

รูปที่ ฉ.4 เฉลยโปรแกรมนับเลข 0000 ถึง 9999 แสดงบน Seven Segment

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เฉลยใบงานที่ 7

การเชื่อมต่อ PIC16F876 กับสวิตช์เมตริกซ์ขนาด 4x4

ลำดับขั้นตอนการทดลอง

1. เขียนโปรแกรมดังรูปที่ จ.39 ทำการแอสเซมเบลอร์
2. ต่อสายเชื่อมต่อระหว่างโมดูลตามรูปที่ จ.38
3. เขียนข้อมูลลงบน PIC16F876 ทำการรันโปรแกรมบน PIC16F876 โดยคลิกปุ่ม RUN
4. เมื่อกดสวิตช์แต่ละตัว ผลที่ได้คือ

ตารางที่ ฉ.2 ผลการติดของ LED ตัวเลข 7 ส่วนจากการกดสวิตช์

กดสวิตช์	แสดงผลที่ LED ตัวเลข 7 ส่วน
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	A
B	B
C	C
D	D
E	E
F	F

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำถามท้ายการทดลอง

1. จงอธิบายหลักการทำงานของโปรแกรมดังรูปที่ จ.39

โปรแกรมรับค่าจากการกดสวิตช์เมตริกซ์ แล้วแสดงผลตามตัวเลขที่กด ทำงานโดยการสแกนคีย์ ไปทีละแถว โดยมีตัวแปรสำหรับบอกค่าปุ่มที่กด ตัวแปรจะเพิ่มค่าขึ้นเรื่อยๆ ถ้าหากไม่มีการกด ทุกๆ ปุ่มที่ทำการสแกน ก็จะทำให้ค่าที่อยู่ในตัวแปรดังกล่าวเพิ่มขึ้นไปตรงกับปุ่มที่สแกน และเมื่อเกิดการกดปุ่มก็จะนำค่าดังกล่าวไปในเปิดตารางแล้วแสดงผลได้

2. จงอธิบายหลักการทำงานของสวิตช์เมตริกส์ขนาด 4X4 มาพอเข้าใจ

การทำงานของสวิตช์เมตริกซ์ขนาด 4x4 จะมีการสแกนแถวทีละแถวแล้ว ตรวจสอบคีย์ในแถวนั้น ว่าถูกกดหรือไม่ คือตรวจสอบทางคอลัมน์นั่นเอง ถ้าหากมีการกดก็จะอ่านค่าของคอลัมน์ได้ตรงตามปุ่มที่กด แล้วสแกนแถวต่อไปเรื่อยๆ ตลอดเวลา

3. จงเขียนโปรแกรมรับค่าจากสวิตช์เมตริกซ์ 4X4 ที่เข้ามาที่พอร์ต C โดยให้พอร์ต C ทางด้าน 4 บิตล่างเป็นเอาต์พุตและ 4 บิตบนเป็นอินพุต

```
  List p=16f876
  #Include "p16f876.inc"
__config _WDT_OFF & _PWRTE_ON & _LVP_OFF & _XT_OSC
COUNT       equ     0x21
COUNTD      equ     0x22
            org     0x000
            bsf     STATUS,RP0
            movlw   0x06
            movwf   ADCON1
            movlw   0xff
            movwf   PORTA
            clrf    PORTB
            movlw   b'11110000'
            movwf   PORTC
            bcf     STATUS,RP0
            movlw   0xFF
            movwf   PORTC

Start       bcf     PORTC,0
            call    Key
            bsf     PORTC,0
            bcf     PORTC,1
            call    Key
            bsf     PORTC,1
            bcf     PORTC,2
            call    Key
            bsf     PORTC,2
            bcf     PORTC,3
            call    Key
            bsf     PORTC,3
            goto    Start

Key         movlw   0x00
            movwf   COUNT
            btfss   PORTC,4
            call    Display
            incf    COUNT,F
            btfss   PORTC,5
            goto    Display
            incf    COUNT,F
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            btfss    PORTC,6
            goto     Display
            incf     COUNT,F
            btfss    PORTC,7
            goto     Display
            return

Display     clrf     PORTA
            movf     COUNT,W
            call     DATA1
            movwf    PORTB
            call     Delay
            movlw    0x01
            movwf    PORTA
            return
DATA1       addwf    PCL,F
            dt       3f, 06, 5b, 4f, 66, 6d, 7d, 07
            dt       7f, 6f, 77, 7c, 39, 5e, 79, 71 ,00

Delay       movlw    0x02
            movwf    COUNTD
Del1        decfsz   COUNTD,F
            goto     Del1
            return
            end
```

รูปที่ ฉ.5 เฉลยโปรแกรมสแกนคีย์ สวิตช์เมตริกซ์ 4x4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เฉลยใบงานที่ 8
การเชื่อมต่อ PIC16F876 กับมอเตอร์ไฟฟ้ากระแสตรง (DC Motor)

