

ภาคผนวก

ภาคผนวก ก

แบบสอบถามความเหมาะสมของ

ชุดฝึกการทดลองการประยุกต์ใช้งานไมโครโปรเซสเซอร์ PIC16F877

แบบสอบถามความเหมาะสมของชุดฝึกการทดลองการประยุกต์ใช้งานไมโครโปรเซสเซอร์ 16F877

คำชี้แจง

แบบสอบถามฉบับนี้มีวัตถุประสงค์เพื่อสอบถามความเหมาะสมของชุดฝึกการทดลองการประยุกต์ใช้งานไมโครโปรเซสเซอร์ PIC16F877 แบ่งออกเป็น 3 ตอน

ตอนที่ 1 ข้อมูลเกี่ยวกับสถานภาพของผู้ตอบแบบประเมิน

ตอนที่ 2 แบบประเมินผลความคิดเห็นของผู้เชี่ยวชาญที่มีผลต่อชุดฝึกการทดลองการประยุกต์ใช้งานไมโครโปรเซสเซอร์ 16F877 ด้านเนื้อหาวิชา ด้านสื่อการเรียนการสอน และด้านประเมินผล

ตอนที่ 3 ข้อเสนอแนะเพิ่มเติม

โดยพิจารณาจากข้อมูลที่ได้วิจัยได้จัดทำขึ้นประกอบด้วย

- 1. แผนการสอนรายวิชาการประยุกต์ใช้งานไมโครโปรเซสเซอร์
- 2. ชุดฝึกการทดลองการประยุกต์ใช้งานไมโครโปรเซสเซอร์ 16F877
- 3. แบบทดสอบรายวิชาการประยุกต์ใช้งานไมโครโปรเซสเซอร์

ตอนที่ 1 ข้อมูลเกี่ยวกับสถานภาพของผู้ตอบแบบประเมิน

หน่วยงานที่สังกัด

ระดับการศึกษาสูงสุดของท่าน.....สาขาวิชา.....

มีประสบการณ์สอนที่เกี่ยวข้องกับไมโครโปรเซสเซอร์หรือไม่(เคย/ไม่เคย)

ประสบการณ์ในการทำงานสอน.....ปี

ตอนที่ 2 แบบประเมินผลความคิดเห็นของผู้เชี่ยวชาญที่มีผลต่อชุดฝึกการทดลองการประยุกต์ใช้งาน ไมโครโปรเซสเซอร์ 16F877 ด้านเนื้อหาวิชา ด้านสื่อการเรียนการสอน และด้านประเมินผล

รายการประเมิน	ระดับความเหมาะสม				
	มากที่สุด	มาก	ปานกลาง	น้อย	น้อยที่สุด
ด้านเนื้อหาวิชา					
1. เนื้อหาครอบคลุมวัตถุประสงค์ของวิชาเรียน					
2. การเรียงลำดับการนำเสนอสอดคล้องกับวัตถุประสงค์					
3. เนื้อหาเหมาะสมกับระดับของผู้เรียน					
4. ภาษาที่ใช้ในเนื้อหาเหมาะสมกับระดับผู้เรียน					
5. มีการสรุปเนื้อหาในแต่ละตอนอย่างเหมาะสม					
ด้านสื่อการเรียนการสอน					
1. การออกแบบการทดลอง					
2. ความสะดวกในการทดลอง					
3. เหมาะสมกับเนื้อหา ช่วยให้ผู้เรียนเกิดความสนใจ					
4. สามารถส่งเสริมให้ผู้เรียนเกิดการเรียนรู้ได้ดี					
ด้านการประเมินผล					
1. คำถามตรงตามวัตถุประสงค์เชิงพฤติกรรม					
2. จำนวนข้อสอบเหมาะสมกับวัตถุประสงค์เชิงพฤติกรรม					
3. คำถามและคำตอบมีเป้าหมายที่ชัดเจน					
4. คำถามมีความยากง่ายเหมาะสม					

ตอนที่ 3 ข้อเสนอแนะจากผู้เชี่ยวชาญ

.....

.....

.....

.....

ภาคผนวก ข

ใบงานการทดลองการประยุกต์ใช้งานไมโครโปรเซสเซอร์ PIC16F877

ชุดฝึกการทดลองการประยุกต์ใช้งานไมโครโปรเซสเซอร์ PIC16F877 ประกอบด้วย การทดลองจำนวน 5 การทดลอง 3 บอร์ดการทดลอง

การทดลองที่ 1 การใช้งานไมโครโปรเซสเซอร์เบื้องต้น มีจำนวน 1 การทดลอง

การทดลองที่ 2 การใช้งานอินพุต และ เอาต์พุตพอร์ต มีจำนวน 8 การทดลอง

การทดลองที่ 2.1 การเขียนโปรแกรมติดต่อกับ Input - Output ขั้นพื้นฐาน

การทดลองที่ 2.2 การเขียนโปรแกรมให้เอาพุททำงานแบบ Toggle

การทดลองที่ 2.3 การเขียนโปรแกรมเพื่อให้เอาพุททำงานแบบ Jogging

การทดลองที่ 2.4 การเขียนโปรแกรมเพื่อแก้ไขการดึงของหน้าสัมผัสสวิตช์

การทดลองที่ 2.5 การเขียนโปรแกรมเพื่อการทำงานซ้ำตามจำนวนครั้งที่กำหนด

การทดลองที่ 2.6 การเขียนโปรแกรมเพื่อควบคุมการเพิ่ม – ลดค่าตัวแปร

การทดลองที่ 2.7 การเขียนโปรแกรมเพื่อควบคุมการเลื่อนขยับบิตข้อมูลในตัวแปร

การทดลองที่ 2.8 การเขียนโปรแกรมเพื่อควบคุมการแสดงผลทาง LED 7 Segment

การทดลองที่ 3 การใช้งานการฟังก์ชัน Analog to Digital Converter มีจำนวน 4 การทดลอง

การทดลองที่ 3.1 การเขียนโปรแกรมแปลงสัญญาณอนาล็อกเป็นดิจิตอลแบบ 8 บิต

การทดลองที่ 3.2 การเขียนโปรแกรมแปลงสัญญาณอนาล็อกเป็นดิจิตอลแบบ 10 บิต

การทดลองที่ 3.3 การเขียนโปรแกรมแปลงสัญญาณอนาล็อกเป็นดิจิตอลแบบ 8 บิต แบบ

หลายช่อง

การทดลองที่ 3.4 การเขียนโปรแกรมแปลงสัญญาณดิจิตอล เป็นอนาล็อก (D/A)

การทดลองที่ 4 การใช้งานร่วมกับจอแสดงผล ชนิด LCD Module และ Digital to Analog Converter มีจำนวน 2 การทดลอง

การทดลองที่ 4.1 การเขียนโปรแกรมเพื่อแสดงผลผ่านจอแอลซีดี (LCD Display)

การทดลองที่ 4.2 การเขียนโปรแกรมประยุกต์การแสดงผลผ่านจอแอลซีดี (LCD Display)

การทดลองที่ 5 การใช้งานการควบคุมมอเตอร์ไฟฟ้ากระแสตรง มีจำนวน 5 การทดลอง

การทดลองที่ 5.1 การเขียนโปรแกรมกำเนิดสัญญาณ PWM

การทดลองที่ 5.2 การเขียนโปรแกรมควบคุมมอเตอร์ดีซีแบบหมุนต่อเนื่อง

การทดลองที่ 5.3 การเขียนโปรแกรมควบคุมมอเตอร์ดีซีแบบหมุนตามจังหวะที่เกิดสวิตช์

การทดลองที่ 5.4 การเขียนโปรแกรมควบคุมความเร็วของมอเตอร์ดีซี

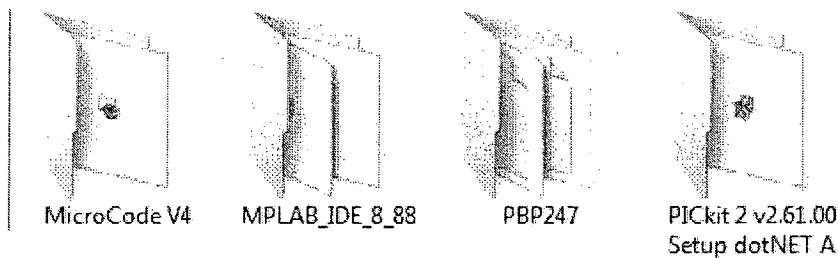
การทดลองที่ 5.5 การเขียนโปรแกรมควบคุมทั้งความเร็วและทิศทางของมอเตอร์ดีซี

การทดลองที่ 1 การใช้งานไมโครโปรเซสเซอร์เบื้องต้น

จุดประสงค์ เพื่อศึกษาการเขียนโปรแกรมให้ไมโครโปรเซสเซอร์และการเชื่อมต่อสำหรับการโปรแกรมรหัสคำสั่งการทำงานไมโครโปรเซสเซอร์

การติดตั้งโปรแกรมที่เกี่ยวข้องกับการใช้งานไมโครโปรเซสเซอร์ PIC 16F877 ประกอบด้วย

1. โปรแกรมสำหรับ การสร้างโปรแกรมภาษาเบสิก มีชื่อว่า Microcode Studio
2. โปรแกรมสำหรับการแปลภาษาเบสิกเป็นภาษาแอสเซมบลี มีชื่อว่าโปรแกรม MPLAB IDE
3. โปรแกรมสำหรับแปลภาษาแอสเซมบลีเป็นรหัสให้ไมโครโปรเซสเซอร์ทำงาน (Operation Code) มีชื่อว่า โปรแกรม PBP
4. โปรแกรมสำหรับการโหลดรหัสการทำงานให้ไมโครโปรเซสเซอร์มีชื่อว่า Pickit 2

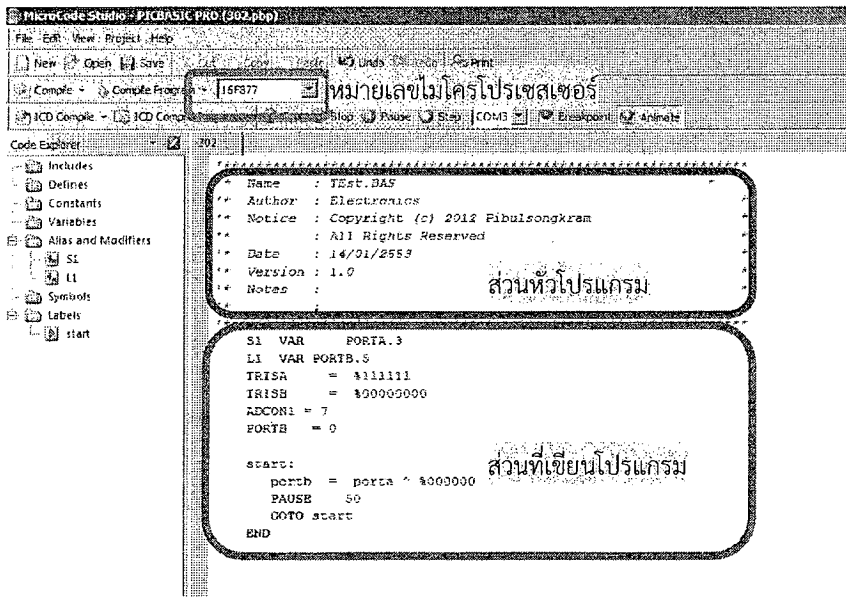


รูปแสดงไฟล์เดอร์โปรแกรมที่จำเป็นสำหรับพื้นฐาน
การเขียนโปรแกรมภาษาเบสิกควบคุมการทำงานไมโครโปรเซสเซอร์ 16F877

โปรแกรม Microcode Studio เป็นโปรแกรมหลักที่ใช้สำหรับในการสร้างโปรแกรมสามารถใช้งานเขียนโปรแกรม แก้ไขโปรแกรม คอมไพล์เป็นรหัสการทำงานไมโครโปรเซสเซอร์ตระกูล PIC ของบริษัทไมโครชิพ จำกัด

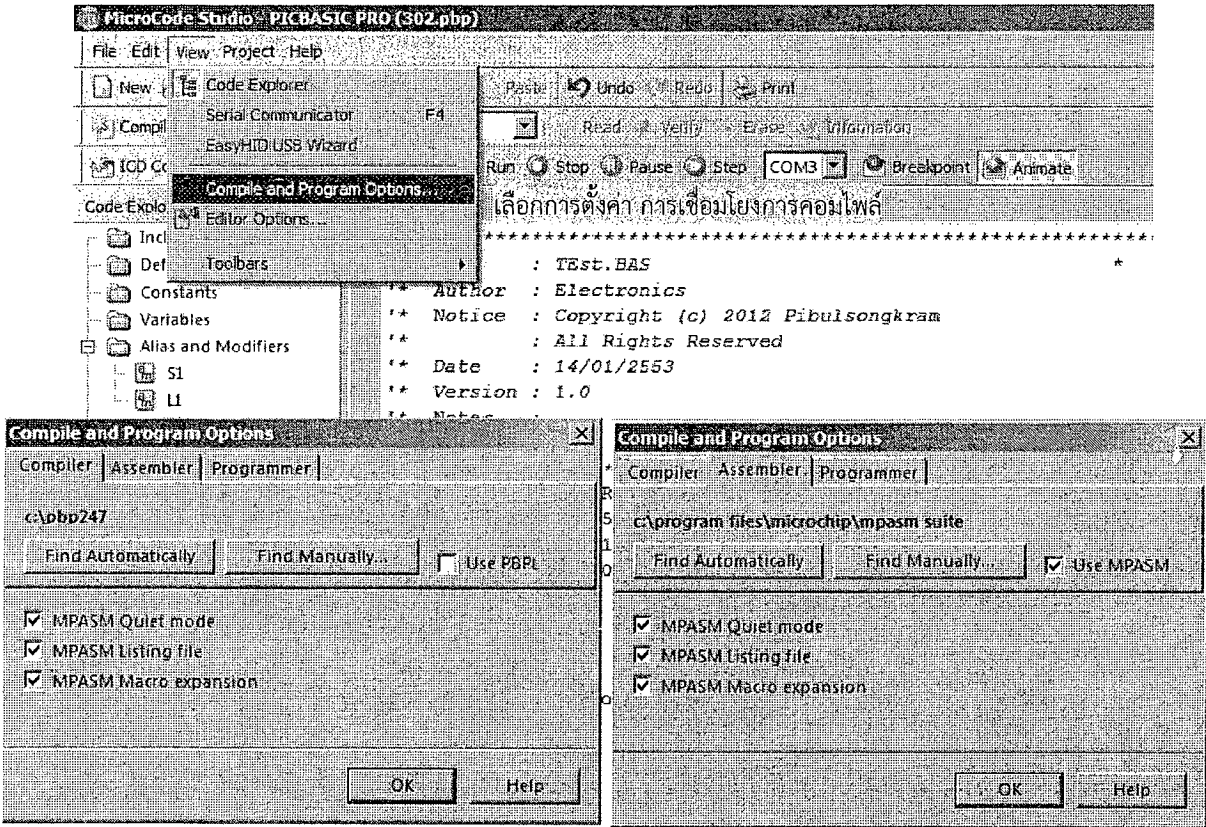
ผู้ใช้งานจะต้องติดตั้งโปรแกรม MPLAB เพื่อใช้งาน โปรแกรม MPASM.EXE สำหรับการเปลี่ยนไฟล์ชนิด ASM เป็นไฟล์ชนิด HEX ซึ่งเป็นรหัสคำสั่ง (Operation Code : OP Code) และนำข้อมูลการคอมไพล์ไฟล์เดอร์ PBP247 ไปวางไว้ใน Drive C เพื่อสะดวกในการใช้งาน

ติดตั้งโปรแกรมสำหรับการดาวน์โหลดไมโครโปรเซสเซอร์ PIC16F877 โดยติดตั้งโปรแกรม PICKIT2



เมื่อติดตั้งโปรแกรม Microcode Studio เสร็จแล้ว ลำดับต่อไปกำหนดค่าการเชื่อมโยงโปรแกรมที่เกี่ยวข้อง 2 ส่วน คือ

- 1. ส่วนคอมไพเลอร์ โดยกำหนดให้เชื่อมโยงไปยังไฟล์ c:\PBP247
- 2. ส่วนแอสเซมเบอ โดยกำหนดให้เชื่อมโยงไปไฟล์ MPASM.EXE ที่อยู่ในไฟล์ c:\Programfiles\microchip\mpasm suit



ทดสอบเขียนโปรแกรม

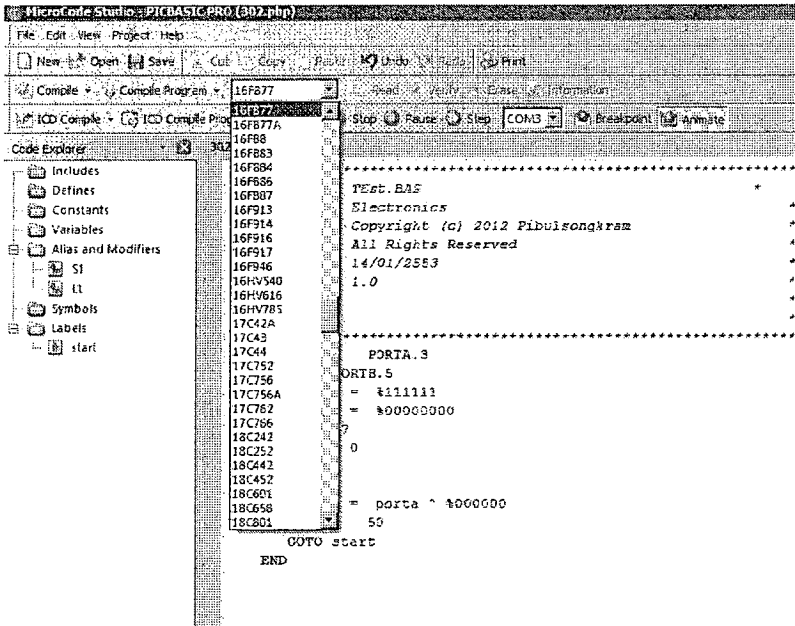
```
OSC DEFINE 10
S1    VAR    PORTA.3
L1    VAR    PORTB.5
TRISA  = %111111
TRISB  = %00000000
ADCON1 = 7
PORTB  = 0

start:

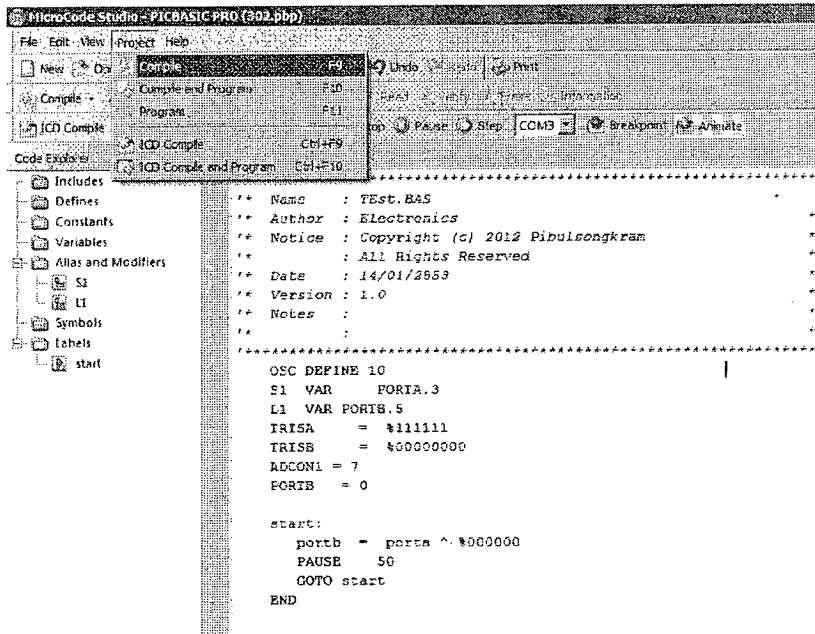
    portb = porta ^ %0000000
    pause 50
    GoTo  start
End
```

เมื่อพิมพ์คำสั่งแล้วบันทึกไฟล์ จะได้ไฟล์ข้อมูลชนิด Basic นามสกุล pbp

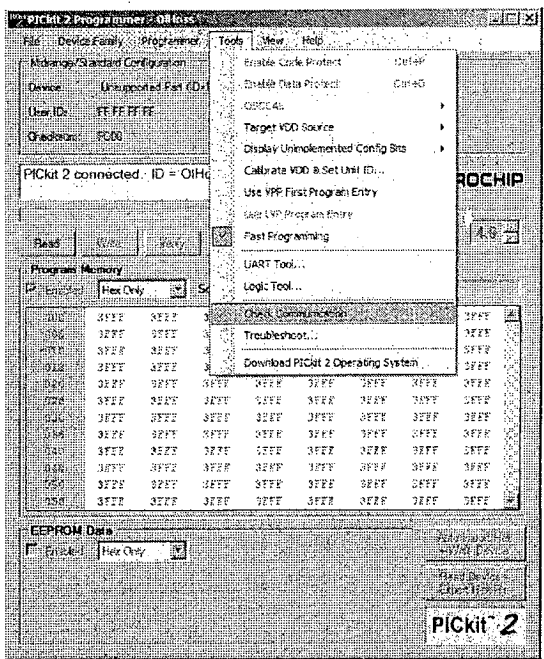
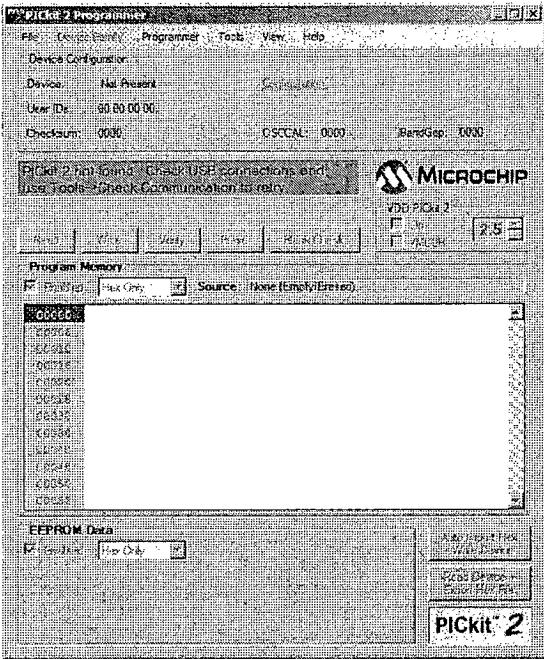
ขั้นตอนการเลือกหมายเลขไมโครโปรเซสเซอร์ที่ใช้ 16F877

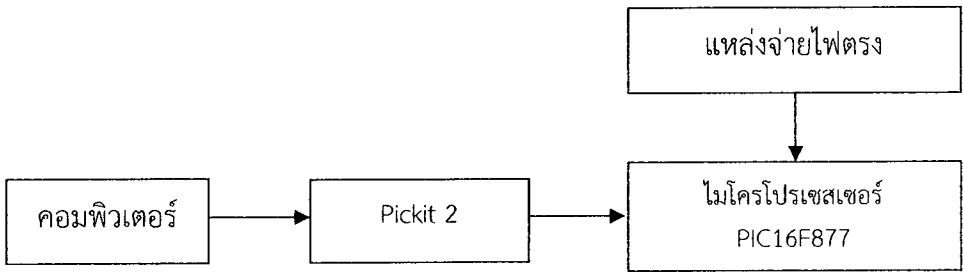


ขั้นตอนการคอมไพล์ไฟล์ pbp ให้เป็นไฟล์ HEX ด้วยการกด F9 หรือเลือกที่เมนู Project เลือกเมนูย่อย Compile ดังรูป

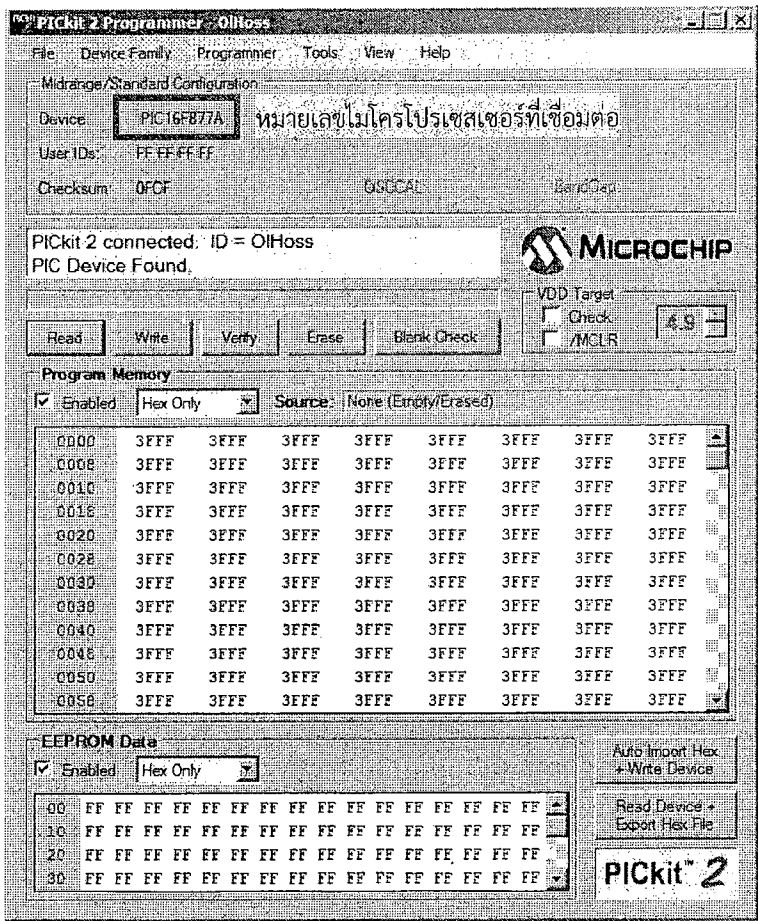


เมื่อติดตั้ง โปรแกรม Pickit 2 จะพบว่าโปรแกรมจะตรวจสอบการเชื่อมต่อกับ ไมโครโปรเซสเซอร์ผ่านพอร์ต USB ให้นักศึกษาเชื่อมต่ออุปกรณ์สำหรับการดาวน์โหลด Pickit 2 และตรวจสอบการเชื่อมต่ออีกครั้งที่เมนู Tools และเลือกเมนูย่อย check Communication ถ้าพบว่าการเชื่อมต่อไมโครโปรเซสเซอร์เรียบร้อยแล้ว จะแสดงหมายเลขไมโครโปรเซสเซอร์ที่ช่อง Device





รูปแสดงการเชื่อมต่อการโปรแกรมไมโครโปรเซสเซอร์ PIC16F877



รูปแสดงผลการเชื่อมต่อไมโครโปรเซสเซอร์เข้ากับคอมพิวเตอร์ผ่าน อุปกรณ์ดาวน์โหลด Pickit 2

เมื่อเชื่อมต่อไมโครโปรเซสเซอร์สำเร็จ นักศึกษา Import ไฟล์ Hex ที่เมนู File เลือกเมนูย่อย Import และหาตำแหน่งไฟล์เดอร์ไฟล์ Hex ที่ต้องการโหลดให้กับไมโครโปรเซสเซอร์ PIC16F877

บันทึกผลการทดลอง

.....