ลำดับขั้นตอนการทดลอง

1. เขียนโปรแกรมดังรูปที่ จ.42 ทำการแอสเซมเบลอร์ จากนั้นเขียนข้อมูลลงบน PIC16F876

2. ต่อสายเชื่อมต่อระหว่างโมดูลตามรูปที่ จ.41

3. ทำการรันโปรแกรมบน PIC16F876 โดยการกดสวิตช์รีเซต สังเกตผลที่เกิดขึ้น

มอเตอร์จะหมุนเมื่อมีการกดสวิตช์ SW1 และ SW2

4. เมื่อกดสวิตช์ SW1 ผลที่ได้คือ

มอเตอร์จะหมุนเมื่อมีการกดสวิตช์ SW1 และ SW2

5. เมื่อกดสวิตช์ SW2 ผลที่ได้คือ

มอเตอร์จะหมุนในทิศทางตามเข็มนาฬิกา

6. เมื่อกดสวิตช์ SW3 ผลที่ได้คือ

มอเตอร์จะหมุนในทิศทางทวนเข็มนาฬิกา

7. เมื่อกดสวิตช์ SW4 ผลที่ได้คือ

มอเตอร์จะหยุดหมุน

คำถามท้ายการทดลอง

1. จงอธิบายหลักการทำงานของโปรแกรมในรูปที่ จ.40

หลักการทำงานของโปรแกรมในรูปที่ จ.40 คือ กำหนดให้พอร์ต A เป็นอินพุตเพื่อใช้ในการตรวจสอบสวิตช์ ส่วนพอร์ต B เป็นเอาต์พุตเพื่อใช้ในการกระตุ้นขดลวดของมอเตอร์เพื่อให้มอเตอร์หมุนได้ ดังนั้นเมื่อกดสวิตช์ SW1 และ SW4 มอเตอร์จะไม่หมุน ถ้ากำหนดสวิตช์ SW2 จะทำให้มอเตอร์หมุนไปในทิศทางหนึ่งถ้ากดสวิตช์ SW3 จะทำให้มอเตอร์หมุนกลับทิศทางกัน

2. จงเขียนโปรแกรมควบคุมมอเตอร์ไฟฟ้ากระแสตรงโดยมีระดับความเร็วของมอเตอร์ 4 ระดับคือ กดสวิตช์ SW1 ให้หมุนช้า กด SW2 ให้หมุนเร็ว กด SW3 ให้หมุนเร็วขึ้น และสุดท้ายกด SW4 จะหมุนเร็วที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            List p=16f877
            #include "p16f877.inc"
            __config _PWRTE_ON & _WDT_OFF & _LVP_OFF & _XT_OSC
tempSw      equ     0x20
            radix   dec

            org     0x00
            bsf     STATUS,RP0
            movlw   0x06
            movwf   ADCON1
            movlw   0xff
            movwf   TRISA
            clrf    TRISB
            movlw   b'11111001'
            movwf   TRISC           ; make drive DC Moter
            bcf     STATUS,RP0
            movlw   249             ; 249:PWM Period = 250Hz
            movwf   PR2
            movlw   125             ; Duty cycle 50 %
            movwf   CCPR1L          ; value of (CCPR1L:CCP1CON<5:4>) = 500
            movlw   b'00001100'
            movwf   CCP1CON         ; CCP1X:CCP1Y = 00, PWM mode
            clrf    TMR2
            movlw   b'00000010'     ; Postscale (no use), Timer2 OFF, Prescaler 16
            movwf   T2CON
Start       bsf     PORTC,1
            bsf     T2CON,TMR2ON
loop        movf    CCPR1L,W
            movwf   PORTB
            call    scank
            andlw   0xff
            btfsc   STATUS,Z
            goto    loop
            sublw   1
            btfss   STATUS,Z
            goto    speedup
speeddown   movf    CCPR1L,W
            sublw   15
            btfsc   STATUS,C
            goto    Continue
            movlw   10
            subwf   CCPR1L,F
            goto    Continue
speedup     movf    CCPR1L,W        ; test CCPR1L > 245 ?
            sublw   245             ;   245 - CCPR1L = ?
            btfss   STATUS,C        ;   test carry
            goto    Continue        ; yes, no action
            movlw   10              ; no, add 10
            addwf   CCPR1L,F
            goto    Continue
;-------------- scan key --------------
scank       movlw   0
            btfss   PORTA,0
            movlw   1
            btfss   PORTA,1
            movlw   2
            subwf   tempSw,F
            movwf   tempSw
            btfsc   STATUS,C
            retlw   0
            return
            end
```

รูปที่ ฉ.6 เฉลยโปรแกรมควบคุมความเร็วดีซีมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เฉลยใบงานที่ 9
การเชื่อมต่อ PIC16F876 กับวงจรมอดูเลชัน ทางความกว้างของพัลส์ (PWM)

ลำดับขั้นตอนการทดลอง

1. เขียนโปรแกรมที่ 14.1 ทำการแอสเซมเบลอร์แล้วเขียนข้อมูลลงบน PIC16F876
2. ต่อสายเชื่อมต่อตามรูปที่ จ.46
3. ทำการรันโปรแกรม โดยการคลิกปุ่ม RUN
4. สังเกตการหมุนของมอเตอร์ และ LED P1 และ P2 ในโมดูลขับมอเตอร์

LED ติดหรือไม่ P1 ไม่ติด P2 ติด

มอเตอร์หมุนทิศทางใด ตามเข็มนาฬิกา

5. กดสวิตช์ SW0 ในโมดูลเบสิกอินพุต 5 ครั้ง สังเกตการเปลี่ยนแปลงความเร็วมอเตอร์เปลี่ยนแปลงอย่างไร มอเตอร์หมุนช้าลง

LED เปลี่ยนแปลงอย่างไร P1 ไม่ติด P2 สว่างน้อยลง

6. กดสวิตช์ SW1 ในโมดูลเบสิกอินพุต 10 ครั้ง สังเกตการเปลี่ยนแปลงความเร็วมอเตอร์เปลี่ยนแปลงอย่างไร มอเตอร์หมุนเร็วขึ้นกว่าครั้งแรก

LED เปลี่ยนแปลงอย่างไร P1 ไม่ติด P2 สว่างมากขึ้น

7. จากโปรแกรม การเปลี่ยนแปลงค่าในรีจีสเตอร์ CCPR1L มีผลต่อการเปลี่ยนแปลงความเร็วของมอเตอร์อย่างไร จงอธิบาย

ค่าในรีจีสเตอร์ CCPRL1 เป็นตัวที่ใช้กำหนดค่า Duty Cycle ของสัญญาณ PWM ซึ่งจะส่งผลให้แรงดันไฟเฉลี่ยที่มอเตอร์ได้รับเปลี่ยนแปลง ทำให้ความเร็วของมอเตอร์เปลี่ยนแปลงไป

คำถามท้ายการทดลอง

1. จงอธิบายการทำงานของโปรแกรมที่ จ.47

โปรแกรมควบคุมความเร็วมอเตอร์โดยใช้ PWM มีหลักการคือ เปลี่ยนค่าในรีจีสเตอร์ CCPRL1 ซึ่งเป็นตัวกำหนด Duty Cycle ก็จำทำให้ความเร็วของมอเตอร์เปลี่ยนไป

2. จงอธิบายส่วนสำคัญของการเขียนโปรแกรม เพื่อควบคุมให้ PIC16F876 ทำงานเป็นวงจร PWM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การอีนาเบิลโมดูล PWM ใน PIC16F876 จากนั้นทำการคำนวณหาค่า ที่ต้องการจากความถี่ของสัญญาณ และค่า Duty Cycle และคำนวณความเป็นไปได้ของสัญญาณ เพื่อกำหนดค่าใน Timer 2 ก็จะสามารถสร้างสัญญาณ PWM ตามต้องการได้

3. จงยกตัวอย่างการนำวงจร PWM ที่มี PIC16F876 เป็นอุปกรณ์หลักไปใช้งาน
   1. วงจรควบคุมแสงสว่างของ LED
   2. วงจรควบคุมความเร็วมอเตอร์ DC