.....

.....

.....

.....

แบบฝึกหัด

- 1. จงเขียนโปรแกรมลำดับขั้นตอนการเขียนโปรแกรมด้วย BASIC PIC PRO
- 2. การกำหนดการเชื่อมโยงโปรแกรมที่จำเป็น Microcode Studio สามารถกำหนดได้โดยเชื่อมโยงไปยังที่อยู่ของไฟล์ มีการกำหนดค่าอะไรบ้าง
- 3. การเชื่อมต่อผ่านพอร์ต USB เข้าสู่อุปกรณ์ช่วยดาวน์โหลดไมโครโปรเซสเซอร์ 16F877 มีการออกแบบใช้งานพอร์ต อย่างไร

การทดลองที่ 2.1 การเขียนโปรแกรมติดต่อกับ Input - Output ขั้นพื้นฐาน

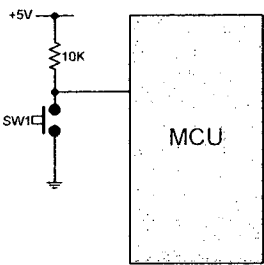
จุดประสงค์ เพื่อศึกษาการเขียนโปรแกรมให้ไมโครโปรเซสเซอร์ รับการกดสวิตช์ และส่งเอาท์พุทออกทางขา I/O เพื่อเป็นพื้นฐานการประยุกต์ใช้งานต่อไป

ทฤษฎีพื้นฐาน

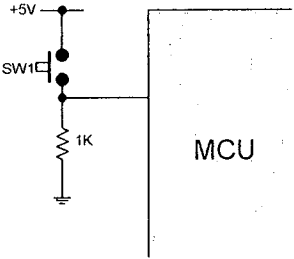
การทำงานของระบบไมโครโปรเซสเซอร์ โดยทั่วไปประกอบด้วย 3 ส่วน คือ ส่วนอินพุต ส่วนประมวลผลหรือส่วนควบคุม และส่วนเอาท์พุท

ส่วนอินพุตประกอบด้วยอุปกรณ์ที่เป็นวงจรทำหน้าที่ป้อนสัญญาณให้แก่ส่วนประมวลผล สัญญาณที่ป้อนมี 2 แบบคือ

- แบบดิจิทัล (Digital Signal) มีระดับแรงเคลื่อนแค่สองสถานะคือระดับที่เป็นลอจิก 1 คือ 5 โวลท์ และระดับลอจิก 0 คือ น้อยกว่า 1.5 โวลท์ หรือ 0 โวลท์ ตัวอย่างวงจร ได้แก่ สวิตช์ปิด – เปิดแบบต่าง ๆ



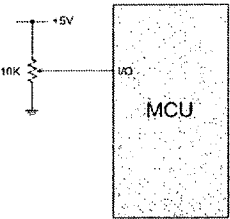
รูปที่ 1 การต่อสวิตช์กับ MCU แบบลอจิกกลับ



รูปที่ 2 การต่อสวิตช์กับ MCU แบบลอจิกบวก

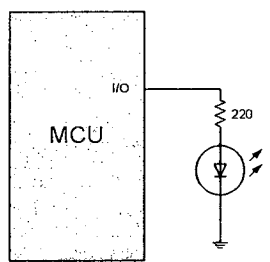
ในทางการนำไปใช้งานจริงนิยมต่อวงจรตามรูปที่ 1 เพื่อป้องกันสัญญาณรบกวนขณะที่ยังไม่กดสวิตช์ เนื่องจากตามรูปที่ 2 ขณะที่ไม่มีการกดสวิตช์ ขา I/O ของ MCU จะอยู่ในสภาวะลอยอยู่ เนื่องจากไม่ได้ต่อโดยตรงกับ Ground

- แบบอนาล็อก (Analog Signal) มีระดับแรงเคลื่อนตั้งแต่ 0 – 5 โวลท์โดยสามารถรับได้ถึง 2 ระดับ หรือ 1064 ระดับ ตัวอย่างอุปกรณ์ได้แก่ตัวตรวจจับและส่งสัญญาณระดับอุณหภูมิ ความดัน ระดับของเหลว หรือ ตำแหน่งหมุนของ Potentiometer เป็นต้น ระดับทั้งหมดที่กล่าวนี้จะต้องถูกปรับเปลี่ยนให้เป็นระดับของแรงเคลื่อน 0-5 โวลท์ก่อนที่จะส่งเข้าขา I/O ของไมโครโปรเซสเซอร์



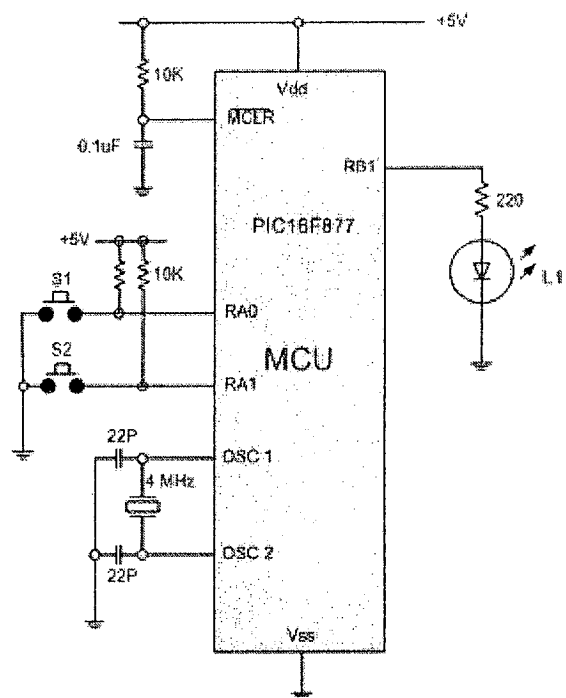
รูปที่ 3 การต่อวงจรรับสัญญาณแอนาล็อกจากตัวต้านทานปรับค่าได้

การต่อสัญญาณอนาล็อกเข้ากับขา I/O ต้องต่อให้ตรงกับพอร์ทที่สามารถรับสัญญาณอนาล็อกได้ซึ่ง MCU แต่ละยี่ห้อ และแต่ละรุ่นจะไม่เหมือนกัน ให้อ่านใน Datasheet (รายละเอียดให้ศึกษาในใบงานที่เกี่ยวข้องต่อไป) ส่วนเอาพุทเป็นส่วนที่รับสัญญาณจาก MCU มาทำงาน โดยทั่วไปจะเป็นสัญญาณแบบดิจิทัล สิ่งที่ต้องคำนึงถึงในการออกแบบคือ กระแสที่ MCU ส่งออกมาขับอุปกรณ์เอาพุทของแต่ละขา I/O แต่ละยี่ห้อ และแต่ละรุ่นจะไม่เหมือนกัน สำหรับ MCU ของไมโครชิพตระกูล PIC I/O แต่ละขาสามารถขับกระแสได้เต็มที่ 25 mA ในการออกแบบจะต้องเผื่อความปลอดภัยไว้จะต้องไม่เกิน 10 mA



รูปที่ 4 การต่อวงจรแสดงผลด้วยหลอด LED

วงจรทดลองตาม การทดลองที่ 2.1



รูปที่ 5 วงจรทดลอง

การทำงานของโปรแกรม

เมื่อกดสวิตช์ S1 ทำให้หลอด L1 ติด กด S2 ทำให้หลอด L1ดับ

โปรแกรมคำสั่ง

```
S1    VAR    PORTA.0
S2    VAR    PORTA.1
L1    VAR    PORTB.1
      TRISA   =  %111111
      TRISB   =  %00000000
      ADCON1  =  7
      PORTB   =  0
LOOP: IF (S1 = 0) AND (S2 = 1) THEN HIGH L1
      IF (S1 = 1) AND (S2 = 0) THEN LOW L1
      PAUSE 50
      GOTO   LOOP
      END
```

บันทึกผลการทดลอง

.....

.....

.....

.....

.....

แบบฝึกหัด

- 1. ถ้าต้องการออกแบบให้มีการเชื่อมต่อ LED ที่ PORT D.1 โดยมีสวิตช์ต่อแบบ HIGH Pull-UP จะต้องแก้ไขส่วนใดของโปรแกรมบ้าง
- 2. คำสั่ง TRISA = %111111 และ TRISB = %00000000 มีความหมายว่าอย่างไร
- 3. นักศึกษาคิดว่าสามารถออกแบบให้ PORTD.0 และ PORTD.1 เชื่อมต่อกับสวิตช์ S1 และ S2 และให้ PORTD.2 เชื่อมต่อกับ L1 ดังรูปการทดลอง ได้หรือไม่ เพราะเหตุใด

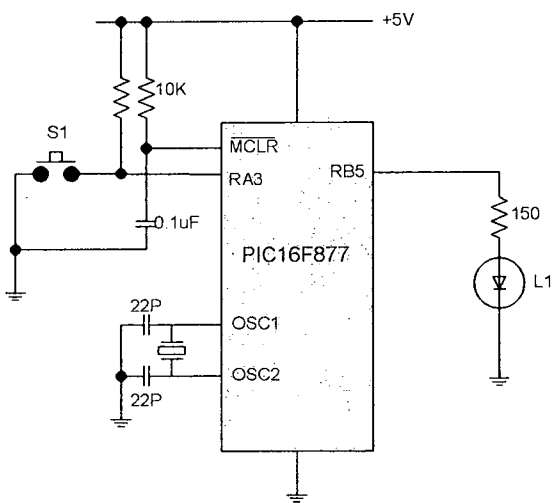
การทดลองที่ 2.2 การเขียนโปรแกรมให้เอาพุทำงานแบบ Toggle

จุดประสงค์ เพื่อศึกษาการเขียนโปรแกรมเปลี่ยน Tac Switch ให้ทำงานเป็น Toggle Switch เพื่อนำไปประยุกต์ใช้ทำสวิตช์ควบคุมการเปิด - ปิด อุปกรณ์ไฟฟ้า เป็นต้น

ทฤษฎีพื้นฐาน

เนื่องจากอุปกรณ์สวิตช์ที่ใช้เชื่อมต่อกับไมโครโปรเซสเซอร์ ส่วนมากจะออกแบบให้ทำงานในลักษณะเป็นสวิตช์ปุ่มกดแบบกดติด-ปล่อยดับ ดังนั้นหากเราต้องการให้ได้ผลเอาพุเป็นแบบ Toggle หรือกดครั้งแรกติดและกดครั้งที่สอง ดับ สลับกันไป จะต้องเขียนโปรแกรมบังคับตัวเอง

วงจรทดลองตาม การทดลอง 2



การทำงานของโปรแกรม

เมื่อกด S1 ทำให้หลอด L1 ติดเมื่อกด S1 อีกครั้ง หลอด L1 ดับ และเมื่อกด S1 อีกครั้งหลอด L1 จะติดเป็นแบบนี้สลับกัน เรียกว่า “Toggle” โดยใช้คำสั่ง TOGGLE

โปรแกรมคำสั่ง

```
S1    VAR    PORTA.3
L1    VAR    PORTB.5
TRISA = %1111111
TRISB = %00000000
ADCON1 = 7
PORTB = 0

start:
    IF S1 = 1 Then goto start
    TOGGLE L1
    Pause 100

loop:
    IF S1 = 0 Then goto loop
    Pause 100
    GoTo start
End
```

บันทึกผลการทดลอง

.....

.....

.....

.....

.....

แบบฝึกหัด

1. ถ้าเปลี่ยนค่า Pause 100 เป็น 1000 จะมีผลการทดลองเช่นไร
2. ถ้าต้องการให้มีผลการกดสวิตช์ S1 และ S2 ตรงกันข้ามจะเขียนโปรแกรมอย่างไร

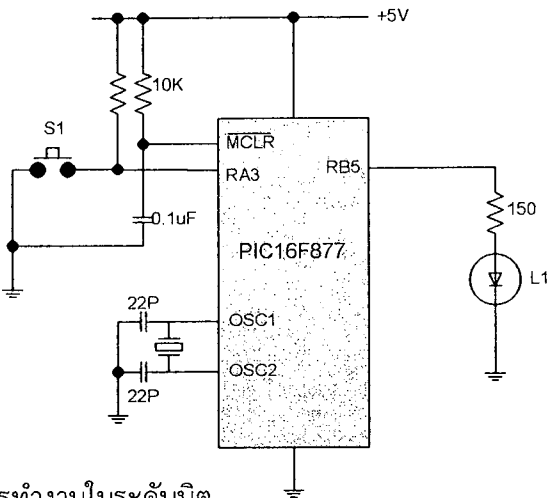
การทดลองที่ 2.3 การเขียนโปรแกรมเพื่อให้เอาพุทำงานแบบ Jogging

จุดประสงค์ เพื่อศึกษาการเขียนโปรแกรมให้ไมโครโปรเซสเซอร์ ส่งค่าเอาพุเมื่อกดสวิตช์ และหยุดส่งเมื่อไม่กดปุ่มสวิตช์ เพื่อเป็นพื้นฐานในการประยุกต์ใช้งานต่อไป

ทฤษฎีพื้นฐาน

การทำงานแบบ Jogging คือการเขียนโปรแกรมให้ไมโครโปรเซสเซอร์ ส่งค่าเอาพุออกมาตลอดเวลาขณะที่ยังกดสวิตช์อินพุตค้างอยู่ และหยุดส่งเมื่อเราปล่อยสวิตช์ ตัวอย่าง งานที่ใช้การควบคุมแบบ Jogging ได้แก่ การควบคุม แขน บั่นจั่น และการบังคับเป็นจังหวะ เช่น การบังคับการเคลื่อนที่ของหุ่นยนต์แบบ Manual เป็นต้น หลักการเขียนโปรแกรมจะต้องใช้คำสั่งวน คอยตรวจสอบสถานการณ์กดสวิตช์ โดยมีเงื่อนไขว่าหากสวิตช์ยังถูกกดอยู่ต้องส่งค่าเอาพุออกไป หากปล่อยสวิตช์จะหยุดส่งเอาพุออก

วงจรทดลองตาม การทดลองที่ 2.3



โปรแกรมคำสั่งที่ 1 การทำงานในระดับบิต

```
S1    VAR    PORTA.3
L1    VAR    PORTB.5
      TRISA  = %111111
      TRISB  = %00000000
      ADCON1 = 7
      PORTB  = 0

start:
      if S1 = 0 Then
          HIGH  L1
      else
          LOW   L1
      endif
      pause 50
      GoTo  start
      End
```

โปรแกรมคำสั่งที่ 2 การทำงานในระดับพอร์ท

สำหรับใช้กับวงจรที่มีสวิตช์อินพุตต่ออยู่หลายตัว โดยไม่ต้องเขียนโปรแกรมให้ใช้คำสั่ง IF.. THEN คอยตรวจสอบเงื่อนไขการกดสวิตช์ทุกตัว ซึ่งจะทำให้โปรแกรมทำงานช้าลงโดยไม่จำเป็น และจะทำให้โปรแกรมสั้นลงมาก

```
S1    VAR    PORTA.3
L1    VAR    PORTB.5
      TRISA   = %111111
      TRISB   = %00000000
      ADCON1 = 7
      PORTB   = 0

start:
      portb = porta ^ %0000000
      pause 50
      GoTo  start
      End
```

การทำงานของโปรแกรม

Portb = porta ^ %0000000 หมายความว่า ให้เอาค่าลอจิกที่พอร์ท A ทำ Exclusive – OR กับเลขไบนารี %0000000 กำหนดเป็นค่าพอร์ท B ซึ่งเป็นเอาพุท

บันทึกผลการทดลอง

.....

.....

.....

.....

.....

แบบฝึกหัด

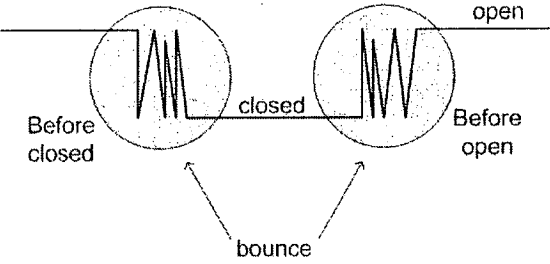
- 1. คำสั่ง pause 50 มีความหมายว่าอย่างไร และสามารถช่วยการทำงานของ Jogging ได้อย่างไร
- 2. ถ้าต้องการให้มีผลการกดสวิตช์ S1 และ S2 ตรงกันข้ามจะเขียนโปรแกรมอย่างไร

การทดลองที่ 2.4 การเขียนโปรแกรมเพื่อแก้ไขการค้างของหน้าสัมผัสสวิตช์

จุดประสงค์ เพื่อศึกษาการเขียนโปรแกรมในการแก้ไขการค้างของหน้าสัมผัสสวิตช์ (De - bounce) หรือเพื่อให้ไมโครโปรเซสเซอร์ รับรู้การกด และการถอนนิ้วจากปุ่มสวิตช์

ทฤษฎีพื้นฐาน

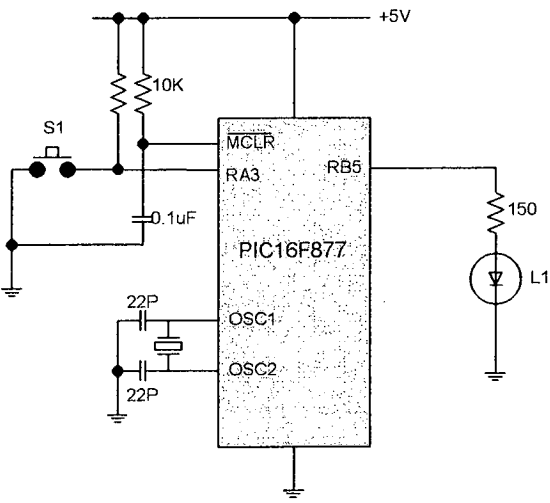
การใช้สวิตช์เป็นอุปกรณ์ป้อนสัญญาณลอจิกให้กับไมโครโปรเซสเซอร์ จะต้องระวังเรื่องการค้างของหน้าสัมผัส ซึ่งจะเกิดขึ้นระหว่างหน้าสัมผัสกำลังจะติด และหน้าสัมผัสกำลังจะจากกันโดยจะเกิดรูปพัลส์หัวแตกขึ้นหลายลูกในช่วงเวลา 1 - 10 mS รูปพัลส์หัวแตกดังกล่าวจะทำให้เกิดการ ทำงานของโปรแกรมผิดพลาดขึ้นได้



รูปที่ 1 แสดงการเกิดการค้างของหน้าสัมผัสสวิตช์

การเขียนโปรแกรมเพื่อแก้ไขการค้าง จะต้องใช้คำสั่งหน่วงเวลาช่วงสั้นๆ ก่อนที่จะไปตรวจสอบค่าลอจิกการกดสวิตช์ โดยทั่วไปจะอยู่ระหว่าง 10 - 50 mS เพื่อให้การรับรู้สัญญาณลอจิกเลยตำแหน่งการค้างของหน้าสัมผัสสวิตช์ไปก่อน

วงจรทดลองตาม การทดลองที่ 2.4



การทำงานของโปรแกรม

จากวงจร ขณะยังไม่กดสวิตช์ S1 หลอด L1 ดับ เมื่อกดสวิตช์ S1 ค้าง หลอดยังคงดับ เมื่อปล่อย S1 หลอดจึงจะติด และเมื่อกด S1 ค้าง หลอด L1 ยังคงติด จนกว่าจะปล่อย S1 หลอดจึงดับ ซึ่งเป็นการทำงานแบบ Toggle แต่เอาพุทจะทำงานก็ต่อเมื่อเราได้ปล่อย S1 ซึ่งเป็นวิธีการที่ถูกต้องในการนำไปประยุกต์ใช้งาน

โปรแกรมคำสั่ง

```
S1      VAR    PORTA.3
L1      VAR    PORTB.5
TRISA   = %11111111
TRISB   = %00000000
ADCON1  = 7
PORTB   = 0
start:
  if S1 = 1 then goto start
  pause 50
loop:
  if S1 = 0 then goto loop
  toggle L1
  pause 50
  goto start
end
```

บันทึกผลการทดลอง

.....

.....

.....

.....

.....

แบบฝึกหัด

- 1. ทดสอบแก้ไขค่า 50 คำสั่ง pause 50 หาค่าที่เหมาะสมในการกดสวิตช์แล้วมีความแม่นยำสูงที่สุด

การทดลองที่ 2.5 การเขียนโปรแกรมเพื่อการทำงานซ้ำตามจำนวนครั้งที่กำหนด

จุดประสงค์ เพื่อศึกษาการเขียนโปรแกรมในการ ประมวลผลการทำงานซ้ำตามจำนวนครั้งที่กำหนด

ทฤษฎีพื้นฐาน

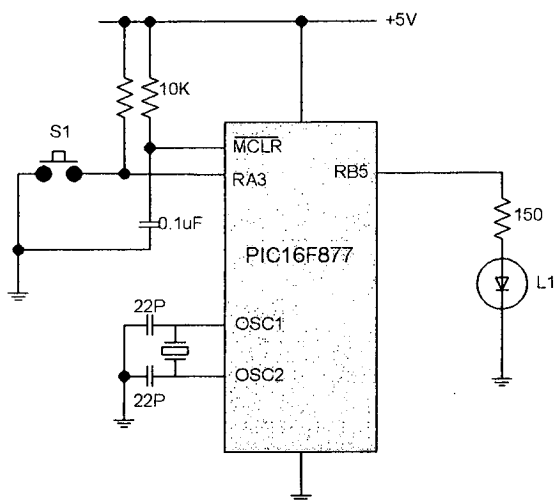
ในการเขียนโปรแกรมเพื่อกำหนดให้ประมวลผลการทำงานซ้ำ จะมีอยู่ 3 ลักษณะคือ

- ใช้คำสั่งกระโดดย้อนกลับไปเริ่มต้นทำงานใหม่โดยไม่มีเงื่อนไขกำหนดด้วยคำสั่ง GOTO คำสั่งนี้ จะใช้กรณีย้อนกลับไปเริ่มต้นทำงานที่จุดเริ่มต้นใหม่ ไม่รู้จบ

- ใช้คำสั่งในการประมวลผลการทำงานตามจำนวนครั้งที่กำหนด ด้วยคำสั่ง FOR ... NEXT

- ใช้คำสั่งในการประมวลผลการทำงานซ้ำจนกว่าเงื่อนไขที่กำหนดจะเป็นจริง หรือ เท็จ ด้วยคำสั่ง WHILE ... WEND

วงจรทดลองตาม การทดลองที่ 2.5



โปรแกรมคำสั่ง

```

S1    VAR    PORTA.3
L1    VAR    PORTB.5
i     VAR    byte
      TRISA  = %11111111
      TRISB  = %00000000
      ADCON1 = 7
      PORTB  = 0
start: if s1 = 1 then goto start
      pause 50
loop:  if s1 = 0 then goto loop
      for i = 1 to 5
        high I1
        pause 500
        low I1
        pause 500
      next i
      goto start
end
  
```

บันทึกผลการทดลอง

.....

.....

.....

.....

.....

แบบฝึกหัด

- 1. จงบอกถึงความแตกต่างคำสั่ง if กับ คำสั่ง for
- 2. ถ้าเปลี่ยน TRISA = %11111111 เป็น TRISA = %111111 ทำไมผลการดำเนินงานจึงไม่เปลี่ยนแปลง จงอธิบาย

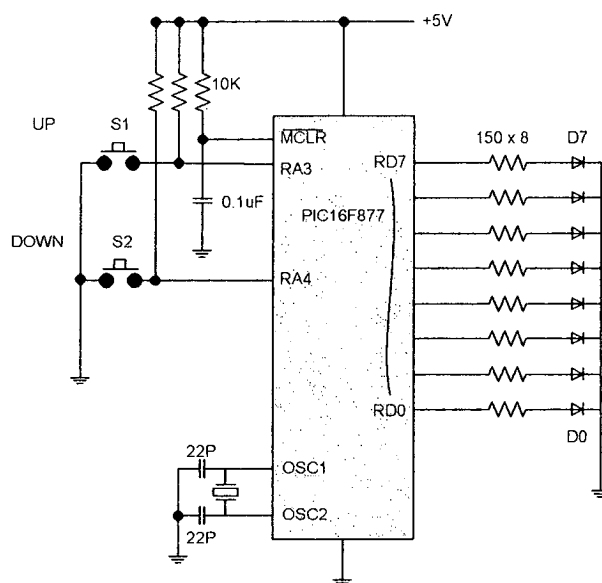
การทดลองที่ 2.6 การเขียนโปรแกรมเพื่อควบคุมการเพิ่ม – ลดค่าตัวแปร

จุดประสงค์ เพื่อศึกษาการเขียนโปรแกรมในการควบคุมการเพิ่ม-ลดค่าตัวแปรด้วยวิธีการต่าง ๆ

ทฤษฎีพื้นฐาน

การควบคุมการเพิ่ม – ลดค่าตัวแปรจากอุปกรณ์สวิตช์ที่ขา I/O เป็นพื้นฐานสำคัญในการนำไปประยุกต์เขียนโปรแกรมเพื่อการประมวลผลการควบคุมค่าที่มีระดับมาก-น้อย ตัวอย่างเช่น การเพิ่ม-ลดความเร็วมอเตอร์ การเพิ่ม-ลดอุณหภูมิ เป็นต้น ในการออกแบบอุปกรณ์ทางด้านอินพุท อย่างน้อยต้องประกอบด้วยสวิตช์ 2 ตัว สำหรับการกดเพิ่มค่า และลดค่าการเขียนโปรแกรมในการประมวลผลจะมี 2 ลักษณะคือ การควบคุมให้เพิ่ม-ลดค่าต่อเนื่อง เมื่อกดสวิตช์ค้างไว้ หรือเรียกว่า continuous operation และการควบคุมให้ทำงานค้างที่ละสแต็ปหรือเรียกว่า step operation

วงจรทดลองตาม การทดลองที่ 2.6



โปรแกรมคำสั่งที่ 1 Continuous operation

```

S1 VAR    PORTA.3
S2 VAR    PORTA.4
k  var byte
TRISA  =  %111111
TRISD  =  %00000000
ADCON1 = 7
portd = 0
LOOP: IF (S1 = 0) AND (S2 = 1) THEN
    pause 50
    k = k+1
    portd = k
    if k = 255 then k = 254
endif IF (S1 = 1) AND (S2 = 0) THEN
    pause 50
    if k = 0 then k=1
    k = k - 1
    portd = k
endif
PAUSE 150
GOTO  LOOP
END

```

บันทึกผลการทดลอง

โปรแกรมคำสั่งที่ 2 Step operation

```

S1  VAR    PORTA.3
S2  VAR    PORTA.4
k   var    byte
TRISA  =  %111111
TRISD  =  %00000000
ADCON1 = 7
portd = 0
LOOP: IF (S1 = 0) AND (S2 = 1) THEN
    pause 50
    k = k+1
    portd = k
    if k = 255 then k = 254
idle1: if (S1 = 0) AND (S2 = 1) THEN idle1
endif
    IF (S1 = 1) AND (S2 = 0) THEN
        pause 50
        if k = 0 then k=1
        k = k - 1
        portd = k
idle2: if (S1 = 1) AND (S2 = 0) THEN idle2
endif
    PAUSE 150
    GOTO  LOOP
END

```

บันทึกผลการทดลอง

แบบฝึกหัด

1.