   3. วงจรสร้างสัญญาณ PWM แบบปรับความถี่ และ Duty Cycle ได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เฉลยใบงานที่ 10
การเชื่อมต่อ PIC16F876 กับสเตปปิ้งมอเตอร์ (Stepping Motor)

ลำดับขั้นตอนการทดลอง

1. เขียนโปรแกรมดังรูปที่ จ.55 ทำการแอสเซมเบลอร์ จากนั้นเขียนข้อมูลลงบน PIC16F876

2. ต่อสายเชื่อมต่อระหว่างโมดูลตามรูปที่ จ.54

3. ทำการรันโปรแกรมบน PIC16F876 โดยการกดสวิตช์รีเซต สังเกตผลที่เกิดขึ้น

4. เมื่อกดสวิตช์ SW1 ผลที่ได้คือ

มอเตอร์จะหมุนไปในทิศทางตามเข็มนาฬิกา

5. เมื่อกดสวิตช์ SW2 ผลที่ได้คือ

มอเตอร์จะหมุนไปในทิศทางทวนเข็มนาฬิกา

6. เขียนโปรแกรมดังรูปที่ จ.56 ทำการแอสเซมเบลอร์ จากนั้นเขียนข้อมูลลงบน PIC16F876

7. ต่อสายเชื่อมต่อระหว่างโมดูลตามรูปที่ จ.54

8. ทำการรันโปรแกรมบน PIC16F876 โดยการกดสวิตช์รีเซต สังเกตผลที่เกิดขึ้น

9. เมื่อกดสวิตช์ SW1 ผลที่ได้คือ

มอเตอร์จะหมุนไปในทิศทางตามเข็มนาฬิกา

10. เมื่อกดสวิตช์ SW2 ผลที่ได้คือ

มอเตอร์จะหมุนไปในทิศทางทวนเข็มนาฬิกา

คำถามท้ายการทดลอง

1. จงอธิบายการทำงานของโปรแกรมที่ จ.48 มาพอเข้าใจ

หลักการทำงานของโปรแกรมในรูปที่ จ.48 คือ กำหนดให้พอร์ต A เป็นอินพุตเพื่อตรวจสอบการกดสวิตช์โดยที่กดสวิตช์ SW1 จะหมุนไปในทิศทางหนึ่ง ในทางกลับกันกดสวิตช์ SW2 มอเตอร์ก็จะหมุนไปในทิศทางตรงข้ามและกำหนดให้พอร์ต B เป็นเอาต์พุตเพื่อขับสเตปปิ้งมอเตอร์ ซึ่งเป็นแบบเวฟคือ ป้อน 1 ที่ละบิตเรียงกันไปเรื่อยๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. จากโปรแกรมที่ จ.48 เป็นการกระตุ้นและควบคุมการหมุนสเตปปิ้งมอเตอร์แบบเวฟ เพราะเหตุใด

เป็นการกระตุ้นแบบเวฟเนื่องจากเป็นการป้อนข้อมูลที่พอร์ต B เป็นการป้อน 1 เพื่อไปกระตุ้นทีละขดลวดไล่เรียงกันไป ซึ่งตรงกับการกระตุ้นแบบเวฟของมอเตอร์ และจากโปรแกรมจะมีทั้งกระตุ้นให้มอเตอร์หมุนตามเข็มและทวนเข็มนาฬิกา

3. จงเขียนโปรแกรมควบคุมเสต็ปมอเตอร์แบบครึ่งสเตปโดยมีระดับความเร็วของมอเตอร์ 4 ระดับคือกด SW1 ให้หมุนช้า กด SW2 ให้หมุนเร็ว กด SW3 ให้หมุนเร็วขึ้น และกด SW4 จะหมุนเร็วที่สุด

ทำโดยการเปลี่ยนข้อมูลในตารางเป็นข้อมูลของแบบครึ่งสเตปแทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เฉลยใบงานที่ 11

การใช้งาน PIC16F876 แปลงสัญญาณแอนะลอกเป็นดิจิตอล

ลำดับขั้นตอนการทดลอง

1. เขียนโปรแกรมดังรูปที่ จ.64 ทำการแอสเซมเบลอร์แล้วเขียนข้อมูลลงบน PIC 16F876
2. ต่อสายเชื่อมต่อตามรูปที่ จ.62
3. ทำการรันโปรแกรม โดยการคลิกปุ่ม RUN
4. ปรับตัวต้านทาน AN0 ไปที่ตำแหน่งกึ่งกลาง
5. สังเกตผลที่ได้เป็นดังนี้

LED จะแสดง “01110110” แปลงเป็นฐานสิบได้ 118

6. ปรับตัวต้านทาน AN0 ไปที่ตำแหน่งซ้ายสุด
7. สังเกตผลที่ได้เป็นดังนี้

LED จะแสดง “00000000”

8. ปรับตัวต้านทาน AN0 ไปที่ตำแหน่งขวาสุด
9. สังเกตผลที่ได้เป็นดังนี้

LED จะแสดง “11111111” แปลงเป็นฐานสิบได้ 255

10. เขียนโปรแกรมดังรูปที่ จ.65 ทำการแอสเซมเบลอร์แล้วเขียนข้อมูลลงบน PIC 16F876
11. ปรับตัวต้านทาน AN0 ไปที่ตำแหน่งกึ่งกลางเช่นเดียวกับข้อ 4.
12. สังเกตผลที่ได้เมื่อเทียบกับข้อ 4. เป็นดังนี้

LED 7 segment จะแสดงเป็นตัวเลขฐานสิบ 125

13. ปรับตัวต้านทาน AN0 ไปที่ตำแหน่งซ้ายสุดเช่นเดียวกับข้อ 6.
14. สังเกตผลที่ได้เมื่อเทียบกับข้อ 6. เป็นดังนี้

LED 7 segment จะแสดงเป็นตัวเลขฐานสิบ 0

15. ปรับตัวต้านทาน AN0 ไปที่ตำแหน่งซ้ายสุดเช่นเดียวกับข้อ 8.
16. สังเกตผลที่ได้เมื่อเทียบกับข้อ 8. เป็นดังนี้

LED 7 segment จะแสดงเป็นตัวเลขฐานสิบ 255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำถามท้ายการทดลอง

1. จงอธิบายหลักการแปลงสัญญาณแอนะลอกเป็นดิจิตอลทั้ง 3 แบบมาพอเข้าใจ

หลักการพื้นฐานของการแปลงสัญญาณแอนะลอกเป็นดิจิตอลคือ จะต้องมีการแซมปลิ้งสัณญาณแอนะลอกด้วยความถี่ที่มากกว่าความถี่ของสัญญาณเป็นสองเท่า แล้วทำการจัดระดับให้เข้ากับระดับของเลขฐานสองและจำนวนบิตของระบบ จากนั้นก็ทำการแปลงระดับสัญญาณให้เป็นเลขฐานสอง

2. จงอธิบายหลักการทำงานโปรแกรมดังรูปที่ จ.64

ในไมโครคอนโทรลเลอร์ PIC16F876 มีวงจรแปลงสัญญาณแอนะลอกเป็นดิจิตอลอยู่ภายใน สามารถทำการป้อนอินพุตที่เป็นแอนะลอกเข้าไป การเขียนโปรแกรมก็จะไปอ่านผลลัพธ์จากรีจีสเตอร์ที่ทำการแปลง แล้วนำออกแสดงผลยัง LED

3. จงเขียนโปรแกรมเพื่อแปลงสัญญาณแอนะลอกเป็นดิจิตอลแบบ 8 บิต แสดงผลทาง LED 7 segment

```
list p=16f876
include "p16f876.inc"
__config _WDT_OFF & _PWRTE_ON & _LVP_OFF & _XT_OSC

B_ADRESH    equ     0x20
B_ADRESL    equ     0x21
Digit1      equ     0x22
Digit2      equ     0x23
Digit3      equ     0x24
Digit4      equ     0x25
B_Delay1    equ     0x26
Refresh1    equ     0x27
Refresh2    equ     0x28

            org     0x00
IniPort     bsf     STATUS,RP0
            movlw   0x0e
            movwf   ADCON1              ; Left Justified, Analog Bit0
            movlw   0xff
            movwf   TRISA
            clrf    TRISB
            clrf    TRISC
            bcf     STATUS,RP0
            movlw   b'01000001'    ; Fosc/8, Channel0, A/D On
            movwf   ADCON0
            clrf    PORTB
            clrf    PORTC

Start       call    ADConvt
            call    BinToBCD
            movlw   0x50
            movwf   Refresh1
            movlw   0xff
Refrsh1     movwf   Refresh2
            call    Display
Refrsh2     decfsz  Refresh2,F
            goto    Refrsh2
            decfsz  Refresh1,F
            goto    Refrsh1
            goto    Start
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;_____________________________________A to D Convertion
ADConvt     btfsc   ADCON0,2
            goto    ADConvt
            bsf     STATUS,RP0
            movf    ADRESL,W
            bcf     STATUS,RP0
            movwf   B_ADRESL
            movf    ADRESH,W
            movwf   B_ADRESH
            bsf     ADCON0,2
            return

;_____________________________________Convert ADResult Binary 10 Bit to BCD 4
Digit
BinToBCD    clrf    Digit1
            clrf    Digit2
            clrf    Digit3
            clrf    Digit4
Con0        movlw   0xfa                ; Digit4  ADResult - 1000
            subwf   B_ADRESH,W
            btfss   STATUS,C
            goto    Con1
            movlw   0xfa
            subwf   B_ADRESH,F
            incf    Digit4,F
            goto    Con0
Con1        movlw   0x19                ; Digit3  Remain - 100
            subwf   B_ADRESH,W
            btfss   STATUS,C
            goto    ConShift
            movlw   0x19
            subwf   B_ADRESH,F
            incf    Digit3,F
            goto    Con1
ConShift    rlf     B_ADRESL,F          ; Shift Left 2 Bit
            rlf     B_ADRESH,F
            rlf     B_ADRESL,F
            rlf     B_ADRESH,F
Con2        movlw   0x0a                ; Digit2  Remain - 10
            subwf   B_ADRESH,W
            btfss   STATUS,C
            goto    Con3
            movlw   0x0a
            subwf   B_ADRESH,F
            incf    Digit2,F
            goto    Con2
Con3        movf    B_ADRESH,W      ; Digit1  Remain
            movwf   Digit1
            return

;_____________________________________Display Seven Segmant 4 Digit
Display
            clrf    PORTB
            clrf    PORTC
            movlw   Digit1
            movwf   FSR
DP1         movf    INDF,W              ; Digit1
            call    Table7Seg
            movwf   PORTB
            call    Delay
            clrf    PORTB
            incf    FSR,F
            incf    PORTC,F
            btfss   PORTC,2
            goto    DP1
            clrf    PORTC
            return