จงหาข้อแตกต่างของโปรแกรมที่ 1 และ 2
2.

ถ้าต้องการให้โปรแกรมที่ 1 มีผลการกดสวิตช์มีการเพิ่มค่าครั้งละ 5 ค่าจะต้องแก้ไขส่วนใดของโปรแกรม

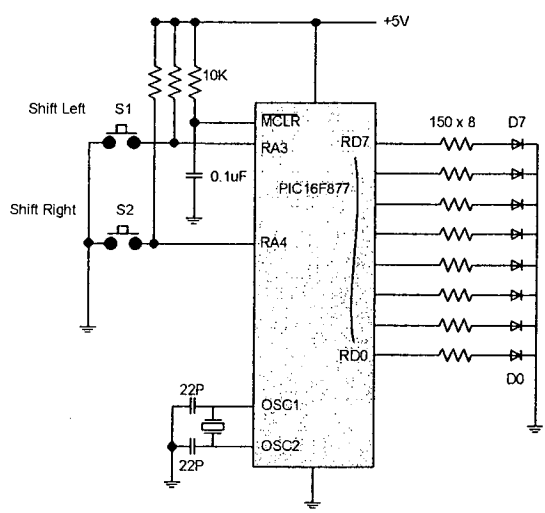
การทดลองที่ 2.7 การเขียนโปรแกรมเพื่อควบคุมการเลื่อนขยับบิตข้อมูลในตัวแปร

จุดประสงค์ เพื่อศึกษาการเขียนโปรแกรมในการควบคุมการเลื่อนขยับบิตของข้อมูลในตัวแปร ด้วยวิธีการต่าง ๆ

ทฤษฎีพื้นฐาน

การเลื่อน และขยับบิตข้อมูลในตัวแปรเป็นพื้นฐานสำคัญทั้งในการประมวลผลทางคณิตศาสตร์ และการประยุกต์ใช้งาน เช่น การควบคุมจังหวะการทำงานของเครื่องจักร การทำสัญญาณควบคุมไฟวิ่ง เป็นต้น เครื่องหมายที่ใช้ในการประมวลผลที่สำคัญในการเลื่อนบิตไปทางซ้าย - ขวา คือ << และ >> ข้อสำคัญในการใช้คำสั่ง คือ เมื่อการเลื่อนข้อมูลไปถึงบิตสุดท้ายทั้ง MSB หรือ LSB ก็ตาม เมื่อเลยไปจากนี้ บิตข้อมูลจะตกขอบ ไม่มีการย้อนกลับไปเริ่มต้นใหม่ ซึ่งแตกต่างไปจากการหมุน (Rotate) บิตข้อมูล และในภาษา Pic Basic จะไม่มีคำสั่งสำหรับการหมุนข้อมูลรองรับ ดังนั้นหากต้องการ จะต้องใช้ภาษาแอสเซมบลี แทนเอง หรือต้องเขียนโปรแกรมให้โหลดข้อมูลย้อนกลับมาเริ่มต้นใหม่

วงจรทดลองตาม การทดลองที่ 2.7



โปรแกรมคำสั่งที่ 1 การควบคุมการเลื่อนบิตไปทางซ้าย - ขวา

```
S1    VAR    PORTA.3
S2    VAR    PORTA.4
k     VAR    byte
TRISA = %111111
TRISD = %00000000
ADCON1 = 7
portd = %10000000
LOOP: PAUSE 400
      IF (S1 = 0) THEN
        portd = portd >> 1
        if portd = %00000000 then portd = %10000000
      else
        portd = portd << 1
        if portd = %00000000 then portd = %00000001
      endif
      GOTO LOOP
END
```

บันทึกผลการทดลอง

โปรแกรมคำสั่งที่ 2 การควบคุมการขยับบิดไปทางซ้าย – ขวา ทีละสเต็ป

```

S1      VAR    PORTA.3
S2      VAR    PORTA.4
k       VAR    byte
        TRISA  =  %1111111
        TRISD  =  %00000000
        ADCON1 = 7
        portd  = %10000000
LOOP:
        IF (S1 = 0) and (S2 = 1) THEN
            pause 50
            portd = portd >> 1
            if portd = %00000000 then portd = %10000000
            idle1: IF (S1 = 0) and (S2 = 1) THEN idle1
        endif
        if (S1 = 1) and (S2 = 0) THEN
            pause 50
            portd = portd << 1
            if portd = %00000000 then portd = %00000001
            idle2: IF (S1 = 1) and (S2 = 0) THEN idle2
        endif
        GOTO LOOP
    END

```

บันทึกผลการทดลอง

โปรแกรมคำสั่งที่ 3 การควบคุมการขยับบิตไปทางซ้าย – ขวา แบบค้างสถานะต่อเนื่อง

```

S1      VAR    PORTA.3
S2      VAR    PORTA.4
k       VAR    byte
direction VAR    bit
        TRISA  =  %111111
        TRISD  =  %00000000
        ADCON1 = 7
        direction = 0
        portd  = %10000000
LOOP:
        IF (S1 = 0) and (S2 = 1) THEN direction = 1
        if (S1 = 1) and (S2 = 0) THEN direction = 0
        pause 50
        if direction = 1 then
            portd = portd >> 1
            if portd = %00000000 then portd = %10000000
        else
            portd = portd << 1
            if portd = %00000000 then portd = %00000001
        endif
        pause 150
        GOTO LOOP
END

```

บันทึกผลการทดลอง

.....

.....

.....

.....

แบบฝึกหัด

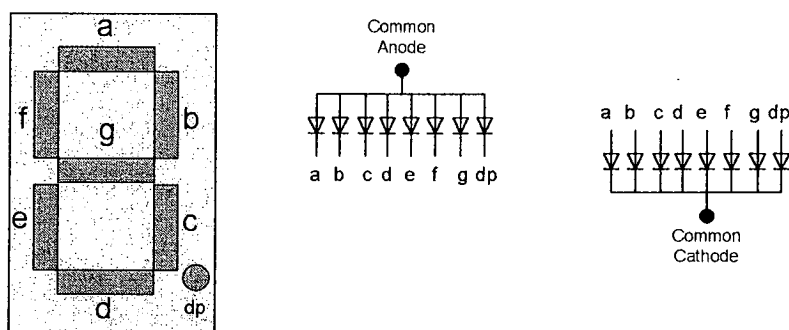
1. ถ้าเปลี่ยนคำสั่ง `portd = portd >> 1` เป็นคำสั่ง `portd = portd << 1` และคำสั่ง `portd = portd << 1` เป็นคำสั่ง `portd = portd >> 1` ในโปรแกรมที่ 1 ผลจะเป็นอย่างไร
2. คำสั่ง `ADCON1 = 7` มีความหมายว่าอย่างไร

การทดลองที่ 2.8 การเขียนโปรแกรมเพื่อควบคุมการแสดงผลทาง LED 7- Segment

จุดประสงค์ เพื่อศึกษาการเขียนโปรแกรมในการควบคุมการแสดงผลทาง LED 7- Segment แบบ Common Anode

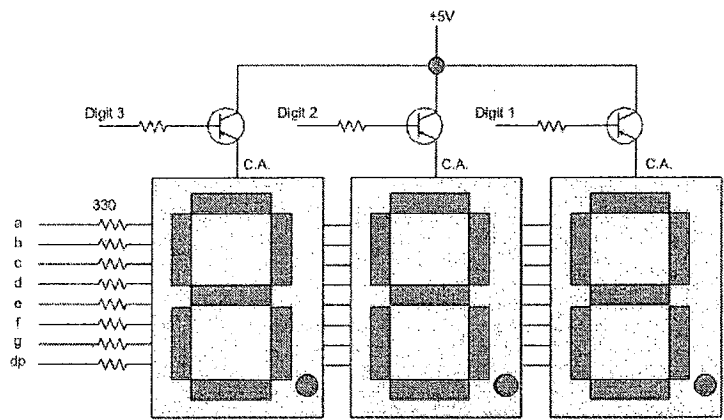
ทฤษฎีพื้นฐาน

การแสดงผลตัวเลขแบบ LED 7 – Segment ปัจจุบันยังคงมีใช้งานอย่างต่อเนื่องเนื่องจากมีความสว่างเห็น และอ่านตัวเลขได้ชัดเจน LED 7 – Segment แบบตัวเล็กขนาดความสูงไม่เกิน 1 นิ้ว ส่วนมากจะสร้างเป็นโมดูล โมดูลละ 1 หลัก 2 หลัก 4 หลัก หรือมากกว่านั้น แต่ละโมดูลจะมีโครงสร้างประกอบด้วย หลอด LED ต่อเรียงกันเป็นตัวเลข 7 ส่วน การต่อจะมี 2 แบบ คือ แบบอะโนดร่วม (Common Anode) และแคโทดร่วม (Common Cathode)

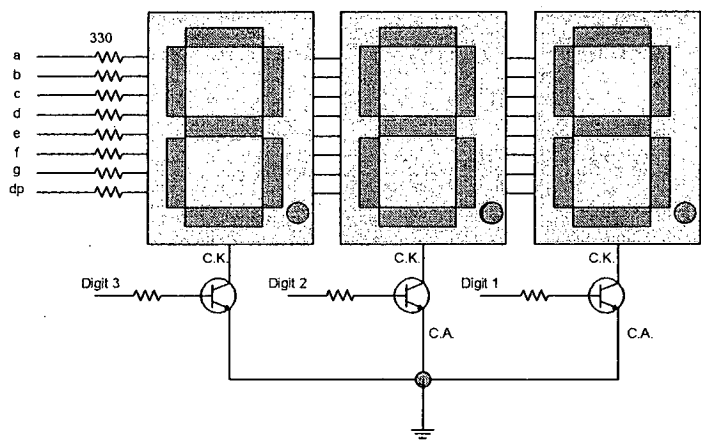


รูปที่ 1 แสดงการต่อวงจรภายในโมดูล LED 7-Segmentแบบ Common Anode และ แบบ Common Cathode

การต่อวงจรเพื่อแสดงผลสำหรับโมดูลแบบ Common Anode จะต้องต่อไฟ +5V เข้าที่ขา Common และหากต้องการให้ Segment ใดติดสว่าง จะต้องต่อ Segment นั้นลง Ground หรือป้อนลอจิก 0 ส่วนโมดูลแบบ Common Cathode จะตรงกันข้าม คือ ที่ขา Common ต้องต่อลง Ground และต้องป้อนไฟบวก หรือลอจิก 1 ที่ Segment ที่ต้องการให้ติดสว่าง ในการออกแบบวงจรขับจะต้องต่อตัวต้านทานอนุกรมเข้าที่ขา Segment เพื่อจำกัดกระแสไหลเข้าหลอด LED ซึ่งไม่ให้เกิน 20 mA ตัวต้านทานที่มาก่อนุกรมจะมีค่าระหว่าง 150 – 470 โอห์ม สำหรับกระแสรวมที่ขา Common หากติดทุก Segment จะมีค่าประมาณ 100 – 160 mA ดังนั้นในการขับหลายโมดูลแบบมัลติเพล็กซ์ ไม่สามารถต่อกับขา I/O ของไมโครโปรเซสเซอร์ ได้โดยตรง จำเป็นต้องใช้ทรานซิสเตอร์เป็นตัวขับ ตามรูปที่ 2



รูปที่ 2 แสดงการต่อแสดงผล 3 หลักแบบมัลติเพล็กซ์ของโมดูลแบบ Common Anode



รูปที่ 3 แสดงการต่อแสดงผลแบบมัลติเพล็กซ์ของโมดูลแบบ Common cathode

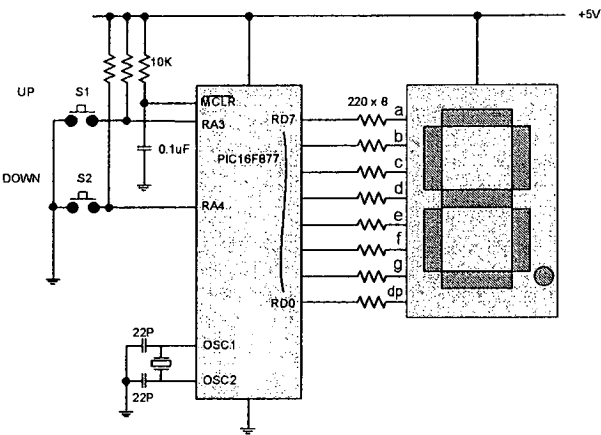
การต่อวงจรแสดงผลแบบมัลติเพล็กซ์ตามรูปที่ 2 และ 3 ขา Segment จะต่อพ่วงกันทุกหลัก และขา Common จะต้องใช้ทรานซิสเตอร์มาขับ เทคนิคในการแสดงผล 3 หลักแบบมัลติเพล็กซ์ จะต้องเขียนโปรแกรมให้ส่งเข้าพุทออกมาเปิดแสดงผลทีละหลัก(Digit) เรียงกันไป โดยเริ่มตั้งแต่หลักหน่วย ไปจนถึงหลักร้อย แสดงซ้ำ ๆ กันด้วยความเร็วเกิน 30 ครั้ง ต่อนาทีสายตาเราจะมองเห็นติดทั้ง 3 หลักพร้อม ๆ กัน เทคนิคการแสดงผลแบบมัลติเพล็กซ์มีข้อดี คือลดจำนวนขา I/O ของไมโครโปรเซสเซอร์ ที่จะใช้ขับ แต่มีข้อเสีย คือโปรแกรมจะต้องเสียเวลาส่วนใหญในการวนกลับมาขับแสดงผลซ้ำเพื่อไม่ให้กระพริบ

สำหรับการแสดงผลแบบหลักเดียวเราสามารถต่อขา Common เข้ากับแหล่งจ่ายไฟได้โดยตรงโดยไม่ต้องใช้ทรานซิสเตอร์มาขับ ตามวงจรในบอร์ดทดลองจะต้องถอด Jumper ทั้ง 8 ตัวออกเพื่อใช้สายต่อขา I/O เข้ากับขา Segment โดยตรงในการเขียนโปรแกรมเราต้องใช้คำสั่ง LOOKUP มาใช้เพื่อเก็บค่ารหัสที่ขับ Segment เป็นตัวเลข 0 - 9 ไว้ในตารางรหัสขับ Segment

สำหรับโมดูลแบบ Common Anode ได้จัดทำเป็นเลขฐานสิบหก(Hexa-decimal) ตามตารางรูปที่ 4

Decimal	Hex	dp	g	f	e	d	c	b	a
0	\$C0	1	1	0	0	0	0	0	0
1	\$F9	1	1	1	1	1	0	0	1
2	\$A4	1	0	1	0	0	1	0	0
3	\$B0	1	0	1	1	0	0	0	0
4	\$99	1	0	0	1	1	0	0	1
5	\$92	1	0	0	1	0	0	1	0
6	\$82	1	0	0	0	0	0	1	0
7	\$F8	1	1	1	1	1	0	0	0
8	\$80	1	0	0	0	0	0	0	0
9	\$90	1	0	0	1	0	0	0	0
dp		0							

รูปที่ 4 แสดงตารางเลขรหัสขับแสดงตัวเลข 0 -9 ของโมดูล LED 7-Segment แบบ Common Anode



โปรแกรมคำสั่ง

การทำงานของโปรแกรม เริ่มต้นโปรแกรมจะส่งเอาพุทขับแสดงตัวเลขนับขึ้นตลอดเวลา การนับขึ้น หรือลงโดยการกดสวิตช์ S1 และ S2

```

S1      var   PORTA.3 'for count up
S2      var   PORTA.4 'for count down
num     var   byte
disp    var   byte
direction var bit
        TRISA = %111111
        TRISD = %00000000
        ADCON1 = 7
        direction = 1
        num = 0
        portd = 255
LOOP:
        IF (S1 = 0) and (S2 = 1) THEN direction = 1
        IF (S1 = 1) and (S2 = 0) THEN direction = 0
        pause 50
        if direction = 1 then
            lookup num,[$c0,$f9,$a4,$b0,$99,_,
                        $92,$82,$f8,$80,$90],disp
            portd = disp
            num = num + 1
            if num > 9 then num = 0
        else
            lookup num,[$c0,$f9,$a4,$b0,$99,_,
                        $92,$82,$f8,$80,$90],disp
            portd = disp
            if num = 0 then num = 10
            num = num - 1
        endif
        pause 1000
        GOTO LOOP
END

```

บันทึกผลการทดลอง

.....

.....

.....

.....

แบบฝึกหัด

1. ถ้าเปลี่ยน LED 7 Segment ในการทดลองเป็น ชนิด Common Cathode จะต้องแก้ไข ส่วนใดของโปรแกรม

การทดลองที่ 3.1 การเขียนโปรแกรมแปลงสัญญาณอนาล็อกเป็นดิจิตอล(A/D) แบบ 8 บิต

จุดประสงค์ เพื่อศึกษาการเขียนโปรแกรมในการแปลงสัญญาณอนาล็อกเป็นดิจิตอล(A/D) และแสดงผลทางจอ LCD

ทฤษฎีพื้นฐาน

การแปลงสัญญาณอนาล็อกเป็นดิจิตอลเป็นวงจรสำคัญในการนำไปใช้ในการตรวจจับ (Sensor) สัญญาณที่เป็นระดับเช่น ความดัน ระดับของเหลว อุณหภูมิ เป็นต้น ในการตรวจจับ สัญญาณดังกล่าว อันดับแรกต้องมีวงจรแปลงสัญญาณเหล่านี้เป็นสัญญาณทางไฟฟ้าก่อน จากนั้น จึงได้แปลงมาเป็นสัญญาณดิจิตอล ระดับสัญญาณทางไฟฟ้าโดยปกติจะมีค่ามาตรฐานอยู่ที่ 0 – 5 Vdc หรือ 0 – 2.5Vdc ปัจจุบันผู้ผลิตชิพไมโครโปรเซสเซอร์ จะฝังวงจร A/D เข้าไปไว้ในตัว ชิพ โดยสามารถป้อนสัญญาณอนาล็อกมาตรฐานเข้าที่ขา I/O ได้โดยตรง โดยมีรีจิสเตอร์ และ คำสั่งที่เป็นฟังก์ชันการทำงานไว้เบ็ดเสร็จ และถือเป็นขีดความสามารถในการแข่งขันในการพัฒนา ชิพ MCU อย่างหนึ่ง สำหรับชิพ MCU ของไมโครชิพเบอร์ PIC16F877 รวมถึงเบอร์อื่น ๆ ที่มี 40 ขา จะมีขา I/O ที่รับสัญญาณ A/D ได้ถึง 8 ช่องโดยมีพอร์ท A จำนวน 5 ขาและ พอร์ท E จำนวน 3 ขา วงจร A/D ที่อยู่ในตัวชิพจะเป็นแบบมัลติเพล็กซ์ ความเร็วในการแปลงสัญญาณ ต่อช่องประมาณ 18 ไมโครวินาทีที่ความละเอียด 8 บิตสำหรับความละเอียดสามารถกำหนดได้ 2 ระดับคือ ระดับ 8 บิต และระดับ 10 บิต โดยระดับ 8 บิตจะแปลงสัญญาณอนาล็อก 0 – 5 V มาเป็นสัญญาณดิจิตอลได้ 256 ค่า และระดับ 10 บิตจะแปลงได้ 1096 ค่า ซึ่งมีความละเอียด กว่ามาก แต่จะใช้เวลาในการแปลงมากกว่า 8 บิต ในการแปลงค่า A/D จะมีรีจิสเตอร์ที่ทำหน้าที่อยู่ 4 ตัว ได้แก่

- ADCON0 เป็นรีจิสเตอร์ควบคุมการทำงาน A/D ตัวที่ 1
 - ADCON1 เป็นรีจิสเตอร์ควบคุมการทำงาน A/D ตัวที่ 2
 - ADRESH เป็นรีจิสเตอร์เก็บค่าผลลัพธ์การแปลง A/D ในไบท์สูง
 - ADRESL เป็นรีจิสเตอร์เก็บค่าผลลัพธ์การแปลง A/D ในไบท์ต่ำ
- สำหรับโมดูล A/D ของ MCU เบอร์นี้ยังคงทำงานได้แม้จะอยู่ใน Sleep Mode และยัง

จะเชื่อมต่อกับระบบการ Interrupt

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on MCLR, WDT
0Bh,8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	TDIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
1Eh	ADRESH	A/D Result Register High Byte								xxxx xxxx	nnnn nnnn
9Eh	ADRESL	A/D Result Register Low Byte								xxxx xxxx	nnnn nnnn
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS0	GO/DONE			ADON	0000 00-0	0000 00-0
9Fh	ADCON1	ADFM				PCFG3	PCFG2	PCFG1	PCFG0	--0- 0000	--0- 0000
85h	TRISA			PORTA Data Direction Register						--11 1111	--11 1111
05h	PORTA			PORTA Data Latch when written; PORTA pins when read						--0x 0000	--0u 0000
89h ⁽¹⁾	TRISE	IBF	OBF	ISOV	PSPMODE		PORTE Data Direction bits			0000 -111	0000 -111
09h ⁽¹⁾	PORTE						RE2	RE1	RE0	---- -xxx	---- -nnn

Legend: x = unknown, u = unimplemented, read as 0'. Shaded cells are not used for A/D conversion.
Note 1: These registers/bits are not available on the 28-pin devices.

ตารางที่ 1 แสดงค่ารีจิสเตอร์ที่เกี่ยวข้องกับการทำงาน A/D และค่าเมื่อเกิดสถานะเริ่มทำงาน

REGISTER 11-1: ADCON0 REGISTER (ADDRESS: 1Fh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	ADON	
bit 7							bit 0

- bit 7-6
ADCS1:ADCS0: A/D Conversion Clock Select bits
00 = FOSC/2
01 = FOSC/8
10 = FOSC/32
11 = FRC (clock derived from the internal A/D module RC oscillator)
- bit 5-3
CHS2:CHS0: Analog Channel Select bits
000 = channel 0, (RA0/AN0)
001 = channel 1, (RA1/AN1)
010 = channel 2, (RA2/AN2)
011 = channel 3, (RA3/AN3)
100 = channel 4, (RA5/AN4)
101 = channel 5, (RE0/AN5)⁽¹⁾
110 = channel 6, (RE1/AN6)⁽¹⁾
111 = channel 7, (RE2/AN7)⁽¹⁾
- bit 2
GO/DONE: A/D Conversion Status bit
If ADON = 1:
1 = A/D conversion in progress (setting this bit starts the A/D conversion)
0 = A/D conversion not in progress (this bit is automatically cleared by hardware when the A/D conversion is complete)
- bit 1
Unimplemented: Read as '0'
- bit 0
ADON: A/D On bit
1 = A/D converter module is operating
0 = A/D converter module is shut-off and consumes no operating current

Note 1: These channels are not available on PIC16F873/876 devices.

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

REGISTER 11-2: ADCON1 REGISTER (ADDRESS 9Fh)

U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM				PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

- bit 7
ADFM: A/D Result Format Select bit
1 = Right justified. 6 Most Significant bits of ADRESH are read as '0'.
0 = Left justified. 6 Least Significant bits of ADRESL are read as '0'.
- bit 6-4
Unimplemented: Read as '0'
- bit 3-0
PCFG3:PCFG0: A/D Port Configuration Control bits:

PCFG3: PCFG0	AN7 ⁽¹⁾ RE2	AN6 ⁽¹⁾ RE1	AN5 ⁽¹⁾ RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	VREF+	VREF-	Chan/ Refs ⁽²⁾
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	RA3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0011	D	D	D	A	VREF+	A	A	A	RA3	VSS	4/1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0
0101	D	D	D	D	VREF+	D	A	A	RA3	VSS	2/1
011x	D	D	D	D	D	D	D	D	VDD	VSS	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	RA3	RA2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	5/0
1010	D	D	A	A	VREF+	A	A	A	RA3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	RA3	RA2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	RA3	RA2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	RA3	RA2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	RA3	RA2	1/2

A = Analog Input D = Digital I/O

REGISTER 11-2: ADCON1 REGISTER (ADDRESS 9Fh)

U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM				PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

- bit 7
ADFM: A/D Result Format Select bit
1 = Right justified. 6 Most Significant bits of ADRESH are read as '0'.

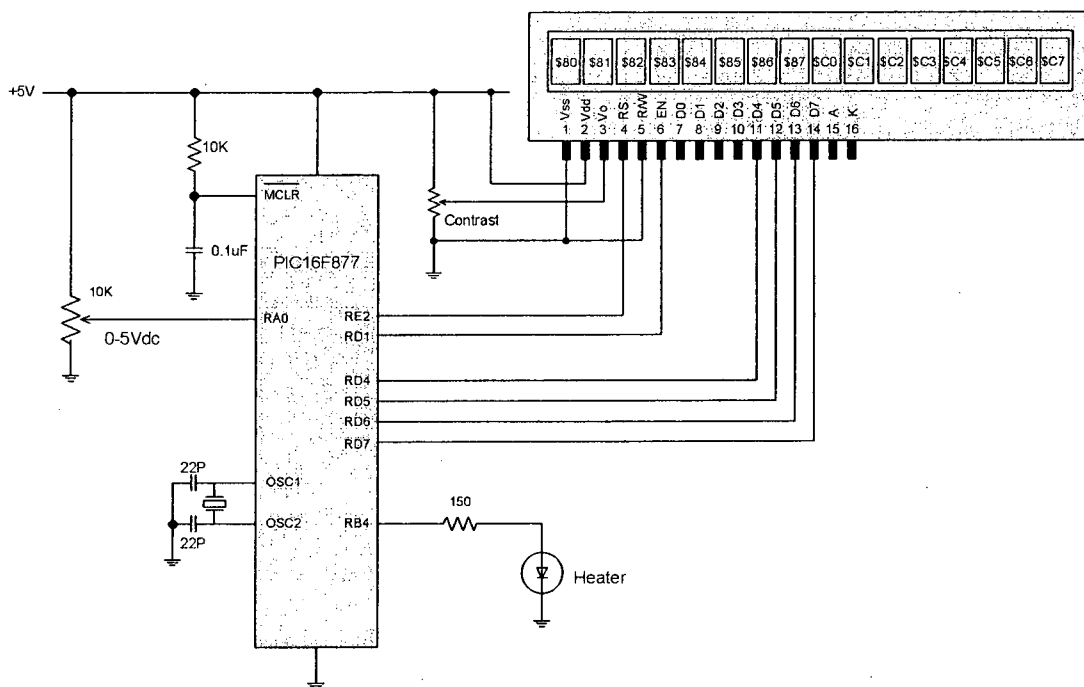
ค่าความเร็วในการแปลง A/D สามารถคำนวณได้จากสมการต่อไปนี้

$$\begin{aligned}
 T_{ACQ} &= \text{Amplifier Settling Time} + \\
 &\quad \text{Hold Capacitor Charging Time} + \\
 &\quad \text{Temperature Coefficient} \\
 &= T_{AMP} + T_C + T_{COFF} \\
 &= 2\mu s + T_C + [(Temperature - 25^\circ C)(0.05\mu s/^\circ C)] \\
 T_C &= CHOLD (R_{IC} + R_{SS} + R_S) \ln(1/2047) \\
 &= -120pF (1k\Omega + 7k\Omega + 10k\Omega) \ln(0.0004885) \\
 &= 16.47\mu s \\
 T_{ACQ} &= 2\mu s + 16.47\mu s + [(50^\circ C - 25^\circ C)(0.05\mu s/^\circ C)] \\
 &= 19.72\mu s
 \end{aligned}$$

การออกแบบวงจรป้อนสัญญาณ A/D จะต้องกำหนดให้มีค่าอิมพีแดนซ์ (Input impedance) ประมาณ 10 กิโลโอห์ม ขั้นตอนในการแปลงสัญญาณ A/D สำหรับในภาษาแอสเซมบลีจะมีความซับซ้อนบ้าง รายละเอียดให้ศึกษาเพิ่มเติมใน Data sheet

สำหรับการใช้ภาษา Pic Basic Pro Compiler จะมีรีจิสเตอร์ที่เกี่ยวข้องเพียงตัวเดียวคือ ADCON1 ซึ่งจะเป็นตัวกำหนดว่า I/O ขาใดของพอร์ต A และ E ที่จะรับสัญญาณดิจิทัลหรืออนาล็อก และจะจัดเรียงบิตชิตไปทางซ้าย หรือชิตไปทางขวา(กรณีการแปลง A/D แบบ 10 บิต)

วงจรทดลองตาม การทดลองที่ 3.1



คำสั่งที่ใช้ในการแปลง A/D คือ ADCIN หมายเลขช่อง, ตัวแปรที่เก็บค่าผลลัพธ์รายละเอียด
ให้ศึกษาเพิ่มเติมในคู่มือการใช้โปรแกรม Pic Basic Pro Compiler โปรแกรมคำสั่ง

```

DEFINE LCD_DREG PORTD
DEFINE LCD_DBIT 4
DEFINE LCD_RSREG PORTE
DEFINE LCD_RSBIT 2
DEFINE LCD_EREG PORTD
DEFINE LCD_EBIT 1
heater var portb.4
temp var byte
temp1 var byte
    TRISA = %111111
    ADCON1 = 0
'----- main program -----
    adcin 0,temp
    adcon1 = 7
    lcdout $fe,1,"Temp = "
    lcdout $fe,$c0,dec temp,$fe,$c4,"C"
,
LOOP: gosub get_temp
    if temp <> temp1 then
        gosub display
        gosub action
        temp1 = temp
    endif
    PAUSE 50
    GOTO LOOP
END
'----- End of Main Program -----
,
'----- Subroutine Start Here -----
,
get_temp:
    adcon1 = 0
    adcin 0,temp
    pause 50
    return
,
display:
    adcon1 = 7
    lcdout $fe,$c0," "
    lcdout $fe,$c0,dec temp
    return
,
action:
    if temp > 25 then
        low heater
    else
        high heater
    endif
    return
,
'----- End of Subroutine -----

```

บันทึกผลการทดลอง

.....

.....

.....

.....

แบบฝึกหัด

- 1. จำนวนช่องสัญญาณ A/D ของไมโครโปรเซสเซอร์ PIC 16F877 มีจำนวนกี่ช่องสามารถใช้
งานได้พร้อมกันในเวลาเดียวกันได้หรือไม่อย่างไร
- 2. การรับสัญญาณอนาลอกที่ PORT A กับ PORT E มีความแตกต่างกันหรือไม่อย่างไร

การทดลองที่ 3.2 การเขียนโปรแกรมแปลงสัญญาณอนาล็อกเป็นดิจิตอล(A/D) แบบ 10 บิต

จุดประสงค์ เพื่อศึกษาการเขียนโปรแกรมในการแปลงสัญญาณอนาล็อกเป็นดิจิตอล(A/D) ที่ความละเอียด 10 บิตและแสดงผลทางจอ LCD

ทฤษฎีพื้นฐาน

ในการแปลงสัญญาณ A/D แบบ 10 บิตจะมีข้อแตกต่างกับการแปลงแบบ 8 บิต คือ

1. ค่าสัญญาณดิจิตอลที่แปลงได้จะมากกว่า 3 หลัก(เกินกว่า 256) ดังนั้นการกำหนดตัวแปรที่ใช้เก็บค่าต้องมีขนาด Word (16 bit) เพื่อรองรับค่า A/D ค่า 0 - 1096
2. ค่าที่แปลงได้จะถูกนำเก็บพักไว้ที่รีจิสเตอร์ 2 ตัว คือ ADRESH และ ADRESL ดังนั้นจะต้องมีการจัดเรียงให้ชิดขอบขวา (Right Justify) ก่อน โดยจะต้องกำหนดบิตที่ 7 ของรีจิสเตอร์ ADCON1 ดังนี้ คือ

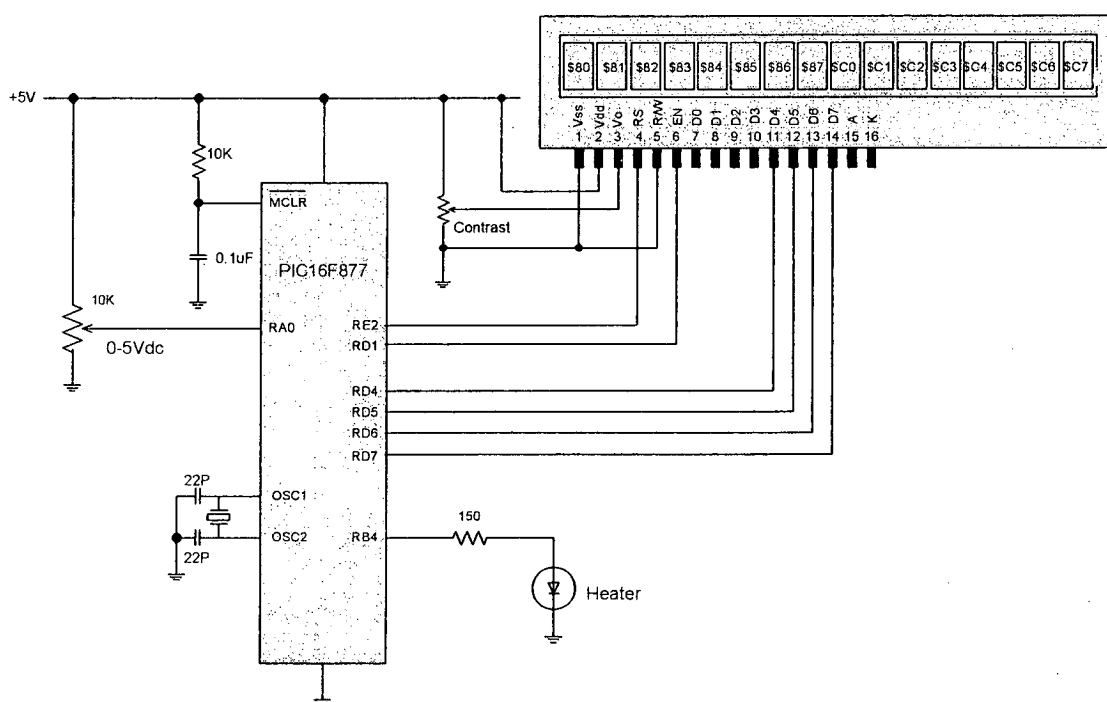
$$\text{ADCON1} . 7 = 1$$

$$\text{หรือ } \text{ADCON1} = \%10000000$$

3. ต้องนิยามการแปลง A/D แบบ 10 บิต คือ DEFINE ADC_BITS 10 ไว้ที่หัวของโปรแกรม

4. ในการแสดงผลผ่าน LED 7-Segment หรือ LCD จะต้องกำหนดเนื้อที่การแสดงค่าตัวเลขไว้อย่างน้อย 4 หลัก เนื่องจากค่าดิจิตอลที่แปลงได้สูงสุดเกินหลักพันในฐานสิบ

วงจรทดลองตาม การทดลองที่ 3.2



โปรแกรมคำสั่ง

```

DEFINE LCD_DREG PORTD
DEFINE LCD_DBIT 4
DEFINE LCD_RSREG PORTE
DEFINE LCD_RSBIT 2
DEFINE LCD_EREG PORTD
DEFINE LCD_EBIT 1
DEFINE ADC_BITS 10
,

heater var portb.4
temp var word
temp1 var word
    TRISA = %111111
    ADCON1 = %10000000
'----- main program -----
    adcin 0,temp
    adcon1 = 7
    lcdout $fe,1,"Temp = "
    lcdout $fe,$c0,dec temp,$fe,$c5,"C"
,
LOOP: gosub get_temp
    if temp <> temp1 then
        gosub display
        gosub action
        temp1 = temp
    endif
    PAUSE 50
    GOTO LOOP
END
'----- End of Main Program -----
,
'----- Subroutine Start Here -----
,
get_temp:
    adcon1 = %10000000
    adcin 0,temp
    pause 50
    return
,
display:
    adcon1 = 7
    lcdout $fe,$c0," "
    lcdout $fe,$c0,dec temp
    return
,
action:
    if temp > 25 then
        low heater
    else
        high heater
    endif
    return
,
'----- End of Subroutine -----

```

บันทึกผลการทดลอง

แบบฝึกหัด

1.

ถ้าต้องการเปลี่ยนค่าการแสดงผลอุณหภูมิให้มีหน่วยเป็นฟาเรนไฮต์ จะต้องแก้ไขโปรแกรมส่วนใดบ้าง

โปรแกรมคำสั่ง

```

DEFINE LCD_DREG PORTD
DEFINE LCD_DBIT 4
DEFINE LCD_RSREG PORTE
DEFINE LCD_RSBIT 2
DEFINE LCD_EREG PORTD
DEFINE LCD_EBIT 1
'
heater var portb.4
temp var byte
temp1 var byte
level var byte
level1 var byte
TRISA = %111111
ADCON1 = 0
'----- main program -----
    adcin 0,temp
    adcin 1,level
    adcon1 = 7
    lcdout $fe,1,"Temp="
    lcdout $fe,$85,dec temp
    lcdout $fe,$c0,"Lev="
    lcdout $fe,$c4,dec level
'
LOOP: gosub get_temp
    if (temp <> temp1) or (level <> level1) then
        gosub display
        gosub action
        temp1 = temp
        level1 = level
    endif
    PAUSE 50
    GOTO LOOP
END
'----- End of Main Program -----
'
'----- Subroutine Start Here -----
'
get_temp:
    adcon1 = 0
    adcin 0,temp
    adcin 1,level
    pause 50
    return
'
display:
    adcon1 = 7
    lcdout $fe,$85," "
    lcdout $fe,$85,dec temp
    lcdout $fe,$c4," "
    lcdout $fe,$c4,dec level
    return
'
action:
    if temp = level then
        low heater
    else
        high heater
    endif
    return
'
'----- End of Subroutine -----

```

บันทึกผลการทดลอง

.....

.....

.....

.....

แบบฝึกหัด

- 1. โปรแกรมการทดลองจางเขียนไฟล์ชาร์ตการทำงาน

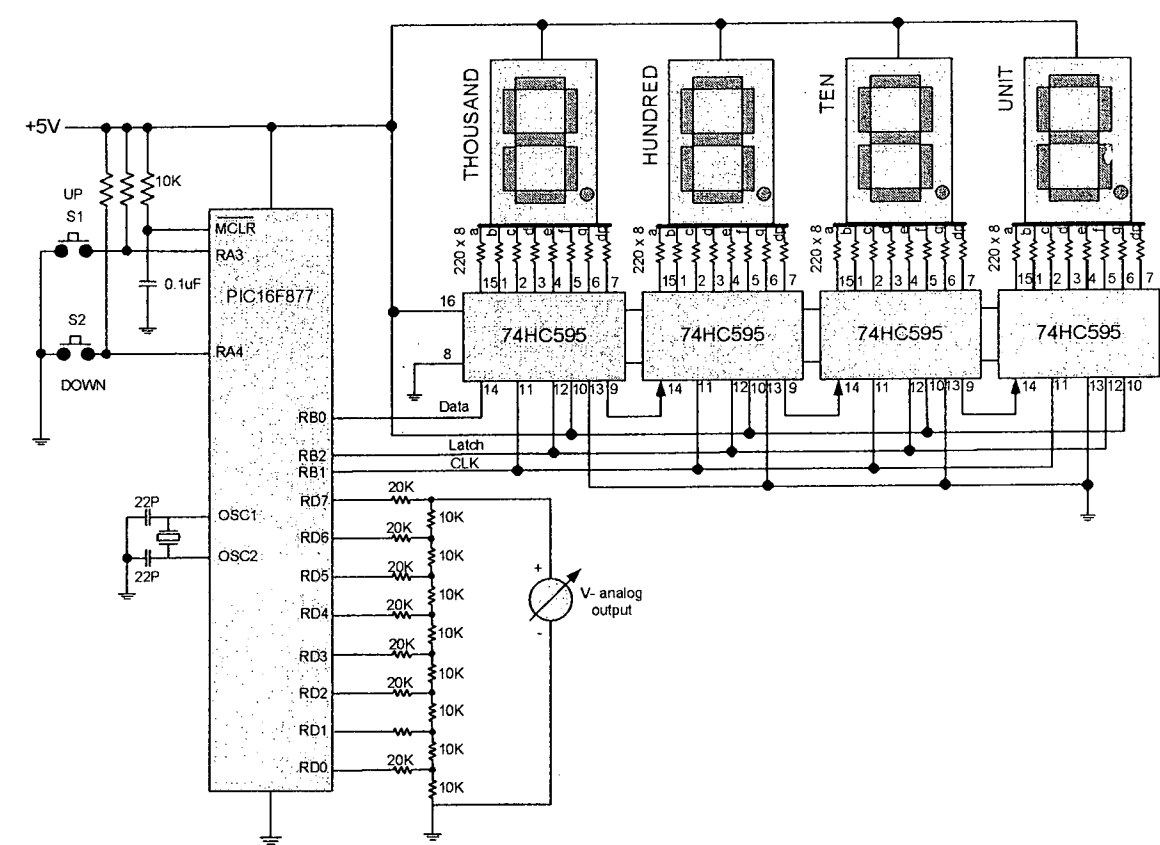
การทดลองที่ 3.4 การเขียนโปรแกรมแปลงสัญญาณดิจิทัล เป็นอนาล็อก (D/A)

จุดประสงค์ เพื่อศึกษาการเขียนโปรแกรมเพื่อแปลงสัญญาณดิจิทัลเป็นอนาล็อก (D/A) ขนาด 8 บิต โดยใช้วงจร R-2R Ladder Network

ทฤษฎีพื้นฐาน

การแปลงสัญญาณดิจิทัล เป็นอนาล็อก มีความจำเป็นที่จะต้องสร้างวงจรอยู่นอกตัวไมโครโปรเซสเซอร์ เนื่องจากผู้ผลิตชิพ MCU ส่วนมากไม่ได้สร้างบรรจุไว้เนื่องจากวงจร D/A ใช้อุปกรณ์ประเภท Passive ได้แก่ตัวต้านทาน เป็นส่วนมาก ซึ่งจะทำให้ต้องใช้แผ่นชิพที่มีขนาดใหญ่ขึ้น และมีความร้อนสะสมในตัวชิพมาก การประยุกต์แปลงสัญญาณ D/A ส่วนมากเพื่อสร้างสัญญาณควบคุมแบบลิเนียร์ เช่น มิเตอร์แบบมีเข็ม สร้างสัญญาณรูปคลื่นแบบต่าง ๆ เป็นต้น

วงจรแปลงสัญญาณ D/A มีหลายแบบ แต่ที่นิยมใช้กันมากมี 2 แบบ ได้แก่ ใช้ไอซี D/A และวงจร R-2R Ladder network สำหรับการทดลองตามใบงานนี้จะใช้วงจร R-2R Ladder network เนื่องจากมีราคาถูก เป็นวงจรที่ง่ายไม่ซับซ้อน และอุปกรณ์เป็นตัวต้านทานเพียง 2 ค่า ที่สามารถหาซื้อได้ง่าย



รูปที่ 1 แสดงวงจรการแปลงสัญญาณดิจิทัลเป็นอนาล็อกโดยใช้วงจร R-2R Ladder

การทำงานของโปรแกรม ตามวงจรรูปแสดงวงจรการแปลงสัญญาณดิจิทัลเป็นอนาล็อก
โดยใช้วงจร R-2R Ladder

เมื่อเริ่มเปิดไฟเข้าวงจร โปรแกรมจะเริ่มนับจาก 0 ไปถึง 255 ทุกค่าที่นับจะส่งออกไปที่ พอร์ต D และจะค้างอยู่ที่ 255 ถ้าต้องการให้นับลง กด S2 ต้องการให้นับขึ้น กด S1 ค่าดิจิตอล จะแสดงด้วยตัวเลข LED 7 - Segment ขนาด 3 หลัก

โปรแกรมคำสั่ง

```

INCLUDE"MODEFS.BAS"
DEFINE SHIFT_PAUSEUS 100
s1      var    porta.3
s2      var    porta.4
dat     var    portb.0
clk     var    portb.1
latch   var    portb.2
stat     var    bit
unit     var    byte
ten      var    byte
hun      var    byte
thou     var    byte
digit    var    byte
num      var    byte
b0       var    byte
b1       var    byte

TRISA    =    %111111
TRISD    =    %0000000
ADCON1   =    7
stat      =    1
thou      =    255
num       =    0

,

begin:
    IF (S1 = 0) and (S2 = 1) THEN stat = 1
    if (S1 = 1) and (S2 = 0) THEN stat = 0
    pause 50
    if stat = 1 then
        num = num + 1
        gosub display
        portd = num
        if num = 255 then num = 254
    else
        num = num - 1
        gosub display
        portd = num
        if num = 0 then num = 1
    endif
    pause 80
    GOTO begin
END

'-----End of program -----
display:
    digit = num dig 0
    lookup digit,[ $c0,$f9,$a4,$b0,$99,___
                    $92,$82,$f8,$80,$90],unit
    digit = num dig 1
    lookup digit,[ $c0,$f9,$a4,$b0,$99,___
                    $92,$82,$f8,$80,$90],ten
    digit = num dig 2
    lookup digit,[ $c0,$f9,$a4,$b0,$99,___
                    $92,$82,$f8,$80,$90],hun
    high latch
    pauseus 10
    ShiftOut dat,clk,1,[unit,ten,hun,thou]

```

Pauseus 10
Low latch
Pauseus 10
high latch
Return

บันทึกผลการทดลอง

.....

.....

.....

.....

แบบฝึกหัด

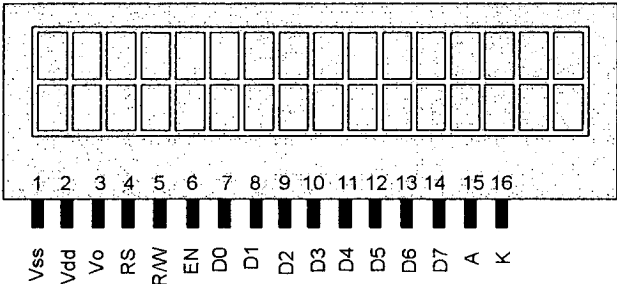
- 1. จากการทดลองสามารถสรุปการทำงานของวงจร R-2R Ladder ได้อย่างไร

การทดลองที่ 4.1 การเขียนโปรแกรมเพื่อแสดงผลผ่านจอแอลซีดี (LCD Display)

จุดประสงค์ เพื่อศึกษาการเขียนโปรแกรมในการเพื่อนำข้อมูลออกแสดงผลทางจอแอลซีดีในลักษณะต่าง ๆ

ทฤษฎีพื้นฐาน

จอแอลซีดี(LCD: Liquid Crystal Display) เป็นอุปกรณ์แสดงผลที่นิยมใช้กันมาก เนื่องจากสามารถแสดงข้อมูลที่เป็นทั้งตัวเลข และตัวอักษรที่เป็นข้อความ ใช้งานง่าย และกินกระแสไฟน้อยกว่า LED ตัวจอLCDผลิตเป็นโมดูล มีตัวไมโครโปรเซสเซอร์ ควบคุมการแสดงผลในตัวเอง สามารถแสดงผลได้ตั้งแต่บรรทัดละ 8 16 และ 32 ตัวอักษระ ตั้งแต่ 1 แถว ไปจนถึงหลายแถว จนถึงการแสดงแบบกราฟิก โปรแกรม Pic Basic Pro Compiler มีคำสั่งสนับสนุนการใช้งานกับจอ LCD ที่ใช้คอนโทรลเลอร์ ของ Hitachi 44780 หรือชิพเบอร์อื่นที่มีคุณสมบัติเดียวกัน เป็นตัวควบคุม ตัวโมดูลมีขา I/O ที่จะติดต่อกับ MCU จำนวน 14 และ 16 ขา ตามรูปที่ 1



รูปที่ 1 แสดงโครงสร้างของจอ LCD แบบ 16 ตัวอักษร x 2 บรรทัด

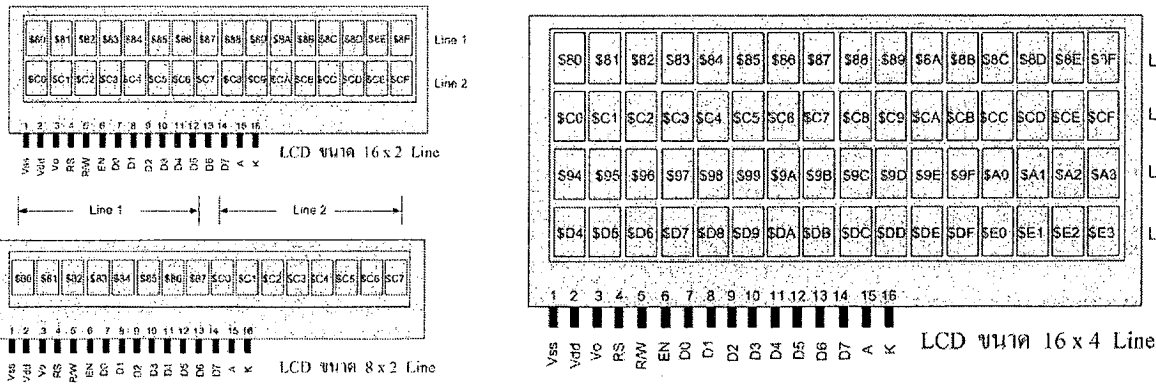
รายละเอียดการต่อขาจอ LCD มีดังนี้คือ

- ขา Vss เป็นขาต่อลงกราวด์ (GND)
- ขา Vdd เป็นขาจ่ายไฟ +5 V
- ขา Vo เป็นขาที่ต่อไฟสำหรับควบคุมความเข้ม (Contrast) ของตัวอักษรที่แสดงผล ปรับได้ 0 – 5 V ถ้าปรับเข้าใกล้ 0 V ตัวอักษรจะเข้มสุด หากไม่ต้องการปรับให้ต่อลง GND
- ขา RS เป็นขาที่ใช้ควบคุมการป้อนคำสั่ง กับข้อมูลที่แสดงผล
- ขา R/W เป็นขาที่ใช้ควบคุมว่าจะให้อ่านค่าตัวอักษรจากหน้าจอ หรือเขียนเขียนตัวอักษรลงไปที่หน้าจอ หากต้องการจะส่งข้อมูลออกแสดงผลเพียงอย่างเดียว ให้ต่อขานี้ลง GND
- ขา EN เป็นขาควบคุมสถานการณ์อ่าน หรือ แสดงผลข้อความ
- ขา D0 - D7 เป็นขาที่ส่งข้อมูลเข้า - ออก เพื่อแสดงผล
- ขา A และ ขา K เป็นขาที่ต่อไฟ Back Light สำหรับเป็นจอที่มี Back light ที่เหมาะกับการใช้งานในที่มืดไฟที่ต่อเป็นไฟตรง 5 V

ข้อความที่แสดงบนจอ แต่ละตัวอักษรจะมีหมายเลขตำแหน่งกำกับ โดยแต่ละแถว หรือ บรรทัดจะมีหมายเลขตำแหน่งเริ่มต้น ซึ่งเป็นเลขฐาน 16 และจะมีรหัสควบคุมดังต่อไปนี้ คือ

Command	Operation
\$FE, 1	Clear Display
\$FE, 2	Return home (beginning of the first line)
\$FE, \$0C	Cursor off
\$FE, \$0E	Underline cursor on
\$FE, \$0F	Blinking cursor on
\$FE, \$10	Move cursor left one position
\$FE, \$14	Move cursor right one position
\$FE, \$C0	Move cursor to beginning of second line
\$FE, \$94	Move cursor to beginning of third line
\$FE, \$D4	Move cursor to beginning of fourth line