Table7Seg   addwf   PCL,F
            dt      3f, 06, 5b, 4f, 66, 6d, 7d, 07
            dt      7f, 6f, 00
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;_____________________________________________Delay
Delay       movlw    0x80
            movwf    B_Delay1
Dly0        decfsz   B_Delay1,F
            goto     Dly0
            return


            end
```

รูปที่ ฉ.7 เฉลยการเขียนโปรแกรมแปลงสัญญาณ A to D แสดงผลที่ Seven Segment

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เฉลยใบงานที่ 12
การใช้งาน PIC16F876 กับวงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอก

ลำดับขั้นตอนการทดลอง

1. เขียนโปรแกรมดังรูปที่ จ.71 ทำการแอสเซมเบลอร์แล้วเขียนข้อมูลลงบน PIC16F876
2. ต่อสายเชื่อมต่อตามรูปที่ จ.70
3. ทำการรันโปรแกรม โดยการคลิกปุ่ม RUN
4. ทำการกดดิพสวิตช์ และสังเกตการแสดงผลที่โมดูลเครื่องมือวัด

ตารางที่ ฉ.3 ผลการกดดิพสวิตช์เมื่อใช้วงจร DAC

ลำดับที่	การกดสวิตช์	แสดงผล
1	00000000	0
2	00000011	0.05
3	00000110	0.10
4	00001100	0.25
5	00011000	0.55
6	00110000	1.15
7	01100000	2.25
8	11000000	4.50

5. ทำการทดลองซ้ำอีกครั้ง แต่เปลี่ยนคอนเนคเตอร์ในโมดูลแปลงสัญญาณจากวงจร DAC0808 เป็น R-2R Ladder

6. ทำการกดดิพสวิตช์ และสังเกตการแสดงผลที่โมดูลเครื่องมือวัดเมื่อเทียบกับการทดลองข้อ 4.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ฉ.4 ผลการกดดิพสวิตช์เมื่อใช้วงจร R-2R Ladder

ลำดับที่	การกดสวิตช์	แสดงผล
1	00000000	0
2	00000011	0.10
3	00000110	0.20
4	00001100	0.45
5	00011000	0.95
6	00110000	1.85
7	01100000	3.75
8	11000000	7.50

คำถามท้ายการทดลอง

1. จงอธิบายโปรแกรมดังรูปที่ จ.71

เป็นโปรแกรมรับข้อมูลจากดิพสวิตช์ส่งไปให้วงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอก

2. จงอธิบายหลักการทำงานของการแปลงสัญญาณดิจิตอลเป็นแอนะลอกแบบ R-2R Ladder มาพอเข้าใจ

ทำโดยการนำสัญญาณดิจิตอลมาผ่านตัวต้านทานที่ต่อกับเป็นขั้นบันได สัญญาณที่ต้องผ่านตัวต้านทานมาก นั้นคือบิต LSB จะทำให้สัญญาณถูกลดทอนไปมาก ส่วนสัญญาณ LSB ผ่านค่าต้านทานน้อยก็จะทำให้ผลแรงดัน DC ที่ได้มีค่ามาก

3. จงอธิบายความแตกต่างของการแปลงสัญญาณดิจิตอลเป็นแอนะลอกแบบที่ใช้ไอซี DAC0808 และแบบ R-2R Ladder มาพอเข้าใจ

การแปลงโดยใช้วงจรทั้งสองแบบ จะไม่มีความแตกต่างในการใช้งานแต่แบบ ใช้ไอซีจะให้ผลที่ดีกว่า เนื่องจากสัญญาณรบกวนน้อย มีเสถียรภาพมากกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เฉลยใบงานที่ 13
การใช้งาน PIC16F876 เชื่อมต่อกับโซลิดสเตทรีเลย์
(Solid State Relay : SSR)

ลำดับขั้นตอนการการทดลอง

1. เขียนโปรแกรมดังรูปที่ จ.73 ทำการแอสเซมเบลอร์แล้วเขียนข้อมูลลงบน PIC16F876
2. ต่อสายเชื่อมต่อตามรูปที่ จ.73
3. นำหลอดไฟ 8 ดวงมาต่อเข้ากับ LOAD1 ถึง LOAD8 ของโมดูลโซลิดสเตทรีเลย์
4. ทำการรันโปรแกรม โดยคลิกปุ่ม RUN
5. สังเกตหลอดไฟทั้ง 8 ดวง ผลเป็นดังนี้

หลอดไฟทุกดวงจะกระพริบ

6. จากโปรแกรมดังรูปที่ จ.73 ถ้าเปลี่ยนข้อมูลจาก movlw 0xff เป็น movlw 0xaa หลอดไฟทั้ง 8 ดวงจะเกิดการเปลี่ยนแปลงดังนี้

หลอดไฟ 4 บิตบนและ 4 บิตล่างจะติดดับสลับกัน

คำถามท้ายการทดลอง

1. จงบอกข้อดีและข้อเสียของโซลิดสเตทรีเลย์มาพอเข้าใจ

ข้อดีของโซสิดสเตทรีเลย์คือ ไม่มีส่วนของกลไกทำให้อายุการใช้งานยาวนาน, สามารถทำงานที่ความเร็วได้

ข้อเสียคือ ราคาแพง ไม่สามารถใช้เปิดปิดแรงดันดีซีได้

2. จงเขียนโปรแกรมควบคุมให้หลอดไฟวิ่งจากซ้ายไปขวาเมื่อวิ่งครบแล้วให้วิ่งจากขวาไปซ้าย ทำเช่นนี้ไปเรื่อยๆ