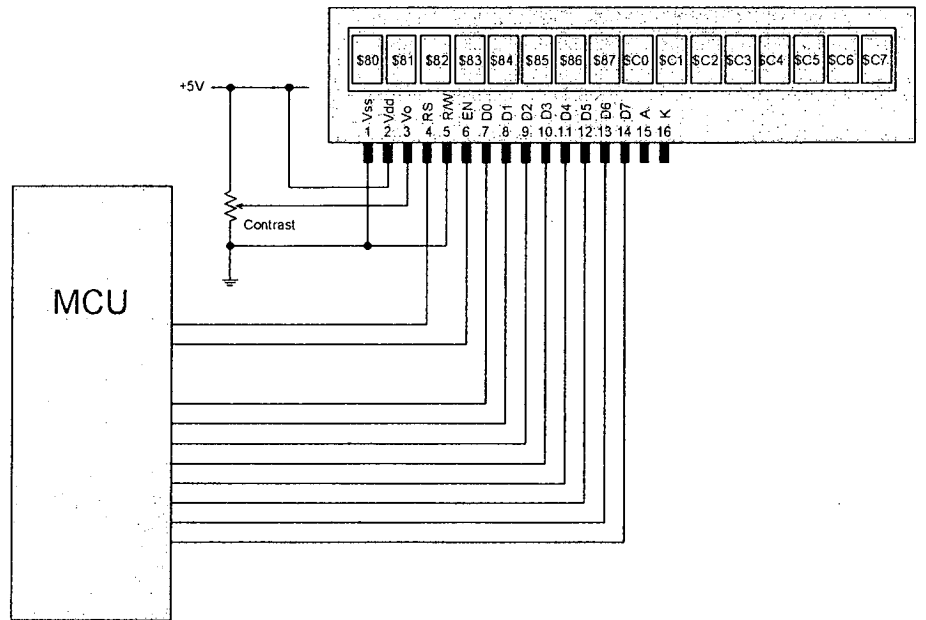
รหัสควบคุมเหล่านี้เป็นเลขฐานสิบหก การป้อนจะต้องมีรหัส \$FE, นำหน้าก่อนเสมอ การป้อนตัวอักษรเข้าแสดงที่จอ LCD ต้องระบุตำแหน่งเริ่มต้นก่อนเสมอ หมายเลขรหัสประจำตำแหน่งของตัวอักษรมีดังนี้



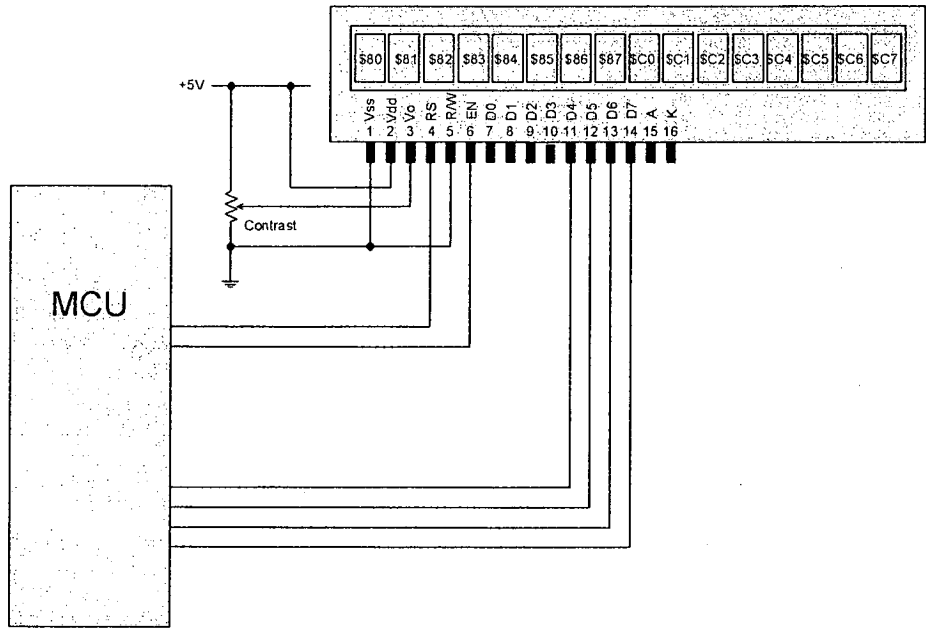
รูปที่ 2 แสดงโครงสร้าง และรหัสตำแหน่งตัวอักษรของ LCD แบบต่าง ๆ

นอกจากนี้ยังมี LCD ขนาดอื่นๆ อีก เช่น 16 x 1 Line เป็นต้น LCD ทุกรุ่นทุกขนาด จะมีขา I/O แบบเดียวกัน ถ้าเป็นแบบที่ใช้คอนโทรลเลอร์แบบเดียวกัน หรือเหมือนกัน

การเชื่อมต่อจอ LCD เข้ากับไมโครโปรเซสเซอร์ มีการต่อ 2 แบบ คือ แบบ 4 บิต และแบบ 8 บิต



รูปที่ 3 แสดงการต่อ LCD เข้ากับ MCU แบบ 8 บิต



รูปที่ 4 แสดงการต่อ LCD เข้ากับ MCU แบบ 4 บิต

การทดลองนี้ใช้การต่อวงจรแบบ 4 บิต ตามรูปที่ 4 เนื่องจากประหยัดขา I/O ของ MCU และในคำสั่งของ Pic Basic Pro ได้รองรับการทำงานแบบนี้อยู่แล้ว

ในการเขียนโปรแกรมการแสดงผลผ่านทางจอ LCD ก่อนการเข้าสู่การประมวลผลเราต้องให้โปรแกรมรู้จักตำแหน่งของขา I/O ที่ต่อกับจอ LCD ขาต่าง ๆ ตามรูปที่ 4 ก่อน ด้วยการนิยามไว้ที่ส่วนหัวของโปรแกรมตามตัวอย่างดังนี้ คือ

```
DEFINE LCD_DREG PORTD
DEFINE LCD_DBIT 4
DEFINE LCD_RSREG PORTE
DEFINE LCD_RSBIT 2
DEFINE LCD_EREG PORTD
DEFINE LCD_EBIT 1
```

ตามตัวอย่าง บรรทัดที่ 1 และ 2 กำหนดว่า ขา Data 4 เส้นต่อที่ Port D เริ่มบิตที่ 4 เป็นต้นไป บรรทัดที่ 3 และ 4 กำหนดว่า ขา RS ต่ออยู่ที่ Port E ขา 2 (RE2)

บรรทัดที่ 5 และ 6 กำหนดว่า ขา EN ต่ออยู่ที่ Port D ขา 1 (RD1)
รายละเอียดนอกเหนือจากนี้ ให้ศึกษาเพิ่มเติมในคู่มือการใช้โปรแกรม

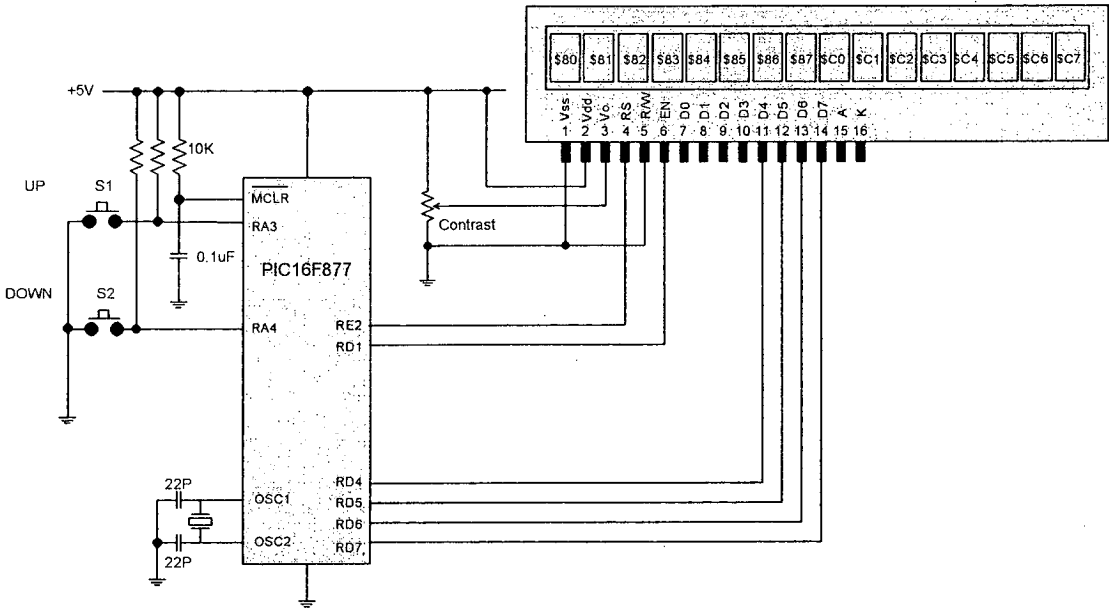
ในการแสดงผล เราสามารถปรับแต่งรูปแบบของข้อมูลที่จะส่งมาแสดงได้ เช่น เป็นเลขฐานสอง ฐานสิบ และฐานสิบหก เป็นต้น โดยจำมีตัวปรับค่าตามตารางดังต่อไปนี้

Modifier	Operation
BIN {1...16}	Send binary digit
DEC {1...5}	Send decimal digit
HEX {1...4}	Send hexa-decimal digit
REP c\ n	Send character c repeated n times
STR ArrayVar{\n}	Send string of n characters

หมายเหตุ

1. หากมีการต่อขาของจอ LCD เข้ากับพอร์ท ที่สามารถรับสัญญาณอนาล็อกได้ เช่น PortA และ PortE ก่อนการใช้คำสั่งแสดงผล จะต้องกำหนดให้พอร์ทดังกล่าวทำงานเป็นดิจิทัลก่อน ด้วยการกำหนดค่าในรีจิสเตอร์ ADCON1 เช่น ADCON1 = 7 เป็นต้น
2. เนื่องจากการแสดงผลของจอ LCD เป็นลักษณะ Static display เนื่องจากมีคอนโทรลเลอร์ของตัวเอง ตัวอักษรยังคงค้างจออยู่จากกว่าจะมีการส่งมาแสดงใหม่ ดังนั้นในการเขียนโปรแกรมแสดงผล ถ้าเป็นข้อความนิ่งๆ พยายามหลีกเลี่ยงการเขียนโปรแกรมวนมาแสดงซ้ำ ๆ ซึ่งอาจทำให้ตัวข้อความสั่นพลิ้ว และเสียเวลาการทำงานของโปรแกรมโดยไม่จำเป็น
3. การต่อไฟผิดขั้วจะทำให้จอLCD เสียหายได้จึงควรระมัดระวัง เนื่องมีราคาค่อนข้างแพง

วงจรทดลองตามการทดลองที่ 4.1



โปรแกรมคำสั่งที่ 1 ส่งข้อความแสดงใน Line 1 และ Line 2

```
DEFINE LCD_DREG PORTD
DEFINE LCD_DBIT 4
DEFINE LCD_RSREG PORTE
DEFINE LCD_RSBIT 2
DEFINE LCD_EREG PORTD
DEFINE LCD_EBIT 1
'-----
adcon1 = 7
lcdout $fe,1,"Hello.."
lcdout $fe,$c0,"World !"
end
```

บันทึกผลการทดลอง

.....

.....

.....

.....

โปรแกรมคำสั่งที่ 2 ส่งข้อความ และค่าตัวแปรเป็นเลขฐานสิบ

```

DEFINE LCD_DREG PORTD
DEFINE LCD_DBIT 4
DEFINE LCD_RSREG PORTE
DEFINE LCD_RSBIT 2
DEFINE LCD_EREG PORTD
DEFINE LCD_EBIT 1
temp var byte
'-----
adcon1 = 7
temp = 25
lcdout $fe,1,"Temp = "
lcdout $fe,$c0,dec temp,$fe,$c4,"C"
end

```

บันทึกผลการทดลอง

.....

.....

.....

.....

แบบฝึกหัด

1. ถ้าต้องการแสดงผล ELECTRONIC บรรทัดที่ 1 และ Pibulsongkram บรรทัดที่ 2 จะต้องเขียนโปรแกรมอย่างไร
2. จากวงจรการทดลองถ้าต้องการเขียนโปรแกรมสำหรับการกด S1 แสดงข้อความ ELECTRONIC บรรทัดที่ 1 เมื่อ กด S2 แสดงข้อความ ELECTRONIC บรรทัดที่ 2 และถ้ากดทั้ง S1 และ S2 แสดงข้อความ PSRU ELECTRONIC ทั้ง 2 บรรทัด

การทดลองที่ 4.2 การเขียนโปรแกรมประยุกต์การแสดงผลผ่านจอแอลซีดี (LCD Display)

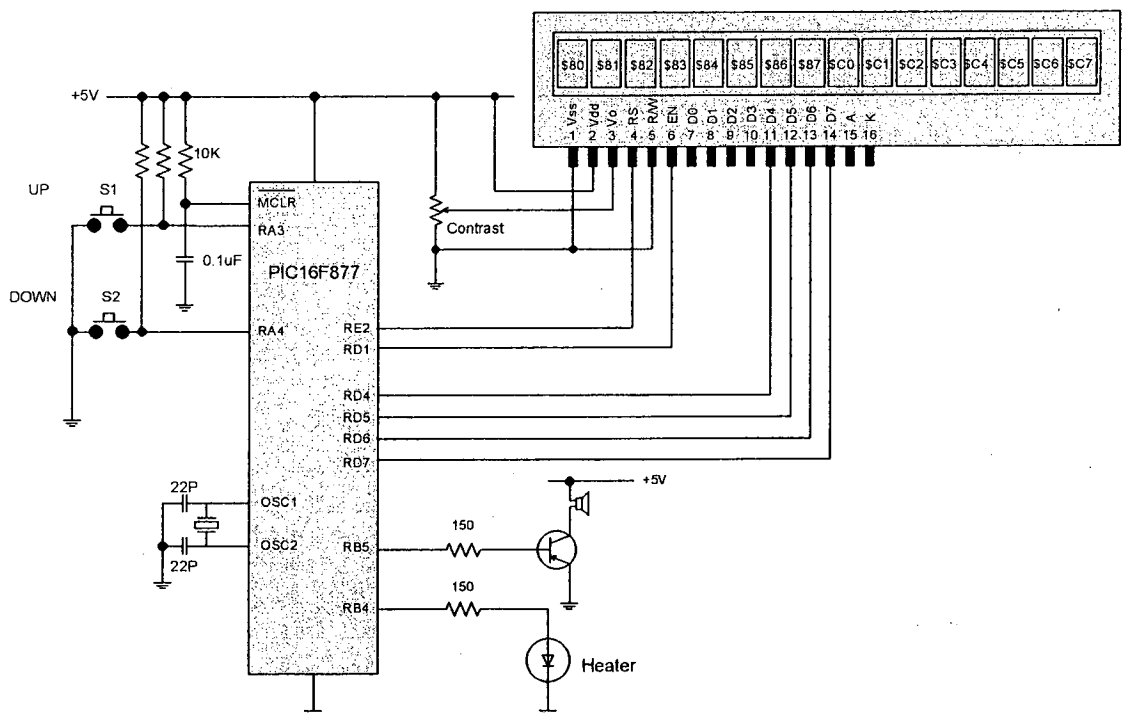
จุดประสงค์ เพื่อศึกษาการเขียนโปรแกรมในการประยุกต์การแสดงผลทางจอ LCD สำหรับใช้
งานในระบบการปรับควบคุมอุณหภูมิ หรือตู้ปรับอากาศ

ทฤษฎีพื้นฐาน

ระบบการทำงานของตู้อบความร้อนประกอบด้วย 4 ส่วน คือ ส่วนตัวตู้อบมีประตูปิด - เปิด มีฉนวนกันความร้อนไม่ให้ระบายออกนอกตัวตู้ ส่วนที่ให้ความร้อน เป็นไส้ขดลวดความร้อน (Heater) เมื่อปล่อยไฟเข้าจะทำให้ขดลวดร้อนและแผ่รังสีความร้อนออกมาเต็มที่ หากจะควบคุมอุณหภูมิของตู้อบต้องคอยปิด - เปิดไฟให้กับขดลวดความร้อนนี้ตลอดเวลา ส่วนควบคุมประกอบด้วยวงจรไมโครโปรเซสเซอร์ และวงจรขับไส้ขดลวดความร้อนเรียกว่า Solid State Relay ส่วนการตรวจจับอุณหภูมิ ซึ่งมีอยู่หลายแบบได้แก่ Thermocouple ขดลวดต้านทาน (RTD) และอุปกรณ์สารกึ่งตัวนำ เช่น ไอซี DS1820 เป็นต้น และส่วนแสดงผลเป็นส่วนที่ทำให้ทราบสถานะของการควบคุม โดยจะแสดงอุณหภูมิที่ปรับตั้ง อุณหภูมิจริงในตู้อบ ส่วนนี้ได้รวมปุ่ม สวิตช์ควบคุมปรับตั้งอุณหภูมิ และส่วนปิดเปิดการทำงานของตู้อบด้วย

สำหรับในใบงานนี้จะแบ่งส่วนการควบคุม และการแสดงผลนำมาศึกษาปฏิบัติก่อน เพื่อให้เข้าใจในด้านการเขียนโปรแกรมประยุกต์ใช้งาน LCD การปรับตั้งอุณหภูมิ และการแสดงผล

วงจรทดลองตาม การทดลองที่ 4.2



การทำงานของโปรแกรมคำสั่ง

เมื่อเปิดเครื่องอุณหภูมิจะถูกตั้งค่าเริ่มต้นไว้ 25 องศาเซลเซียส สวิตช์ S1 และ S2 ใช้ปรับค่าขึ้น-ลง ถ้าปรับค่าเกินกว่า 25 หลอด Heater จะดับ และปรับน้อยกว่า 25 หลอดจะติด ค่าอุณหภูมิจะแสดงที่จอ LCD การปรับแต่ละครั้งจะมีเสียงคลิกการกดปุ่มสวิตช์ อุณหภูมิต่ำสุด 20 องศา และสูงสุด 80 องศา

โปรแกรมคำสั่ง

```

DEFINE LCD_DREG PORTD
DEFINE LCD_DBIT 4
DEFINE LCD_RSREG PORTE
DEFINE LCD_RSBIT 2
DEFINE LCD_EREG PORTD
DEFINE LCD_EBIT 1
S1    VAR    PORTA.3
S2    VAR    PORTA.4
spk    var portb.5
heater var portb.4
temp   var byte
TRISA  = %111111
ADCON1 = 7
temp = 25
lcdout $fe,1,"Temp = "
lcdout $fe,$c0,dec temp,$fe,$c4,"C"
LOOP:  IF (S1 = 0) AND (S2 = 1) THEN
    pause 50
    temp = temp+1
    gosub click
    gosub display
    gosub action
    if temp = 80 then temp = 79
idle1: if (S1 = 0) AND (S2 = 1) THEN idle1
endif
        IF (S1 = 1) AND (S2 = 0) THEN
            pause 50
            if temp = 20 then temp=21
            temp = temp - 1
            gosub click
            gosub display
            gosub action
idle2: if (S1 = 1) AND (S2 = 0) THEN idle2
endif
        PAUSE 50
        GOTO LOOP
END
'----- End of Main Program -----
'
'----- Subroutine Start Here -----
click:
    freqout spk,5,2000
    return

display:
    lcdout $fe,$c0,dec temp
    return

action:
    if temp > 25 then
        low heater
    else
        high heater
    endif
    return
'
'----- End of Subroutine -----

```

บันทึกผลการทดลอง

.....

.....

.....

.....

แบบฝึกหัด

- 1. ถ้าต้องการให้อุณหภูมิต่ำสุด 0 องศา และสูงสุด 180 องศาจะต้องเขียนโปรแกรมอย่างไร
- 2. จงเขียนไฟล์ชาร์ตการทำงานของโปรแกรมที่ 1

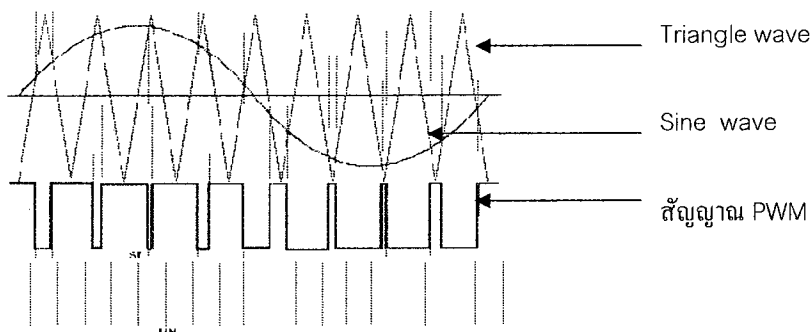
การทดลองที่ 5.1 การเขียนโปรแกรมกำเนิดสัญญาณ PWM

จุดประสงค์ เพื่อศึกษาเทคนิคการเขียนโปรแกรมเพื่อการสร้างสัญญาณ PWM

ทฤษฎีพื้นฐาน

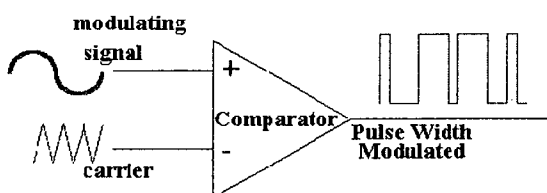
สัญญาณ PWM มาจากชื่อเต็มว่า Pulse Width Modulation เป็นสัญญาณที่เกิดจากการผสมกันระหว่างสัญญาณรูปสามเหลี่ยม (Triangle wave) กับระดับแรงเคลื่อนสัญญาณไฟดีซี หรือสัญญาณอื่น ๆ ที่มีระดับไฟดีซีเป็นตัวรองรับ ผลที่ได้จากการผสมสัญญาณดังกล่าวจะได้รูปคลื่นสัญญาณพัลส์ที่มี 2 สถานะ คือ

ON กับ OFF เมื่อนำสัญญาณพัลส์ที่ได้ไปขับอุปกรณ์กำลัง เช่น หลอดไฟ มอเตอร์ดีซี เราจะได้ผลการควบคุม คือเมื่อสัญญาณพัลส์มีสถานะ ON หลอดไฟจะติดสว่างเต็มที่ ถ้า OFF ไฟจะดับ หากสัญญาณมีความต่อเนื่องเช่นนี้ตลอดก็จะได้การติด - ดับ ของหลอดไฟต่อเนื่อง และหากการติดดับมีความถี่สูง ๆ จนสายตาไม่สามารถสังเกตการณ์ติด - ดับได้ เราก็จะสังเกตเห็นได้เพียงแสงที่หริ่ง หากเราทำให้ช่วงเวลา ON น้อยลง และในทางตรงกันข้าม หากเราทำให้ช่วงเวลา ON มากขึ้นเรื่อย ๆ ก็จะทำให้หลอดไฟสว่างขึ้นเรื่อย ๆ จนถึงสว่างเต็มที่เมื่อ ON ตลอด



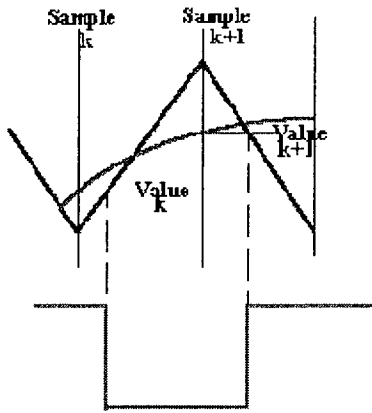
รูปที่ 1 แสดงการเกิดสัญญาณ PWM จากการผสมระหว่างสัญญาณรูปสามเหลี่ยม และรูปไซน์

ในการออกแบบวงจรผสมสัญญาณ จะใช้วงจรไอซี Linear comparator หรือ Analog comparator ตามวงจรรูปที่ 2

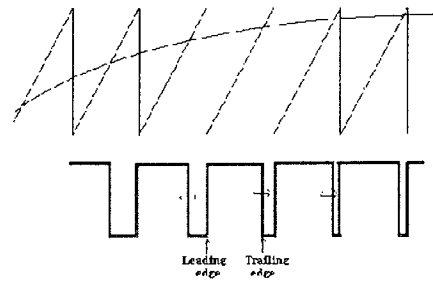


รูปที่ 2 แสดงวงจรการสร้างสัญญาณ PWM

สัญญาณรูปสามเหลี่ยมที่ใช้ในวงจร จะต้องมีความเป็นเชิงเส้น (Linearity) สูง จึงจะได้สัญญาณ PWM ที่มีความผิดเพี้ยนต่ำ สัญญาณรูปสามเหลี่ยมที่นำมาใช้จะมี 2 แบบ คือ Triangle wave และ Saw tooth wave

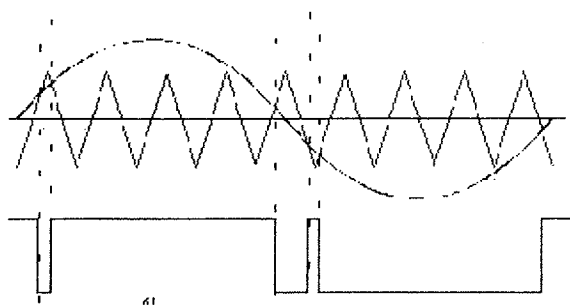


รูปที่ 3 แสดงการใช้คลื่น Triangle wave เป็นพาหะ tooth



รูปที่ 4 แสดงการใช้คลื่น Saw

wave เป็นพาหะ ในการผสมคลื่น สัญญาณคลื่นที่มาผสมจะต้องมีระดับความสูง (Amplitude) อย่างมากที่สุดไม่เกินยอดความสูงของรูปคลื่นสามเหลี่ยม ซึ่งเป็นรูปคลื่นพาหะ หากเกินไปกว่านั้นจะเกิดความผิดเพี้ยนขึ้นกับสัญญาณ PWM เรา เรียกปรากฏการณ์นี้ว่า over modulation



รูปที่ 5 แสดงความผิดเพี้ยนของสัญญาณ PWM เนื่องจากการเกิด over modulation

รายละเอียดของสัญญาณ PWM ประกอบด้วยส่วนต่าง ๆ ดังนี้ คือ

1. ความถี่ คือจำนวนรอบคลื่น ต่อวินาที (Cycle / second) ในการใช้งาน ความถี่จะต้องคงที่ไม่เปลี่ยนแปลง ความถี่ที่เหมาะสมในการใช้งานควบคุมอุปกรณ์กำลังต่าง ๆ คือ ตั้งแต่ 400 Hz - 10 KHz

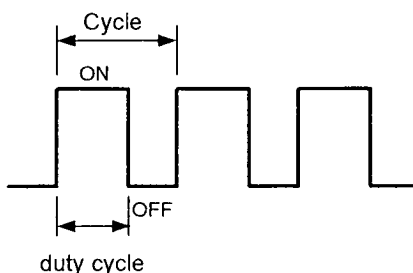
ส่วนความถี่ที่ใช้เป็นพาหะของสัญญาณเสียง (Audio) ควรจะสูงกว่า 50 KHz - 1 MHz

2. ดิวตี้ไซเคิล (Duty cycle) เป็นค่าคาบเวลาช่วง ON ของไซเคิล โดยคิดเป็นเปอร์เซ็นต์ของ Full- cycle

3. ค่าแรงเคลื่อนเอาพุท เป็นค่าเฉลี่ยระหว่างแรงเคลื่อนช่วงเวลา ON กับ OFF คิดใน 1 ไซเคิล

โดยใช้สมการ

$$V_{pwm} = \frac{\%duty\ cycle \times V_{power}}{100}$$



รูปที่ 6 แสดงองค์ประกอบของสัญญาณ PWM

ในโครงสร้างของไมโครโปรเซสเซอร์ ตระกูลไมโครชิพ เบอร์ PIC16F877 ที่ใช้ปฏิบัติในงานทดลองนี้ มีโมดูลสร้างสัญญาณ PWM แบบต่อเนื่องโดยฮาร์ดแวร์จำนวน 2 ชุด โดยใช้รายละเอียดในการปรับค่าดิวตี้ไซเคิล ตั้งแต่ 0 - 100 % จำนวน 256 ค่า โดยมีสัญญาณ PWM ช่องที่ 1 ออกที่ขา RC2 และช่องที่ 2 ออกที่ขา RC1 ส่วนเบอร์อื่น ๆ ให้ดูใน datasheet ให้ดูในขาที่มีฟังก์ชัน CCP1 CCP2 .. คำสั่งภาษา

Pic Basic Pro Compiler มีคำสั่งที่ทำให้ส่งสัญญาณ PWM ออกมา ได้แก่

HPWM Channel, Duty cycle, Frequency

Channel เป็นตำแหน่งขา I/O ที่ส่งสัญญาณ PWM ออกค่า 1 หมายถึง ขา RC2 และ 2 หมายถึงขา RC1 บางตัวอาจมีมากกว่านี้

Duty cycle เป็นค่าเปอร์เซ็นต์คาบเวลา ON ของไซเคิล มีค่าระหว่าง 0 - 255 สามารถกำหนดแทนด้วยตัวแปรขนาด 1 byte ได้

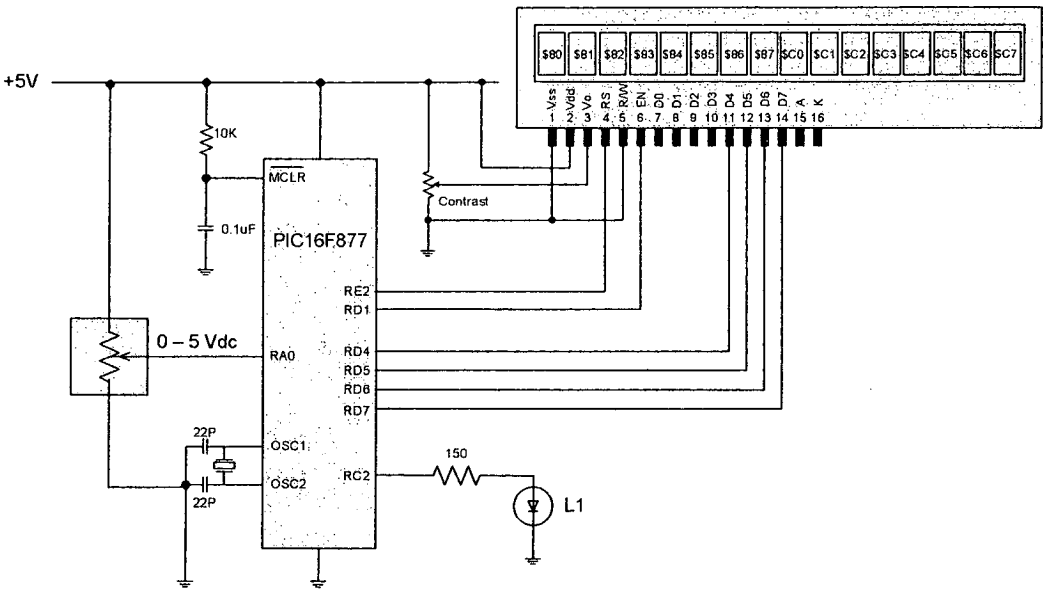
Frequency เป็นค่าความถี่พาหะ มีหน่วยเป็น Hz ความถี่สูงสุดที่ MCU สร้างได้คือ 32767 Hz ส่วนความถี่ต่ำสุด ขึ้นอยู่กับเบอร์ของ MCU และ ค่า OSC. ที่ใช้ ซึ่งเป็นตามตารางตัวอย่างต่อไปนี้

OSC.	14-bit core and 18-bit core	17Cxxx
4 MHz	245 Hz	3907 Hz
10 MHz	611 Hz	9766 Hz
12 MHz	733 Hz	11719 Hz
20 MHz	1221 Hz	19531 Hz
24 MHz	1465 Hz	23438 Hz
33 MHz	2015 Hz	32227 Hz
40 MHz	2442 Hz	ไม่ได้กำหนด

การทำงานของคำสั่ง จะทำตลอดเวลาโดยการสั่งเพียงครั้งเดียว และจะทำไปตามลำพังขณะที่โปรแกรมกำลังทำคำสั่งอื่นอยู่ เราจึงเรียกว่า Hardware Pulse Width Modulation หรือ HPWM ซึ่งจะมีข้อดี คือ จะได้สัญญาณ PWM อย่างต่อเนื่อง โดยไม่ต้องเขียนโปรแกรมให้วนย้อนกลับสร้างสัญญาณ PWM อยู่ตลอดเวลาสำหรับ MCU บางรุ่น ได้แก่ PIC18F452 สามารถสับเปลี่ยนขา I/O ที่ส่งสัญญาณ PWM ได้ โดยการใช้คำสั่งการนิยามขา I/O ดังตัวอย่างต่อไปนี้

```
DEFINE CCP1_REG PORTC
DEFINE CCP1_BIT 2
DEFINE CCP2_REG PORTC
DEFINE CCP2_BIT 1
```

วงจรทดลองตาม การทดลองที่ 5.1



การทำงานของโปรแกรม

ตามวงจรทดลอง เป็นการศึกษาการเขียนโปรแกรมผลิตสัญญาณ PWM เพื่อนำไปขับหลอด L1 ให้สว่างมาก น้อยเป็นไปตามค่าดิวิตีไซเกิล ค่ามาก น้อยของดิวิตีไซเกิลถูกควบคุมโดยค่าตัวแปรที่รับจากการแปลง A/D ที่ Channel 0 ดังนั้นการปรับแรงเคลื่อน A/D จาก 0 – 5V ก็จะทำให้หลอด L1 มีด และสว่างตามไปด้วย ค่าดิวิตีไซเกิล จะถูกนำไปแสดงที่จอ LCD ด้วย

โปรแกรมคำสั่ง

```

DEFINE LCD_DREG PORTD
DEFINE LCD_DBIT 4
DEFINE LCD_RSREG PORTE
DEFINE LCD_RSBIT 2
DEFINE LCD_EREG PORTD
DEFINE LCD_EBIT 1
temp var byte
temp1 var byte
    TRISA = %111111
    ADCON1 = 0

'---- main program ----
    adcin 0,temp
    adcon1 = 7
    lcdout $fe,1,"Temp = "
    lcdout $fe,$c0,dec temp,$fe,$c5,"C"
,

LOOP:
    gosub get_temp
    if temp <> temp1 then
        gosub display
        gosub send_pwm
        temp1 = temp
    endif
    PAUSE 50
    GOTO LOOP

END

'--- End of Main Program ---
,

'-- Subroutine Start Here --
,

get_temp:
    adcon1 = 0
    adcin 0,temp
    pause 50
    return
,

display:
    adcon1 = 7
    lcdout $fe,$c0," "
    lcdout $fe,$c0,dec temp
    return
,

send_pwm:
    HPWM 1,temp,2000
    return
,

'----- End of Subroutine -----

```

บันทึกผลการทดลอง

.....

.....

.....

.....

แบบฝึกหัด

- 1. คำสั่ง HPWM *Channel, Duty cycle, Frequency* ถ้ามีการปรับค่าตัวใดตัวหนึ่ง 50% จงเปลี่ยนความถี่ให้ลดลงเหลือ 1000 Hz สังเกตการณ์เปลี่ยนแปลงของ LED มีผลต่างกันหรือไม่อย่างไร

การทดลองที่ 5.2 การเขียนโปรแกรมควบคุมมอเตอร์ดีซีแบบหมุนต่อเนื่อง

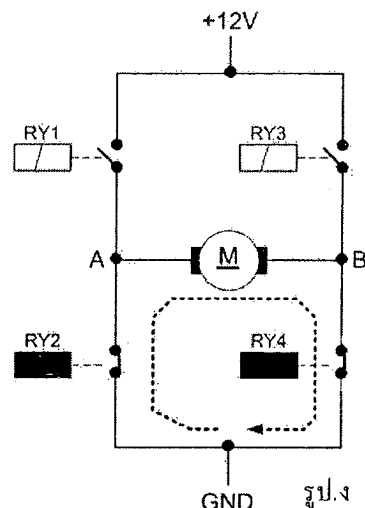
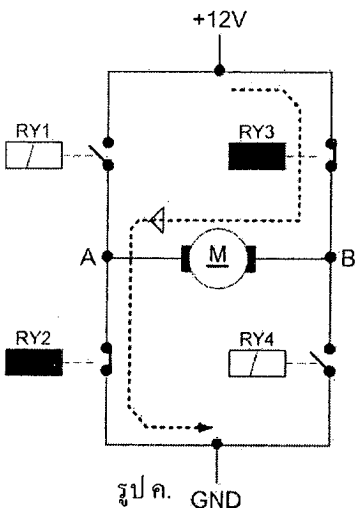
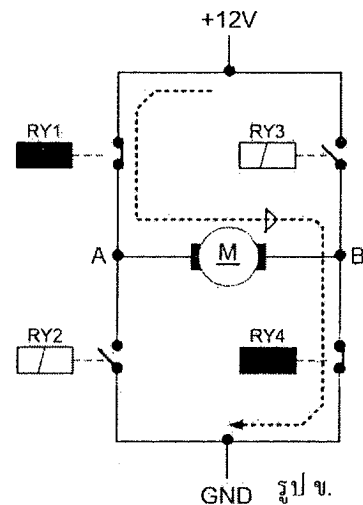
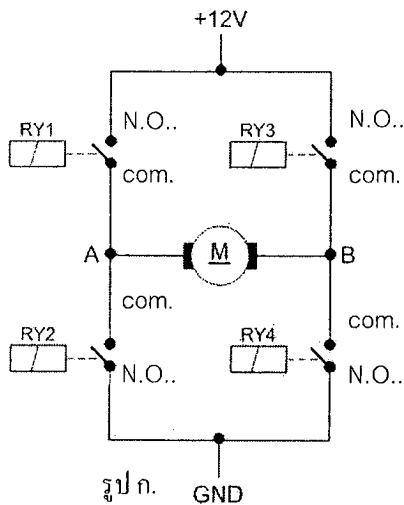
จุดประสงค์ เพื่อออกแบบระบบการควบคุมมอเตอร์แบบใช้ไฟตรง (DC Motor) ในลักษณะการหมุน –

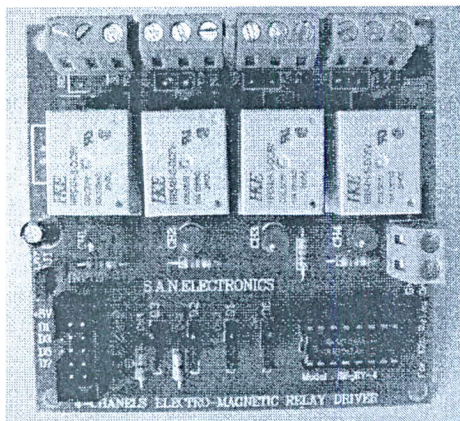
หยุดหมุน และการกลับทิศทางการหมุน แบบการทำงานต่อเนื่อง

ทฤษฎีพื้นฐาน ในการควบคุมมอเตอร์ไฟตรง (DC Motor) มีการควบคุมอยู่ 3 ลักษณะคือ

1. หมุน - หยุดหมุน (ON - OFF Control)
2. ทิศทางการหมุน (Direction Control)
3. ความเร็วการหมุน (Speed Control)

อุปกรณ์ที่ใช้ควบคุมที่นิยมใช้ส่วนมากได้แก่ หน้าคอนแทกซ์รีเลย์ต่อเป็นวงจรในลักษณะ H – Bridge ที่สามารถควบคุมการหมุน – หยุดหมุน พร้อมทั้งการกลับทิศทางการหมุนได้ แต่การใช้หน้าคอนแทกซ์รีเลย์จะไม่สามารถควบคุมความเร็วของการหมุนได้





รูปที่ 1 แสดงบอร์ดรีเลย์แบบ 4 ช่องที่ใช้ทดลอง ตามใบงาน

วงจรควบคุมประกอบด้วยรีเลย์จำนวน 4 ตัว มีหน้าคอนแทคต่อวงจรตามรูป ก. หากเราสามารถควบคุมรีเลย์ให้ทำงานได้ เราสามารถควบคุมการทำงานของมอเตอร์ได้ 4 โหมด คือ

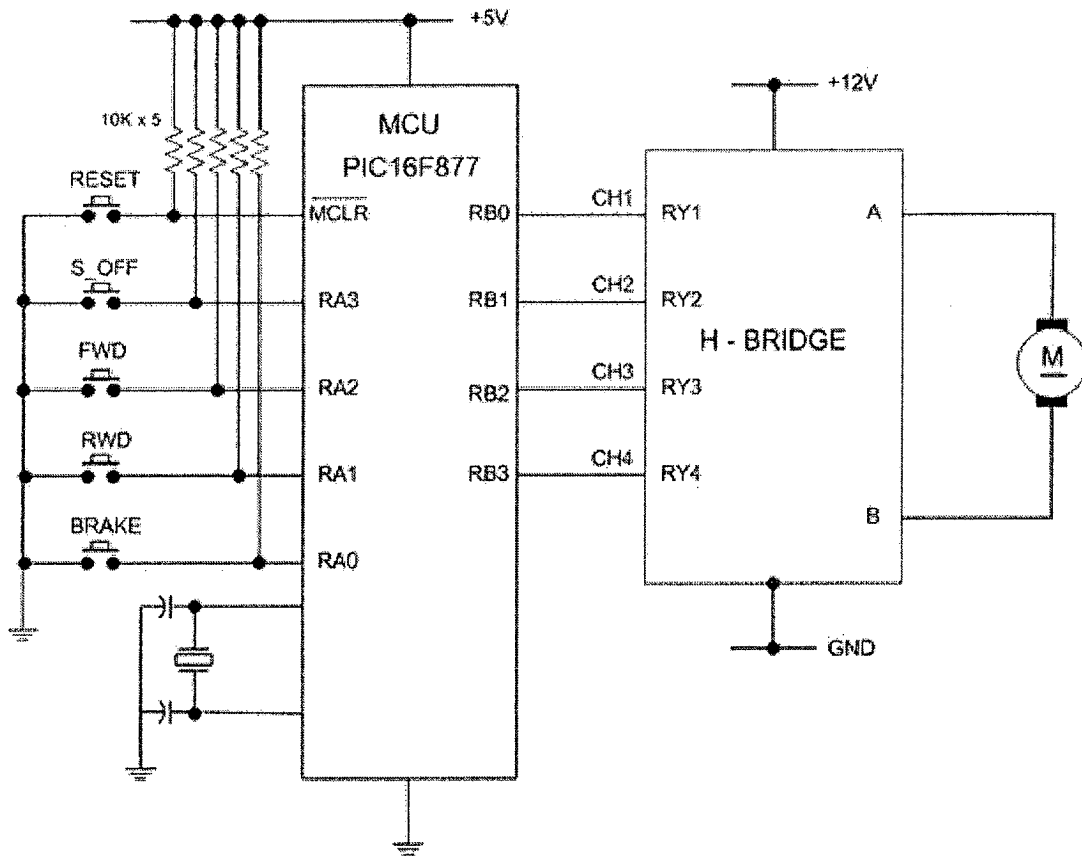
1. Free Run Stop คือการสั่งให้น้ำคอนแทคของรีเลย์เปิดทั้ง 4 ตัว ตามวงจรรูป ก. ทำให้มอเตอร์ไม่มีไฟ มอเตอร์จะหยุดไปเองเมื่อหมดแรงเฉื่อย

2. Forward Drive คือการสั่งให้รีเลย์ RY1 กับ RY4 ทำงานแค่ 2 ตัว ตามวงจรรูป ข. ทำให้มีกระแสไฟไหลเข้าขั้วมอเตอร์ในทิศทางจาก A ไป B เป็นทิศทางสมมติให้มอเตอร์หมุนในทิศทางตามเข็มนาฬิกา

3. Reward Drive คือการสั่งให้รีเลย์ RY2 กับ RY3 ทำงานแค่ 2 ตัว ตามวงจรรูป ค. ทำให้มีกระแสไฟไหลเข้าขั้วมอเตอร์ในทิศทางจาก B ไป A เป็นทิศทางสมมติให้มอเตอร์หมุนในทิศทางทวนเข็มนาฬิกา

4. Fast Stop หรือ Brake คือการสั่งให้รีเลย์ RY2 กับ RY4 หรือ RY1 กับ RY3 ทำงานคู่ใดคู่หนึ่งเพียง 2 ตัวตามวงจรรูป ง. ในลักษณะนี้ ทำให้ขั้วมอเตอร์ลัดวงจรกัน ในขณะที่มอเตอร์กำลังหมุนด้วยแรงเฉื่อยขดลวดทุ่นอาร์มเมเจอร์ที่ตัดสนามแม่เหล็กจะมีกระแสเหนี่ยวนำทำให้มีสนามแม่เหล็กเกิดขึ้นในทิศทางต้านกับการหมุนของขดลวดทุ่นอาร์มเมเจอร์ ทำให้แกนมอเตอร์หยุดหมุนอย่างรวดเร็ว

หากมีรีเลย์ทำงานนอกเหนือไปจากนี้จะมีผลเกิดขึ้น 2 อย่างคือ มอเตอร์ไม่หมุน กับไฟ
ลัดวงจรทำให้หน้าคอนแทกรีเลย์พัง ดังนั้นในการเขียนโปรแกรมควบคุมรีเลย์ทั้ง 4 ตัว
จำเป็นต้องมีการป้องกันผลเสียหายที่เกิดจากการทำงานผิดพลาดนี้ด้วย



หมายเหตุ วงจร H – Bridge ใช้บอร์ดรีเลย์ 4 ช่อง Model SE-RYI-4 โดยการใช้เพียง 4
ช่องโดยเดินสาย ต่อวงจรตามรูป ก. ใช้สายแพต่อเชื่อมกับ PortB ของบอร์ดทดลอง Model
SE-PIC877 จะได้วงจรตามรูปข้างบน

ขั้นตอนการปฏิบัติ

1. ต่อวงจรทดลองตามรูปที่ 2 โดยต่อสายเข้ารีเลย์และมอเตอร์ตามรูป ก.
2. ต่อไฟดีซี 12 โวลต์ เข้ากับวงจร H – Bridge ที่บอร์ดรีเลย์ แล้วพิมพ์โปรแกรม

โปรแกรมคำสั่ง

```
s_off  var    portb.3
fwd    var    portb.2
rwd    var    portb.1
brake  var    portb.0
trisa  =      %1111111
trisb  =      %000000000
adcon1=      7
portb  =      0
```

start:

```
if (s_off = 0) and (fwd = 1) and (rwd = 1) and (brake = 1) then portb =
%00000
```

```
if (s_off = 1) and (fwd = 0) and (rwd = 1) and (brake = 1) then portb =
%1001
```

```
if (s_off = 1) and (fwd = 1) and (rwd = 0) and (brake = 1) then portb =
%0110
```

```
if (s_off = 1) and (fwd = 1) and (rwd = 1) and (brake = 0) then portb =
%1010
```

```
pause 50
```

```
goto start
```

```
end
```

‘----- End of Program -----’

บันทึกผลการทดลอง

.....

.....

.....

.....

แบบฝึกหัด

- 1. วงจรและโปรแกรมการทดลองสามารถนำไปประยุกต์ใช้งานลักษณะใดได้บ้างยกตัวอย่าง
และอธิบายการทำงาน

การทดลองที่ 5.3 การเขียนโปรแกรมควบคุมมอเตอร์ดีซีแบบหมุนตามจังหวะที่กดสวิตช์ (Jogging control)

จุดประสงค์ เพื่อออกแบบระบบการควบคุมมอเตอร์แบบใช้ไฟตรง (DC Motor) ในลักษณะการหมุนหยุดหมุน และการกลับทิศทางการหมุน แบบทำเมื่อกดสวิตช์ควบคุม (Jogging Control)

ทฤษฎีพื้นฐาน

การควบคุมมอเตอร์ไฟตรง (DC Motor) แบบ Jogging ใช้วงจรและทฤษฎีการควบคุมเช่นเดียวกันกับ Experiment ที่ 24 แต่จะแตกต่างกันที่การเขียนโปรแกรมควบคุมมอเตอร์ทำงาน คือ การควบคุมแบบ Jogging มอเตอร์จะทำงานก็ต่อเมื่อมีการกดสวิตช์ค้างไว้เท่านั้น หากปล่อยมือมอเตอร์จะหยุดทันที ดังนั้นวงจรสวิตช์ควบคุมจะเหลือเพียง 2 ปุ่มคือ ปุ่มกดให้หมุนตามเข็มนาฬิกา fwd และ หมุนทวนเข็มนาฬิกา rwd

วงจรทดลอง ใช้วงจรทดลองตาม การทดลองที่ 5.2

โปรแกรมคำสั่ง

```
fwd    var    portb.2
rwd    var    portb.1
trisa  =      %111111
trisb  =      %00000000
adcon1 =      7
portb  =      0
```

start:

```
if (fwd = 1) and (rwd = 1) then portb = %1010
if (fwd = 0) and (rwd = 1) then portb = %1001
if (fwd = 1) and (rwd = 0) then portb = %0110
pause 50
goto start
end
```

‘----- End of Program -----’

บันทึกผลการทดลอง

.....

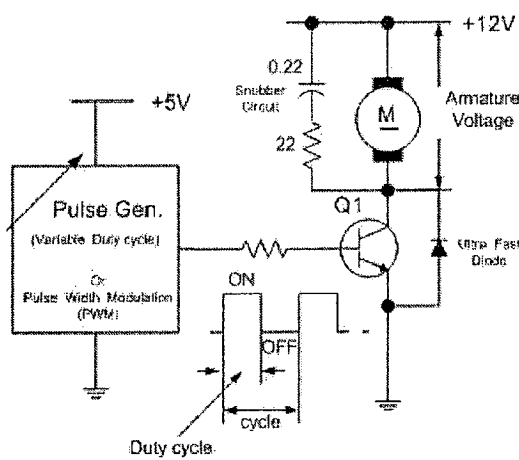
.....

.....

.....

แบบฝึกหัด

- 1. วจรและโปรแกรมการทดลองสามารถนำไปประยุกต์ใช้งานลักษณะใดได้บ้างยกตัวอย่าง
และอธิบายการทำงาน



รูปที่ 2 แสดงหลักการควบคุมความเร็วมอเตอร์ดีซีแบบสวิตชิง หรือ PWM

วงจรในรูปที่ 2 ประกอบด้วยตัวกำเนิดสัญญาณพัลส์ส่งไปควบคุมการทำงานของทรานซิสเตอร์ในลักษณะ ON และ OFF ทำให้มีกระแสไหลผ่านมอเตอร์หมุนเป็นช่วง ๆ ที่เป็นพัลส์บวก เราเรียกว่า “Duty cycle” โดยที่ตัวผลิตพัลส์ (Pulse Generator) จะต้องรักษาช่วงเวลาไซเคิลหรือความถี่ให้คงที่อยู่ที่ค่าใดค่าหนึ่ง โดยปกติจะเลือกค่าคงที่อยู่ระหว่าง 400 Hz - 20 KHz สำหรับงานควบคุมมอเตอร์ ดังนั้นในทุก ๆ ไซเคิลหากเราปรับเปลี่ยนค่าพัลส์บวก หรือค่า duty cycle ก็จะมีผลทำให้ค่าเฉลี่ยแรงเคลื่อนเข้าพุทของทรานซิสเตอร์ที่ไปขับมอเตอร์เปลี่ยนแปลงตามไปด้วย ทำให้ความเร็วของมอเตอร์เปลี่ยนแปลงตามไปด้วย ตามสมการที่ 1 ค่า

$$\% \text{ Duty cycle} = \frac{\text{Pulse width}}{\text{Cycle}} \times 100 \quad \text{-----} \quad 1$$

duty cycle จะคิดเป็นเปอร์เซ็นต์ ของไซเคิล และค่าเฉลี่ยของแรงเคลื่อนเข้าพุทที่ขับโหลดคำนวณได้จากสมการที่ 2 ดังต่อไปนี้

$$V_{\text{output}} = \% \text{Duty cycle} \times V_{\text{supply}} \quad \text{-----} \quad 2$$

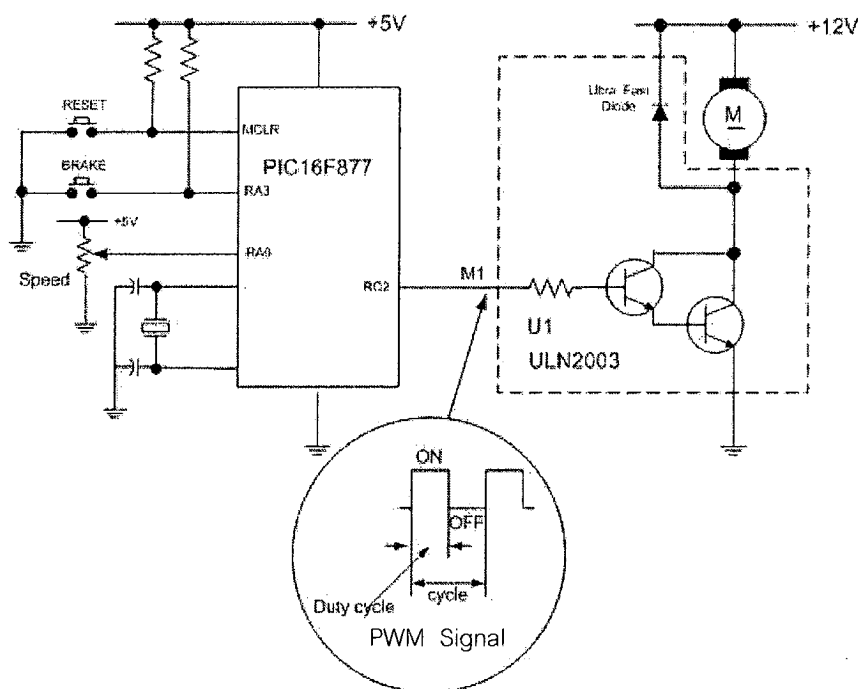
ตามสมการที่ 2 พบว่า แรงเคลื่อนเข้าพุทที่จะออกไปขับมอเตอร์ให้หมุนที่ความเร็วเปลี่ยนไปนั้น จะแปรผันโดยตรงกับค่า %Duty- cycle หรือความกว้างของพัลส์ หรือเราเรียกสัญญาณควบคุมแบบนี้ว่า PWM (PWM : Pulse Width Modulation) หมายความว่า หากเราสามารถเขียนโปรแกรมผลิตสัญญาณ PWM ที่เปลี่ยนค่า Pulse Width ให้ออกมาควบคุมวงจรขับมอเตอร์ได้ เราก็สามารถควบคุมความเร็วของมอเตอร์ได้

การออกแบบเชื่อมต่อการควบคุมความเร็วด้วยไมโครโปรเซสเซอร์ จะต้องเลือกไมโครคอนโทรลเลอร์ ชนิดที่มีขีดความสามารถในการสร้างสัญญาณควบคุมแบบ PWM โดย

ฮาร์ดแวร์ที่ผู้ผลิตสร้างมาให้ เช่น ไมโครโปรเซสเซอร์ เบอร์ PIC16F877 ของบริษัทไมโครชิพ จะมีโมดูลที่สามารถสร้างสัญญาณควบคุมแบบ PWM โดยฮาร์ดแวร์ได้ถึง 2 ชุดที่สามารถสั่งทำงานตามลำพังได้ในขณะที่กำลังประมวลผลคำสั่งอื่นในโปรแกรมอยู่ ทำให้การผลิตสัญญาณควบคุมเป็นไปอย่างต่อเนื่องไม่ขาดช่วง โดยได้กำหนดขา I/O ไว้ 2 ขา คือ RC2 สำหรับ HPWM ช่อง 1 และ ขา RC1 สำหรับ HPWM ช่อง 2

รูปแบบการใช้คำสั่ง คือ HPWM channel, duty cycle, frequency รายละเอียดให้ศึกษาได้จากคู่มือการใช้โปรแกรม PIC BASIC PRO ทดลองตามใบงาน จะใช้ขา RC2 ซึ่งผลิตสัญญาณ PWM channel 1 และค่า duty cycle จะใช้ค่า A/D ที่รับมาจาก RA0 นำมาใช้ปรับเปลี่ยนค่า duty cycle เพื่อควบคุมความเร็วมอเตอร์

วงจรทดลอง



รูปที่ 3 แสดงวงจรการควบคุมความเร็วมอเตอร์ดีซีด้วยไมโครโปรเซสเซอร์

โปรแกรมคำสั่ง

```
DUTY  VAR BYTE
SPEED VAR BYTE
BRAKE VAR PORTA.3
ADCON1=14
TRISA =%111111
TRISC =%00000000

START: IF (BRAKE=0) THEN
        HPWM 1,0,1000
    ELSE
        ADCIN 0,SPEED
        HPWM 1, speed, 1000
    ENDIF
    PAUSE 50
    GOTO  START
END
```

บันทึกผลการทดลอง

.....