```
            List p=16f876
            #Include "p16f876.inc"
            __config _WDT_OFF & _PWRTE_ON & _LVP_OFF & _XT_OSC
Count1      equ     0x20
Count2      equ     0x21
            org     0x00
            bsf     STATUS,RP0
            movlw   0x00            ;พอร์ต B เป็นเอาต์พุต
            movwf   PORTB
            bcf     STATUS,RP0      ;เคลียร์ STATUS
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Start       movlw    0xff          ;LED ดับทุกดวง
            movwf    PORTB
            call     Delay
            movlw    0x00          ;LED ดับทุกดวง
            movwf    PORTB
            call     Delay
            goto     Start
;---------------------------------- หน่วงเวลา
Delay       movlw    0Xff
            Movwf    Count1
Loop2       movlw    0Xff
            Movwf    Count2
Loop1       decfsz   Count2,F
            Goto     Loop1
            decfsz   Count1,F
            goto     Loop2
            return
            end
```

รูปที่ ฉ.8 เฉลยโปรแกรมใช้โซลิดสเตทรีเลย์ทำไฟวิ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เฉลยใบงานที่ 14
การสื่อสารข้อมูลกับคอมพิวเตอร์ผ่านพอร์ตอนุกรม RS-232

ลำดับขั้นตอนการทดลอง

1. เขียนโปรแกรมรูปที่ จ.78 ทำการแอสเซมเบลอร์แล้วเขียนข้อมูลลงบน PIC16F876
2. ต่อสายเชื่อมต่อตามรูปที่ จ.77
3. เปิดโปรแกรมสื่อสารข้อมูลอนุกรม MSCOM.EXE
4. คลิกเมนู CommPort/Properties
5. เลือกพอร์ตอนุกรมที่ต่อสายสัญญาณข้อมูลอยู่
6. ทำการเลือก Maximum Speed = 2400, Data Bits = 8, Parlity = None และ Stop bits = 1 จากนั้นคลิก OK
7. ทำการรันโปรแกรม โดยการคลิกปุ่ม RUN
8. สังเกตที่โปรแกรมสื่อสารข้อมูลอนุกรม MSCOM.EXE

แสดงข้อความ "ชุดปฏิบัติการทดลองไมโครคอนโทรลเลอร์ PIC16F8XX"
"PIC16F8XX Experimentation Practice Set"
"การสื่อสารข้อมูลอนุกรม RS-232"

9. ทดลองกดปุ่มตัวอักษรบนคีย์บอร์ด และสังเกตการแสดงผลได้จาก LED บนโมดูลเบสิกอินพุตเอาต์พุต

ตารางที่ จ.5 ผลการกดคีย์ตัวอักษรบนคีย์บอร์ด

คีย์ที่กด	LED แสดงผล (ฐานสิบหก)
A	0x41
K	0x4B
D	0x64
H	0x68
P	0x50
M	0x6D
Z	0x5A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10.ทดลองกดสวิตช์ใดๆ บนโมดูลเบสิกอินพุตเอาต์พุต และสังเกตการแสดงผลบนโปรแกรม MSCOM.EXE

ตารางที่ ฉ.6 ผลการกดสวิตช์ที่โมดูลเบสิกอินพุตเอาต์พุต

สวิตช์ที่กด	ผลที่ปรากฏบนโปรแกรม MSCOM.EXE
SW1	1
SW2	2
SW3	3
SW4	4
SW5	5

คำถามท้ายการทดลอง

1. จงอธิบายหลักการสื่อสารข้อมูลกับคอมพิวเตอร์ผ่านพอร์ตอนุกรม RS-232 ได้

ในไมโครคอนโทรลเลอร์ PIC16F876 มี USART สำหรับสื่อสารข้อมูลแบบอนุกรมทั้งแบบ Synchronous และแบบ Asynchronous ได้ ซึ่งสามารถกำหนดอัตราเร็วในการส่งข้อมูล และขาสัญญาณสำหรับการเชื่อมต่อ

2. ถ้าต้องการเปลี่ยน Baud rate ในการสื่อสารข้อมูลจะต้องเปลี่ยนค่าในรีจีสเตอร์ใด

รีจีสเตอร์ SPBRG : Baud Rate Generator Register

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