.....

.....

.....

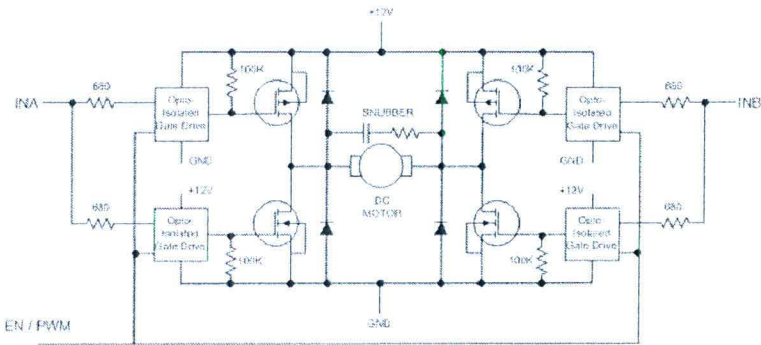
แบบฝึกหัด

- 1. วจรและโปรแกรมการทดลองสามารถนำไปประยุกต์ใช้งานลักษณะใดได้บ้างยกตัวอย่างและอธิบายการทำงาน

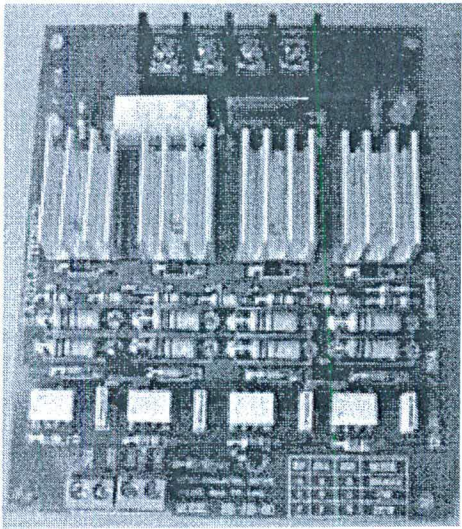
การทดลองที่ 5.5 การเขียนโปรแกรมควบคุมทั้งความเร็วและทิศทางของมอเตอร์ดีซี

จุดประสงค์ เพื่อออกแบบระบบการควบคุมมอเตอร์ดีซี (DC Motor) ในลักษณะควบคุมความเร็ว (Speed Control) พร้อมทั้งควบคุมทิศทางการหมุน (Direction Control)

ทฤษฎีพื้นฐาน ในการควบคุมทั้งความเร็ว และทิศทางการหมุนของมอเตอร์ดีซี (DC Motor) จำเป็น ต้องใช้วงจรอิเล็กทรอนิกส์ H-Bridge เพื่อให้สามารถเป็นสวิตช์ตัดต่อกระแสไฟควบคุมมอเตอร์แทนหน้าคอนแทกที่เร็วพอที่จะใช้สัญญาณ PWM ควบคุมความเร็วไปพร้อมกับการควบคุมการกลับทิศทางการหมุนของมอเตอร์



รูปที่ 1 วงจรแสดงการทำงานของ MOSFET H – Bridge ขับมอเตอร์ดีซี

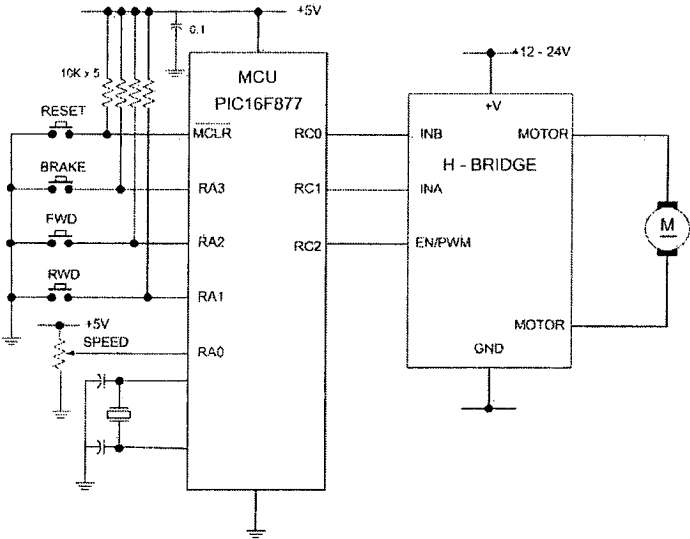


รูปที่ 2 บอร์ด MOSFET H-Bridge ขนาด 30 A ขับมอเตอร์ดีซี

การทำงานของวงจร ประกอบด้วย POWER MOSFET จำนวน 4 ตัว ทำหน้าที่เป็นหน้าคอนแทคสวิตช์ จะถูกควบคุมโดยแรงดันจากวงจร Opto – Isolated Gate Drive อินพุตของวงจร Opto – Isolated Gate Drive แต่ละข้าง จะต่อรวมกันเป็น INA และ INB POWER MOSFET แต่ละข้างของ H – Bridge จะทำงานแบบ Complementary โดยที่ไม่มีโอกาสจะทำงานพร้อมกันได้เลย ได้ออกแบบให้มีขาสำหรับป้อนสัญญาณ EN / PWM เพื่อสามารถกำหนดช่วงเวลาในการเปิด – ปิด สวิตช์ของ POWER MOSFET แต่ละชุดได้ เพื่อใช้ในการควบคุมความเร็ว การทำงานของ H-Bridge เป็นไปตามตาราง รูปที่ 3

EN	INB	INA	
0	x	x	Brake
1	0	1	Forward
1	1	0	Reward
1	1	1	Brake
1	0	0	Brake

รูปที่ 3 แสดงการทำงานของวงจร MOSFET H – Bridge รุ่น SE – HB – 60
วงจรทดลอง



รูปที่ 3 แสดงการต่อวงจรควบคุม MOSFET H-Bridge ด้วยไมโครโปรเซสเซอร์

โปรแกรมคำสั่ง

```

DUTY   VAR BYTE
SPEED  VAR BYTE
BRAKE  VAR PORTA.3
FWD    VAR PORTA.2
RWD    VAR PORTA.1
ADCON1 = 14
TRISA  = %111111
TRISC  = %00000000

```

```

START: IF (BRAKE=0) AND (FWD=1) AND (RWD=1) THEN
        HPWM 1,0,1000
    ELSE
        IF (BRAKE=1) AND (FWD=0) AND (RWD=1) THEN LOW PORTC.0:HIGH PORTC.1
        IF (BRAKE=1) AND (FWD=1) AND (RWD=0) THEN HIGH PORTC.0:LOW PORTC.1
        ADCIN 0, SPEED
        DUTY = SPEED
        HPWM 1, DUTY, 1000
    ENDIF
    PAUSE 50
    GOTO START
END

```

บันทึกผลการทดลอง

.....

.....

.....

.....

แบบฝึกหัด

1. วงจรการทดลองสามารถนำไปใช้ในการจัดทำรถไฟฟ้าได้หรือไม่อย่างไร

ภาคผนวก ค

แบบทดสอบรายวิชาการประยุกต์ใช้งานไมโครโปรเซสเซอร์

แบบทดสอบรายวิชาการประยุกต์ใช้งานไมโครโปรเซสเซอร์

จงตอบคำถามต่อไปนี้ลงในกระดาษคำถาม ข้อสอบมี 40 ข้อ 40 คะแนน

1. ข้อใดกล่าวไม่ถูกต้องเกี่ยวกับไมโครโปรเซสเซอร์ PIC16F877
 - ก. ชนิดตัวถังแบบ PDIP มีจำนวนขาทั้งหมด 40 ขา
 - ข. สามารถดาวน์โหลดโปรแกรมด้วยเทคนิค ICSP (In circuit Serial Programming) ผ่านขา PGD และ PGC
 - ค. สามารถออกแบบให้มีความเร็วสูงสุดในการประมวลผล 1 คำสั่งใช้เวลา 0.05 ไมโครวินาที
 - ง. สามารถเชื่อมต่อแหล่งสัญญาณแอนาลอกได้จำนวน 8 ช่อง และประมวลผลพร้อมกันได้

2. ไมโครโปรเซสเซอร์ PIC 16F877 มีจำนวนขาทั้งสิ้น 40 ขา 5 พอร์ต คือ A, B, C, D และ E พอร์ต ข้อใดคือความแตกต่างของพอร์ต A เมื่อเปรียบเทียบกับพอร์ตอื่นๆ
 - ก. พอร์ต A มีจำนวน 5 บิต ถือว่ามีจำนวนน้อยที่สุด
 - ข. ทุกขาของพอร์ต A สามารถรับสัญญาณแอนาลอกได้
 - ค. พอร์ต A อยู่ใกล้กับขา 40 มากที่สุด
 - ง. พอร์ต A สามารถเชื่อมต่อการสื่อสารแบบขนานได้พร้อมกัน

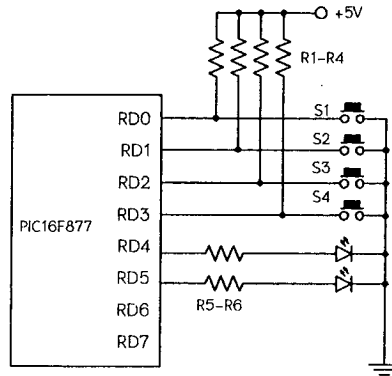
3. รีจิสเตอร์ TRIS มีหน้าที่อะไร
 - ก. สำหรับกำหนดหน้าที่ของพอร์ตทำหน้าที่ ทางเข้า หรือ ทางออกของสัญญาณ
 - ข. สำหรับเก็บข้อมูลชั่วคราวเมื่อมีการประมวลผล
 - ค. สำหรับกำหนดค่าหน่วยความจำค่าแรก ภายในไมโครคอนโทรลเลอร์
 - ง. สำหรับกำหนดความเร็วในการประมวลผลคำสั่งไมโครคอนโทรลเลอร์

4. การออกแบบวงจรการใช้งานไมโครโปรเซสเซอร์ PIC16F877 จะต้องสร้างสัญญาณนาฬิกาให้กับไมโครคอนโทรลเลอร์ที่ขา OSC 1 และ OSC 2 ข้อใดคือขา OSC1 และ OSC2
 - ก. ขาที่ 10 และ 11 ข. ขาที่ 11 และ 12 ค. ขาที่ 12 และ 13 ง. ขาที่ 13 และ 14

5. ขาสัญญาณใดสำหรับการใช้เป็นแหล่งจ่ายไฟฟ้าให้กับไมโครโปรเซสเซอร์ PIC16F877
 - ก. ขา Vdd สำหรับไฟ +5V และ ขา Vss สำหรับไฟ 0 V
 - ข. ขา Vdd สำหรับไฟ 0V และ ขา Vss สำหรับไฟ +5 V
 - ค. ขา Vcc สำหรับไฟ +5V และ ขา Vss สำหรับไฟ 0 V
 - ง. ขา Vcc สำหรับไฟ +5V และ ขา GND สำหรับไฟ 0 V

6. การใช้งานไมโครโปรเซสเซอร์ PIC16F877 สามารถเชื่อมต่ออุปกรณ์ได้จำนวนกี่บิต
 - ก. 30 บิต
 - ข. 31 บิต
 - ค. 32 บิต
 - ง. 33 บิต
7. สัญญาณทางออกจากพอร์ต A ที่ลอจิก 1 มีค่าแรงดัน 5V สามารถจ่ายกระแสไฟฟ้าได้สูงสุดเท่าใด
 - ก. 2.5 mA
 - ข. 10 mA
 - ค. 25 mA
 - ง. 100 mA
8. PIC 16F877 มีชุดคำสั่งภาษาแอสเซมบลี 1 คำสั่งมีจำนวนกี่บิต
 - ก. 8 บิต
 - ข. 10 บิต
 - ค. 14 บิต
 - ง. 16 บิต
9. ขาสัญญาณที่เชื่อมต่อแบบ ICD กับอุปกรณ์สำหรับโหลดโปรแกรมจากเครื่องคอมพิวเตอร์ ขาสัญญาณ Vpp มีหน้าที่ใดเมื่อมีการโหลดโปรแกรม
 - ก. มีหน้าที่รับข้อมูลโปรแกรมแบบอนุกรม
 - ข. มีหน้าที่รับแรงดันไฟฟ้าสำหรับการโปรแกรม
 - ค. มีหน้าที่รับสัญญาณนาฬิกาจากคอมพิวเตอร์
 - ง. มีหน้าที่สร้างสัญญาณไฟฟ้าส่งไปยังคอมพิวเตอร์
10. ขาสัญญาณ AUX ที่พอร์ตเชื่อมต่อ ICD มีหน้าที่อย่างไร
 - ก. รับสัญญาณนาฬิกาจากไมโครคอนโทรลเลอร์
 - ข. รับสัญญาณข้อมูลการโปรแกรมจากคอมพิวเตอร์
 - ค. สร้างสัญญาณควบคุมให้คอมพิวเตอร์และไมโครคอนโทรลเลอร์มีความสัมพันธ์กัน
 - ง. ไม่มีหน้าที่ใดๆ
11. รีจิสเตอร์ใดมีหน้าที่สำหรับ กำหนดขาสัญญาณ AN0-AN7 สามารถรับสัญญาณแบบแอนาลอกได้
 - ก. ADDCON0
 - ข. ADDCON1
 - ค. RPO
 - ง. RP1
12. คุณสมบัติพิเศษของ พอร์ต B คือข้อใด
 - ก. สามารถเชื่อมต่อสวิตช์แบบเมตริกซ์ได้
 - ข. สามารถควบคุมกระแสไฟฟ้าให้เพิ่มมากกว่าปกติได้
 - ค. มีจำนวน 2 บิตสำหรับขาโหลดข้อมูลสำหรับโปรแกรม
 - ง. ถูกทุกข้อ
13. วงจรใด ใช้สำหรับการควบคุมทิศทางการหมุน และ ความเร็วมอเตอร์ไฟฟ้ากระแสตรง
 - ก. De bounce
 - ข. Phase Control
 - ค. Switching
 - ง. H Bridge
14. สัญญาณ PWM จากไมโครคอนโทรลเลอร์ PIC16F877 ขาที่มีชื่อใด
 - ก. CCP
 - ข. PWM
 - ค. TMR
 - ง. WDT

จากวงจรตอบคำถามข้อ 15-16



15. จากวงจรค่าความต้านทาน R1 ต่อร่วมกับ S1 เราเรียกรวมการต่อวงจรลักษณะนี้ว่าอย่างไร
 ก. วงจร Hi-Push UP ข. วงจร Low-Push UP ค. วงจร High-Pull UP ง. วงจร Low-Pull UP
16. จากวงจรค่าความต้านทาน R5 และ R6 ควรมีค่าเท่าใด เมื่อ LED สว่างที่กระแสไฟฟ้า 10 มิลลิแอมป์ โดยที่ LED ไม่เสียหายเมื่อได้รับสัญญาณ ลอจิก 1 จากไมโครคอนโทรลเลอร์
 ก. 1,000 โอห์ม ข. 560 โอห์ม ค. 300 โอห์ม ง. 100 โอห์ม
17. คำสั่งใดภาษา Basic ใช้สำหรับการเปลี่ยนค่าสถานะของพอร์ตทางออกเป็นสถานะตรงกันข้าม
 ก. TOGGLE ข. PAUSE ค. SWITCH ง. IDIE
18. ทำไมต้องติดตั้งโปรแกรม MPLAB เมื่อเขียนโปรแกรมผ่าน Pic Basic Pro
 ก. เพราะต้องใช้โปรแกรม PBP สำหรับแปลงภาษา BAS เป็น ไฟล์ชนิด ASM
 ข. เพราะต้องใช้โปรแกรม PBP สำหรับแปลงภาษา ASM เป็น ไฟล์ชนิด HEX
 ค. เพราะต้องใช้โปรแกรม MPASM สำหรับแปลงภาษา BAS เป็น ไฟล์ชนิด ASM
 ง. เพราะต้องใช้โปรแกรม MPASM สำหรับแปลงภาษา ASM เป็น ไฟล์ชนิด HEX
19. การเชื่อมต่อ USB เข้ากับไมโครคอนโทรลเลอร์ จะต้องใช้อุปกรณ์ช่วยสำหรับการแปลงอินเตอร์เฟซยกเว้นข้อใด
 ก. PICKIT 3 ข. ICD2 ค. HUB ง. REAL ICE
20. PIC 16F877 สามารถใช้สัญญาณนาฬิกาได้สูงสุดเท่าใดในการสร้างจังหวะการทำงาน
 ก. 10 MHz ข. 12 MHz ค. 20 MHz ง. 30 MHz
21. รหัสใดที่ใช้สำหรับการแสดงผลตัวอักษรบนจอ LCD
 ก. รหัส BCD8421 ข. รหัส ASCII ค. รหัส LCD ง. รหัส PCC

22. ไฟล์นามสกุลใดไม่เกิดขึ้นหลังจากการ Compile โปรแกรมที่เกี่ยวข้องเมื่อเขียนด้วยภาษา Basic Pic Pro
 ก. CCS ข. OBJ ค. HEX ง. ASM
23. ความสามารถในการแปลงสัญญาณแอนาลอกเป็นสัญญาณดิจิทัลของ PIC16F877 มีความละเอียดกี่บิต
 ก. 4 บิต ข. 8 บิต ค. 10 บิต ง. 12 บิต
24. แรงดันไฟฟ้าสูงที่สุดที่ PIC16F877 ทนได้มีค่าเท่าใด
 ก. 5 โวลต์ ข. 5.25 โวลต์ ค. 5.5 โวลต์ ง. 5.75 โวลต์
25. ข้อใดเป็นคำสั่งส่งสัญญาณ PWM ออกมาด้วยคำสั่ง Pic Basic Pro ให้ส่งสัญญาณ ช่องที่ 1 ความถี่ 1 KHz และเป็นมีสัญญาณช่วง ON ของไซเคิล 50%
 ก. HPWM 1,1000,50 ข. HPWM 1000,1,50 ค. HPWM 1,50,1000 ง. HPWM 1,1000,50
26. คำสั่ง `Lcdout $fe,$c0," "` มีความหมายว่าอย่างไร
 ก. เลื่อนเคอเซอร์ไปเริ่มต้นบรรทัดที่แรก
 ข. เลื่อนเคอเซอร์ไปเริ่มต้นบรรทัดที่สอง
 ค. เลื่อนเคอเซอร์ไปเริ่มต้นบรรทัดก่อนหน้าตำแหน่งเดิม
 ง. เลื่อนเคอเซอร์ไปเริ่มต้นบรรทัดถัดไปจากตำแหน่งเดิม
27. จอแสดงผล LCD มีขา A และ K มีหน้าที่อย่างไร
 ก. สำหรับจ่ายไฟเลี้ยงขา A จ่ายไฟบวก 5 V และ ขา K ลง GND เพื่อให้จอแสดงผลสว่างเห็นตัวอักษรได้ชัด
 ข. สำหรับจ่ายไฟเลี้ยงขา A จ่ายไฟบวก 12 V และ ขา K ลง GND เพื่อให้จอแสดงผลสว่างเห็นตัวอักษรได้ชัด
 ค. สำหรับจ่ายไฟเลี้ยงขา A จ่ายไฟบวก 5 V และ ขา K ลง GND เพื่อให้จอแสดงผลทำงานได้
 ง. สำหรับจ่ายไฟเลี้ยงขา A จ่ายไฟบวก 12 V และ ขา K ลง GND เพื่อให้จอแสดงผลทำงานได้
28. ข้อใดคือความแตกต่างของจอแสดงผล LCD ชนิด 16 ตัวอักษร 2 บรรทัด กับ ชนิด 16 ตัวอักษร 4 บรรทัด
 ก. เลขรหัสประจำตำแหน่งขนาด 2 บรรทัดมีตั้งแต่ \$80 ถึง \$8F และ \$C0 ถึง \$CF แต่ เลขรหัสประจำตำแหน่งขนาด 4 บรรทัดมีตั้งแต่ \$80 ถึง \$8F และ \$C0 ถึง \$CF และ \$94 ถึง \$A3 และ \$D4 ถึง \$E3
 ข. จำนวนบิตเชื่อมต่อชนิด 2 บรรทัดมีน้อยกว่าชนิด 4 บรรทัด 2 บิต
 ค. ชนิด 2 บรรทัดไม่สามารถเชื่อมต่อข้อมูลแบบ 4 บิตได้ แต่ชนิด 4 บรรทัดสามารถเชื่อมต่อได้ทั้งแบบ 4 บิต และ 8 บิต
 ง. มีชุดคำสั่งการแสดงผลที่ต่างกัน

29. วงจรสับเบอร์มีหน้าที่อย่างไรในการควบคุมมอเตอร์ไฟฟ้ากระแสตรง

- ก. ป้องกันมอเตอร์ไฟฟ้ากระแสตรงเสียหายเมื่อเปลี่ยนทิศทางการเคลื่อนที่ของมอเตอร์
- ข. ป้องกันไฟฟ้าลัดวงจรที่แหล่งจ่ายเมื่อมอเตอร์ขับเคลื่อนด้วยความเร็วสูงสุด
- ค. เพื่อชดเชยค่าความเหนี่ยวนำของขดลวดมอเตอร์เมื่อมอเตอร์ไฟฟ้ากระแสตรง
- ง. เพื่อเพิ่มปริมาณกระแสไฟฟ้าสภาวะเริ่มต้นและลดปริมาณกระแสในสภาวะที่มอเตอร์หมุนคงที่

30. ข้อใดคือหน้าที่ของวงจร R-2R Ladder Network

- ก. เป็นวงจรสำหรับเปลี่ยนสัญญาณแอนาลอกเป็นสัญญาณดิจิตอล
- ข. เป็นวงจรสำหรับเปลี่ยนสัญญาณดิจิตอลเป็นสัญญาณแอนาลอก
- ค. เป็นวงจรแบ่งแรงดันไฟฟ้าสำหรับการเชื่อมต่อสวิตช์หลายๆ ตัว
- ง. เป็นวงจรสำหรับแบ่งกระแสไฟฟ้าสำหรับการเชื่อมต่อโหลดหลายๆ ตัว

31. การเชื่อมต่อวงจรกำเนิดสัญญาณนาฬิกาที่ใช้แร่คริสตัลกับตัวเก็บประจุ เป็นการเชื่อมต่อโหมดใดของการกำหนดค่าให้กับ PIC16F877

- ก. โหมด LP Oscillator ข. โหมด XT Oscillator ค. โหมด HS Oscillator ง. โหมด RC Oscillator

32. การปรับความสว่างของตัวอักษรบนจอแสดงผล LCD สามารถทำได้โดยการเชื่อมต่อตัวต้านทานปรับค่าได้เข้ากับแหล่งจ่ายไฟ โดยจะต้องปรับค่าแรงดันระหว่าง 0-5V เข้าที่ขาใดของจอแสดงผล LCD

- ก. ขา Vdd ข. ขา Vss ค. ขา Vo ง. ขา A

33. OP Code ของ PIC16F877 คือไฟล์ล์ชนิดใด

- ก. ASM ข. BAS ค. PBP ง. HEX

34. สภาวะการทำงานของไมโครคอนโทรลเลอร์เมื่อเข้าสู่โหมด RUN มีกี่สภาวะ อะไรบ้าง

- ก. 2 สภาวะ คือ Run และ Execute ข. 2 สภาวะ คือ Program และ Run
- ค. 3 สภาวะ คือ Program Run และ Execute ง. 3 สภาวะ คือ Program Run และ Wait

35. การเชื่อมต่อจอแสดงผล LCD ถ้าไม่ใช้คำสั่ง LCDIN จะต้องต่อวงจรอย่างไร

- ก. นำขา R/W ต่อลง Ground ข. นำขา R/W ต่อแหล่งจ่ายไฟ 5 โวลต์
- ค. นำขา E ต่อลง Ground ง. นำขา E ต่อแหล่งจ่ายไฟ 5 โวลต์