กฤษดา ใจเย็น และคณะ. **เรียนรู้ไมโครคอนโทรลเลอร์อย่างง่ายกับเบสิกแสตมป์ 2**. กรุงเทพฯ : อินโนเวตีฟ เอ็กเพอร์ริเมนต์. ม.ป.ป.

กฤษดา ใจเย็น และชัยวัฒน์ ลิ้มพรจิตวิไล. **เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ PIC16F84**. กรุงเทพฯ : อินโนเวตีฟ เอ็กเพอร์ริเมนต์. ม.ป.ป.

กนก กุสุมาลย์นุกูล และไกลวุฒิ มั่นเสถียรสิน. **คู่มือการเขียนโปรแกรม Delphi4**. พิมพ์ครั้งที่ 2. กรุงเทพฯ : ซัคเซส มีเดีย. ม.ป.ป.

นพพร วัฒนสิทธิ์. "**ชุดฝึกไมโครคอนโทรลเลอร์ 8051**". ปริญญานิพนธ์ครุศาสตร์อุตสาหกรรมบัณฑิต สาขาอิเล็กทรอนิกส์และคอมพิวเตอร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.2541

Microchip. **Data sheet**. [Online]. Available : http://www.microchip.com/10/lit/pline/picmicro/index.htm. 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อผู้ทำปริญญานิพนธ์	นางสาวจันจิรา เงินรวง
วันเดือนปีเกิด	27 กุมภาพันธ์ 2521
สถานที่เกิด	จังหวัดนครราชสีมา
ภูมิลำเนาเดิม	จังหวัดนครราชสีมา
ที่อยู่ปัจจุบัน	17 ม. 12 ต. ดอนตะหนิน อ.บัวใหญ่ จ. นครราชสีมา 30120
โทรศัพท์	-

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนบ้านทองหลางน้อย
มัธยมศึกษาตอนต้น	โรงเรียนบัวใหญ่
ประกาศนียบัตรวิชาชีพ (ปวช.)	-
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	สถาบันเทคโนโลยีราชมงคล วิทยาเขตภาคตะวันออกเฉียงเหนือ นครราชสีมา
ปริญญาตรี	สาขาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับรางวัล	-
ทุนการศึกษา	-
คติพจน์	ปัญหาทุกปัญหาย่อมมีทางแก้เสมอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อผู้ทำปริญญานิพนธ์	นางสาวสุธาสินี สว่างศรี
วันเดือนปีเกิด	6 พฤษภาคม 2523
สถานที่เกิด	จังหวัดนครราชสีมา
ภูมิลำเนาเดิม	จังหวัดนครราชสีมา
ที่อยู่ปัจจุบัน	104/1 หมู่ 1 ตำบล ในเมือง อำเภอพิมาย จังหวัดนครราชสีมา 30110
โทรศัพท์	044-471179

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนกุลโน
มัธยมศึกษาตอนต้น	โรงเรียนพิมายวิทยา
ประกาศนียบัตรวิชาชีพ (ปวช.)	-
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	สถาบันเทคโนโลยีราชมงคล วิทยาเขตภาคตะวันออกเฉียงเหนือ นครราชสีมา
ปริญญาตรี	สาขาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับรางวัล	-
ทุนการศึกษา	-
คติพจน์	ทำดีที่สุดแล้ว อย่าใช้ตนเองเป็นบรรทัดฐานในการตัดสินผู้อื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อผู้ทำปริญญานิพนธ์	นายสุภสรรณ์ เศิกศิริ
วันเดือนปีเกิด	12 เมษายน 2523
สถานที่เกิด	จังหวัดสงขลา
ภูมิลำเนาเดิม	จังหวัดยโสธร
ที่อยู่ปัจจุบัน	402 ถนนรอบเมือง ตำบลในเมือง อำเภอเมือง จังหวัดยโสธร 35000
โทรศัพท์	045-724368

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนเทศบาล 3
มัธยมศึกษาตอนต้น	โรงเรียนยโสธรพิทยาคม
ประกาศนียบัตรวิชาชีพ (ปวช.)	วิทยาลัยเทคนิคยโสธร
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	วิทยาลัยเทคนิคยโสธร
ปริญญาตรี	สาขาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับรางวัล	-
ทุนการศึกษา	-
คติพจน์	อย่าทำให้ผู้อื่นเดือดร้อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้