36. ขาสัญญาณ MCLR ของ PIC16F877 ย่อมาจากข้อใด

- ก. Master Clear ข. Memory Clear ค. Mode Clear Register ง. Mode Controller

37. ข้อใดกล่าวถูกต้องเกี่ยวกับ WDT ภายในไมโครโปรเซสเซอร์ PIC16F877

- ก. WDT เป็นนาฬิกาทำหน้าที่คอยเฝ้าดูการทำงานของไมโครโปรเซสเซอร์ที่ผิดปกติ
- ข. WDT เป็นหน่วยความจำสำหรับการเก็บข้อมูลนาฬิกาภายในไมโครโปรเซสเซอร์
- ค. WDT เป็นรีจิสเตอร์ทำหน้าที่ตรวจสอบความผิดพลาดการส่งข้อมูล
- ง. WDT เป็นโมดูลที่เปลี่ยนสัญญาณนาฬิกาเป็นกระแสไฟฟ้าในโหมดประหยัดพลังงาน

38. ข้อใดกล่าวไม่ถูกต้องเกี่ยวกับการใช้งานการเชื่อมต่อแบบโมดูลการสื่อสาร

- ก. การใช้งานการสื่อสารแบบ I²C มีการสื่อสารแบบอนุกรมมี 2 สัญญาณ SCL และ SDA
- ข. การใช้งานการสื่อสารแบบขนาน สามารถส่งข้อมูลสูงสุดได้ครั้งละ 8 บิต
- ค. การใช้งานการสื่อสารแบบ USART ใช้แอดเดรสขนาด 8 บิตในการตรวจสอบตำแหน่งข้อมูล
- ง. การใช้งานการสื่อสารแบบสายเดี่ยว จะใช้คำสั่ง OWIN สำหรับการอ่านข้อมูล และ OWOUT สำหรับการส่งข้อมูลไปยังอุปกรณ์สื่อสารสายเดี่ยว

39. ลักษณะการควบคุมงานแบบใดควรมีการออกแบบการควบคุมแบบ Jogging

- ก. การควบคุมขั้นบันได
- ข. การกดคีย์บอร์ดโทรศัพท์เคลื่อนที่
- ค. การควบคุมความเร็วรถไฟฟ้า
- ง. การควบคุมทิศทางการหมุนของมอเตอร์

40. จากโปรแกรมการทดลอง ข้อใดกล่าวไม่ถูกต้อง

```

S1    VAR    PORTA.0
S2    VAR    PORTA.1
L1    VAR    PORTB.1

TRISA    =    %111111
TRISB    =    %00000000
ADCON1 = 7
PORTB    = 0

LOOP: IF (S1 = 0) AND (S2 = 1) THEN HIGH L1
      IF (S1 = 1) AND (S2 = 0) THEN LOW L1
      PAUSE 50
      GOTO LOOP
END

```

- ก. มีการกำหนดให้พอร์ต A เป็นอินพุตชนิดสัญญาณดิจิทัล
- ข. การต่อวงจรสวิตช์ S1 และ S2 เมื่อกดจะได้ลอจิก 1
- ค. ถ้ายังกดสวิตช์ S1 แล้วจะได้ลอจิก 1 ที่ L1 ตลอดเวลาจนกว่าจะกด S2
- ง. ถ้ากด S1 และ S2 พร้อมกันจะทำให้ L1 กระพริบ

ภาคผนวก ง

ประวัติผู้เชี่ยวชาญ

ประวัติผู้เชี่ยวชาญ

อาจารย์สมบุรณ์ เนียมกล้า

การศึกษา

- ปริญญาตรี ครุศาสตร์อุตสาหกรรมบัณฑิต สาขาครุศาสตร์ไฟฟ้า
สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ
วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมไฟฟ้า
มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี
- ปริญญาโท ครุศาสตร์อุตสาหกรรมมหาบัณฑิต สาขาบริหารอาชีพและเทคโนโลยีศึกษา
สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ประสบการณ์

- ศึกษาดูงาน / ฝึกงาน Electronic Engineering, Telecommunication, Industrial
Automation Space and Satellite Technology, Robotic
Technology ประเทศญี่ปุ่น
- ศึกษาดูงาน / ฝึกงาน CAD / CAM Technology, Microprocessor and Microcontroller
ประเทศแคนาดา
- ศึกษาดูงาน / ฝึกงาน Process Instruments Standard Calibration ประเทศฟินแลนด์
- ศึกษาดูงาน / ฝึกงาน Power Electronics, Automotives Electronics, Process Control
System Mechatronics , Robot ประเทศสหพันธ์สาธารณรัฐเยอรมัน
- ศึกษาดูงาน / ฝึกงาน Electronic Process Control and Instrumentation ประเทศอิตาลี

- วิทยากรอบรม ด้านการออกแบบวงจรดิจิทัลด้วยชิพ CPLD และ FPGA ให้กับหน่วยงาน
ด้านการศึกษาและอุตสาหกรรมทั้งภาครัฐและเอกชนหลายแห่ง
- วิทยากรอบรม ไมโครคอนโทรลเลอร์และการประยุกต์ใช้งาน ให้กับหน่วยงาน
ด้านการศึกษาและอุตสาหกรรมทั้งภาครัฐและเอกชนหลายแห่ง
- วิทยากรอบรม การสร้างและการเขียนโปรแกรมควบคุมหุ่นยนต์ ให้กับหน่วยงาน
ด้านการศึกษาและอุตสาหกรรมทั้งภาครัฐและเอกชนหลายแห่ง
- อาจารย์พิเศษ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีปทุมวัน
- วิจัยและพัฒนา ออกแบบระบบสื่อสารและควบคุมระบบไฟฟ้าในรถยนต์ด้วย
ไมโครคอนโทรลเลอร์ ออกแบบและพัฒนาหุ่นยนต์อัตโนมัติลำเลียงชิ้นส่วน
ในสายงานการผลิตของโรงงานผลิตกล้องถ่ายรูป NIKON

ประวัติการทำงาน

- หัวหน้าแผนก และหัวหน้าคณะวิชาไฟฟ้ากำลัง วิทยาลัยเทคนิคสกลนคร
รองผู้อำนวยการวิทยาลัยเทคนิคนครพนม (มหาวิทยาลัยนครพนม)
เจ้าหน้าที่ฝึกอบรม ระดับ 7 สถาบันพัฒนาครูอาชีวศึกษา
นักพัฒนาทรัพยากรบุคคล ระดับ 8 สำนักพัฒนาสมรรถนะครูและบุคลากรอาชีวศึกษา

อาจารย์พันธุ์ โคมพิทยา

การศึกษา

- ปริญญาตรี วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมอิเล็กทรอนิกส์
มหาวิทยาลัยเทคโนโลยีมหานคร
- ปริญญาโท วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมอิเล็กทรอนิกส์
มหาวิทยาลัยเทคโนโลยีมหานคร

ประสบการณ์

- วิทยากรอบรม ด้านการออกแบบวงจรอิเล็กทรอนิกส์ ให้กับหน่วยงาน
ด้านการศึกษาทั้งภาครัฐและเอกชน
- วิทยากรอบรม ไมโครคอนโทรลเลอร์และการประยุกต์ใช้งาน ให้กับหน่วยงาน
ด้านการศึกษาและอุตสาหกรรมทั้งภาครัฐและเอกชนหลายแห่ง
- วิจัยและพัฒนา การออกแบบวงจรอิเล็กทรอนิกส์ การออกแบบวงจรรวม
การออกแบบการควบคุมด้วยไมโครคอนโทรลเลอร์
ที่ปรึกษาจัดทำปริญญานิพนธ์ นักศึกษาระดับ ปวส. และ ปริญญาตรี

ประวัติการทำงาน

- แผนกวิชาช่างอิเล็กทรอนิกส์ วิทยาลัยเทคนิคสุโขทัย
- อาจารย์แผนกวิชาช่างอิเล็กทรอนิกส์ วิทยาลัยการอาชีพธาตุพนม มหาวิทยาลัยนครพนม
- รองคณบดีฝ่ายวิชาการ วิทยาลัยการอาชีพธาตุพนม มหาวิทยาลัยนครพนม

อาจารย์มณฑล พักเฒ

การศึกษา

- ปริญญาตรี อุตสาหกรรมศาสตรบัณฑิต สาขาวิศวกรรม
มหาวิทยาลัยศรีปทุม
- ปริญญาโท วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมไฟฟ้า
มหาวิทยาลัยนเรศวร

ประสบการณ์

- วิจัยและพัฒนา การประมวลผลภาพด้วยคอมพิวเตอร์
การออกแบบเครื่องจักรกลการเกษตรด้วยไมโครคอนโทรลเลอร์
ที่ปรึกษาจัดทำปริญญานิพนธ์ นักศึกษาระดับ ปริญญาตรี

ประวัติการทำงาน

- ครูช่าง ภาควิศวกรรมไฟฟ้าและคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร
- อาจารย์สาขาวิชาเทคโนโลยีอิเล็กทรอนิกส์ คณะเทคโนโลยีอุตสาหกรรม
มหาวิทยาลัยราชภัฏพิบูลสงคราม

อาจารย์วิษณุ บัวเทศ

การศึกษา

- ปริญญาตรี วิทยาศาสตร์บัณฑิต สาขาไฟฟ้าอุตสาหกรรม
มหาวิทยาลัยราชภัฏกำแพงเพชร
- ปริญญาโท ครุศาสตร์อุตสาหกรรมมหาบัณฑิต สาขาไฟฟ้า
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ประสบการณ์

- วิทยาการอบรม ด้านการอนุรักษ์พลังงาน

- วิจัยและพัฒนา การสร้างและหาประสิทธิภาพชุดการสอน วิชาการควบคุมมอเตอร์
การออกแบบและสร้างระบบน้ำพุและรดน้ำต้นไม้อัตโนมัติ
ด้วยพลังงานแสงอาทิตย์
การออกแบบการควบคุมด้วยไมโครคอนโทรลเลอร์
ที่ปรึกษาจัดทำปริญญานิพนธ์ นักศึกษาระดับ ปวส. และ ปริญญาตรี

ประวัติการทำงาน

อาจารย์สาขาวิชาเทคโนโลยีไฟฟ้าอุตสาหกรรม คณะเทคโนโลยีอุตสาหกรรม
มหาวิทยาลัยราชภัฏกำแพงเพชร

อาจารย์มนูญ บุญประมุข

การศึกษา

- ปริญญาตรี วิทยาศาสตร์บัณฑิต สาขาไฟฟ้าอุตสาหกรรม
มหาวิทยาลัยราชภัฏกำแพงเพชร
- ปริญญาโท วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมไฟฟ้า
มหาวิทยาลัยเกษตรศาสตร์

ประสบการณ์

- วิทยาการอบรม การเขียนโปรแกรมและการออกแบบวงจรไมโครคอนโทรลเลอร์
โปรแกรมวิชาไฟฟ้า คณะเทคโนโลยีอุตสาหกรรม

- วิจัยและพัฒนา การออกแบบการควบคุมด้วยไมโครคอนโทรลเลอร์
ที่ปรึกษาจัดทำปริญญานิพนธ์ นักศึกษาระดับปริญญาตรี

ประวัติการทำงาน

อาจารย์สาขาวิชาเทคโนโลยีไฟฟ้าอุตสาหกรรม คณะเทคโนโลยีอุตสาหกรรม
มหาวิทยาลัยราชภัฏกำแพงเพชร

ภาคผนวก จ

การเผยแพร่ผลงานวิจัย

ชุดการทดลองการประยุกต์ใช้งานไมโครโปรเซสเซอร์ PIC16F877
PIC16F877 Microprocessor Application Laboratory

เกียรติชัย บรรลุผลสกุล

คณะเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยราชภัฏพินุลสงคราม

คำย่อ

การวิจัยครั้งนี้มีวัตถุประสงค์เพื่อสร้างและหาประสิทธิภาพชุดฝึกการทดลองการประยุกต์ใช้งานไมโครโปรเซสเซอร์ PIC16F877 เครื่องมือที่ใช้ในการวิจัยนี้ ได้แก่ ชุดฝึกการทดลองการประยุกต์ใช้งานไมโครโปรเซสเซอร์ PIC16F877 ประกอบไปด้วย บอร์ดการทดลองจำนวน 5 การทดลอง ใบงานการทดลองการประยุกต์ใช้งานไมโครโปรเซสเซอร์ PIC16F877 และ แบบทดสอบรายวิชาการประยุกต์ใช้ไมโครโปรเซสเซอร์ โดยนำชุดฝึกการทดลองให้ผู้เชี่ยวชาญจำนวน 5 คน ประเมินความเหมาะสมของการทดลอง และนำไปทดลองกับกลุ่มตัวอย่าง คือนักศึกษามหาวิทยาลัยราชภัฏพินุลสงครามที่เรียนเรียนวิชาการประยุกต์ใช้งานไมโครโปรเซสเซอร์ ภาคเรียนที่ 2 ปีการศึกษา 2553 สุ่มแบบจงจำนวน 10 คน เมื่อนักศึกษากลุ่มตัวอย่างได้เรียนด้วยชุดฝึกการทดลองแล้ว จึงทำการทดสอบแบบฝึกหัดท้ายการทดลอง และ แบบทดสอบรายวิชาการประยุกต์ใช้งานไมโครโปรเซสเซอร์และคะแนนทั้งสองส่วนวิเคราะห์และสรุปผล พบว่าชุดฝึกการทดลองการประยุกต์ใช้งานไมโครโปรเซสเซอร์ PIC16F877 มีคะแนนเฉลี่ย 4.35 มีความเหมาะสมในระดับมาก มีประสิทธิภาพ 98.4.25 เป็นไปเกณฑ์มาตรฐานที่ตั้งไว้ แสดงว่าชุดการทดลองการประยุกต์ใช้งานไมโครโปรเซสเซอร์ PIC16F877 สามารถนำไปใช้ในการเรียนการสอนได้อย่างมีประสิทธิภาพ

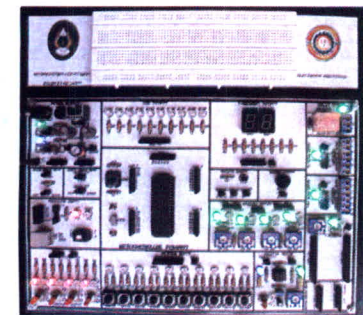
นำ

สาขาวิชาเทคโนโลยีอิเล็กทรอนิกส์ คณะเทคโนโลยีอุตสาหกรรม [1] กำหนดวัตถุประสงค์เพื่อบัณฑิตให้เป็น นักเทคโนโลยีอุตสาหกรรม และมีทักษะด้านวิชาชีพเทคโนโลยีอิเล็กทรอนิกส์ ที่มีสมบรูณทั้งทางร่างกายและจิตใจ มีคุณธรรม และจริยธรรม มีความคิดริเริ่มสร้างสรรค์ไม่เรียนรู้และก้าววิจัย เพื่อพัฒนาเทคโนโลยีอิเล็กทรอนิกส์อย่างต่อเนื่องและเป็นระบบมีประสิทธิภาพและสิทธิผล ผู้สอนได้นำเอารูปแบบสมรรถนะฐานที่มุ่งเน้นการพัฒนาทางด้านทฤษฎีและปฏิบัติ และพฤติกรรมเข้ามามีส่วนร่วมกับการเรียนการสอนเพื่อให้ผู้เรียน ประสบผลสัมฤทธิ์ทางการเรียน มีความรู้ มีทักษะ และนิสัยที่พึงปรารถนา ซึ่งพัฒนา และ ณรงค์วิทย์ [2,3] กล่าวว่าทัศนคติ พฤติกรรม ความรู้ และทักษะการปฏิบัติงานให้เกิดคุณภาพ ประสิทธิภาพและประสิทธิผลสูงสุด ในการปฏิบัติหน้าที่ของบุคคล [4] กล่าวว่า ชุดการสอน เป็นวัดกรรมการศึกษา และการสอนที่เกี่ยวกับการปฏิรูปหลักสูตรเป็นระบบการนำสื่อประสมที่สอดคล้องกับเนื้อหาและประสบการณ์ของแต่ละหน่วยมาช่วยให้เกิดการเปลี่ยนแปลงพฤติกรรมการเรียนรู้ของผู้เรียนแต่ละคน ซึ่งผู้วิจัยนำแนวคิดนี้มาสร้างชุดการทดลองการประยุกต์ใช้งานไมโครโปรเซสเซอร์ PIC16F877 สำหรับการปฏิบัติการของนักศึกษาที่ลงทะเบียนเรียนการประยุกต์ใช้งานไมโครโปรเซสเซอร์ ซึ่งสมุนหาและ อรรถพรณ [5] กล่าวว่า การเรียนรู้แบบมีส่วนร่วมช่วยเตรียมนักศึกษาให้พร้อมที่จะเผชิญกับชีวิตจริง เพราะลักษณะของการเรียนรู้แบบมีส่วนร่วม เปิดโอกาสให้นักเรียนได้รับผิดชอบต่อการเรียนรู้ของตนเอง ได้ลงมือปฏิบัติ ทำกิจกรรมกลุ่ม ผึกฝนทักษะเรียนรู้ทักษะการปฏิบัติงาน

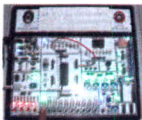
ประสงค์

เพื่อสร้างชุดฝึกการทดลองการประยุกต์ใช้งานไมโครโปรเซสเซอร์ PIC16F877

เพื่อหาประสิทธิภาพชุดฝึกการทดลองการประยุกต์ใช้งานไมโครโปรเซสเซอร์ PIC16F877



ภาพแสดงบอร์ดการทดลอง



ภาพแสดงตัวอย่างผลการทดลองที่ 1



ภาพแสดงตัวอย่างผลการทดลองที่ 2

วิธีการดำเนินงาน

1. วิเคราะห์เนื้อหา และจัดประสงครายวิชาการประยุกต์ใช้งานไมโครโปรเซสเซอร์ และ ออกแบบชุดฝึกการทดลองการประยุกต์ใช้งานไมโครโปรเซสเซอร์ PIC16F877
2. การสร้างชุดฝึกการทดลองการประยุกต์ใช้งานไมโครโปรเซสเซอร์ PIC16F877 นำไปให้ผู้เชี่ยวชาญตรวจสอบและปรับปรุงตามคำแนะนำ
3. ประเมินความเหมาะสมชุดฝึกการทดลองการประยุกต์ใช้งานไมโครโปรเซสเซอร์ PIC16F877 โดยผู้เชี่ยวชาญ จำนวน 5 ท่าน
4. นำชุดฝึกการทดลองการประยุกต์ใช้งานไมโครโปรเซสเซอร์ PIC16F877ทดลองใช้กับนักศึกษาในกลุ่มตัวอย่าง
5. เก็บข้อมูลประสิทธิภาพของกระบวนการจากผลการทดสอบด้วยแบบฝึกหัดท้ายการทดลอง ผลการปฏิบัติงาน และประสิทธิภาพของผลลัพธ์ที่วัดได้จากการทำแบบทดสอบรายวิชาการประยุกต์ใช้งานไมโครโปรเซสเซอร์
6. วิเคราะห์ข้อมูล สรุปผลการวิจัย



ภาพแสดงตัวอย่างผลการทดลองที่ 3



ภาพแสดงตัวอย่างผลการทดลองที่ 4



ภาพแสดงตัวอย่างผลการทดลองที่ 5

สรุปผลการวิจัย

1. ผลความเหมาะสมชุดฝึกการทดลองการประยุกต์ใช้งานไมโครโปรเซสเซอร์ PIC16F877 โดยรวมอยู่ในระดับมาก มีค่าเฉลี่ยเท่ากับ 4.35 ผลรายข้อพบว่าในภาพรวมโดยผู้เชี่ยวชาญมีความคิดเห็นอยู่ในระดับมากถึงมากที่สุด ประเด็นที่ผู้เชี่ยวชาญมีระดับความคิดเห็นที่มีค่าเฉลี่ยสูงสุดคือ จำนวนข้อสอบเหมาะสมกับวัตถุประสงค์เชิงพฤติกรรม ค่าเฉลี่ยเท่ากับ 4.6 รองลงมาคือคำถามตรงตามวัตถุประสงค์เชิงพฤติกรรม ค่าเฉลี่ยเท่ากับ 4.4 และข้อที่มีค่าเฉลี่ยต่ำที่สุด ได้แก่ คำถามและคำตอบมีเป้าหมายที่ชัดเจน คำถามมีความยากง่ายเหมาะสม ค่าเฉลี่ยเท่ากับ 4.2
2. ผลการวิเคราะห์หาประสิทธิภาพชุดฝึกการทดลองการประยุกต์ใช้งานไมโครโปรเซสเซอร์ PIC16F877 ซึ่งเกณฑ์ที่กำหนดไว้คือ 80/80 ผู้วิจัยนำชุดฝึกการทดลองการประยุกต์ใช้งานไมโครโปรเซสเซอร์ PIC16F877 ไปใช้กับกลุ่มตัวอย่างจำนวน 10 คน ผลปรากฏว่านักศึกษากลุ่มตัวอย่างมีคะแนนทำแบบฝึกหัดหลังการทดลองร้อยละ 85.29 และ คะแนนแบบทดสอบรายวิชาการประยุกต์ใช้งานไมโครโปรเซสเซอร์ PIC16F877 ร้อยละ 84.25 เป็นผลให้ประสิทธิภาพชุดฝึกการทดลองการประยุกต์ใช้งานไมโครโปรเซสเซอร์ PIC16F877 มีค่าเป็น 85.29/84.25

เอกสารอ้างอิง

- [1] สภาวิชาการ (2543). หลักสูตรสถาบันราชภัฏสาขาวิชาวิทยาศาสตร์โปรแกรมวิชาเทคโนโลยีอุตสาหกรรม ระดับปริญญาตรี(หลักสูตรปริญญา) หลักสูตรพุทธศักราช 2543
- [2] วิไลนา พัฒนพงศ์ . (2546). BSC และ KPI เพื่อการเติบโตขององค์กร อย่างยั่งยืน. (หน้า 33)กรุงเทพมหานคร : สำนักพิมพ์ แปซิฟิค.
- [3] ณรงค์วิทย์ แสนทอง.(2547). มารูจัก COMPETENCY กันเถอะ. (หน้า 199) กรุงเทพมหานคร : เอช อาร์ เซ็นเตอร์.
- [4] ไชยศ เรืองสุวรรณ..(2526). เทคโนโลยีทางการศึกษา หลักการและแนวปฏิบัติ.(หน้า 199) กรุงเทพมหานคร : สำนักพิมพ์วัฒนาพานิช.
- [5] สมณฑา พรหมบุญ และอรรถพรณ พรสิมา. (2549). “การเรียนรู้แบบมีส่วนร่วม” ในทฤษฎีการเรียนรู้แบบมีส่วนร่วม : ต้นแบบการเรียนรู้ทางด้านหลักทฤษฎีและแนวปฏิบัติ. (หน้า 34-35).กรุงเทพมหานคร : สำนักงานคณะกรรมการการศึกษาแห่งชาติ.

ชุดการทดลองการประยุกต์ใช้งานไมโครโปรเซสเซอร์ PIC16F877

เกียรติชัย บรรณผลสกุล

คณะเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยราชภัฏพิบูลสงคราม

*corresponding author email : kiatchai10@hotmail.com

บทคัดย่อ

การวิจัยครั้งนี้มีวัตถุประสงค์เพื่อสร้างและหาประสิทธิภาพชุดฝึกการทดลองการประยุกต์ใช้งานไมโครโปรเซสเซอร์ 16F877

เครื่องมือที่ใช้ในการวิจัยนี้ ได้แก่ ชุดฝึกการทดลองการประยุกต์ใช้งานไมโครโปรเซสเซอร์ 16F877 ประกอบไปด้วย บอร์ดการทดลองจำนวน 5 การทดลอง ใบงานการทดลองการประยุกต์ใช้งานไมโครโปรเซสเซอร์ PIC16F877 และ แบบทดสอบรายวิชาการประยุกต์ใช้งานไมโครโปรเซสเซอร์ โดยนำชุดฝึกการทดลองให้ผู้เชี่ยวชาญจำนวน 5 คน ประเมินความเหมาะสมชุดฝึกการทดลอง และนำไปทดลองกับกลุ่มตัวอย่าง คือ นักศึกษามหาวิทยาลัยราชภัฏพิบูลสงครามที่ลงทะเบียนเรียนวิชาการประยุกต์ใช้งานไมโครโปรเซสเซอร์ ภาคเรียนที่ 2 ปีการศึกษา 2553 ซึ่งสุ่มแบบเจาะจงจำนวน 10 คน เมื่อนักศึกษากลุ่มตัวอย่างได้เรียนด้วยชุดฝึกการทดลองแล้ว จึงทำการทดสอบด้วยแบบฝึกหัดท้ายการทดลอง และ แบบทดสอบรายวิชาการประยุกต์ใช้งานไมโครโปรเซสเซอร์และนำผลคะแนนทั้งสองส่วนวิเคราะห์และสรุปผล

ผลการวิจัยปรากฏว่า ชุดฝึกการทดลองการประยุกต์ใช้งานไมโครโปรเซสเซอร์ PIC16F877 มีความเหมาะสมคะแนนเฉลี่ย 4.24 มีความเหมาะสมในระดับมาก มีประสิทธิภาพ 85.29/84.25 สูงกว่าเกณฑ์ที่กำหนด 80/80 ซึ่งเป็นไปตามเกณฑ์มาตรฐานที่ตั้งไว้ แสดงว่าชุดการทดลองการประยุกต์ใช้งานไมโครโปรเซสเซอร์ 16F877 สามารถนำไปใช้ในการเรียนการสอนได้อย่างมีประสิทธิภาพ

คำสำคัญ : ชุดการทดลอง ไมโครโปรเซสเซอร์ PIC16F877

PIC16F877 Microprocessor Application Laboratory

Kiatchai Banlupholsakul

Faculty of Industrial Technology, Pibulsongkram Rajabhat University

corresponding author email : kiatchai10@hotmail.com

Abstract

The purposes of this research were to develop and find an Efficiency of PIC16F877 Microprocessor Application Laboratory

The tools used in this research include PIC16F877 Microprocessor Application Laboratory 5 Lab, Microprocessor application experiment document and Final test of Applications microprocessor course. Evaluating by 5 expert of electronic and Trial with 10 student sampling by purpose in second semester of 2010. When students learn a set of training samples to trial. The exercises test the experiments and tests microprocessor applications course and the results of both the analysis and conclusions.

The findings appear Training trials Applications PIC16F877 microprocessor is appropriate average 4.24 is appropriate on many levels. Higher efficiency criteria set 85.29/84.25 80/80, which is the benchmark set. The experiments show that the application 16F877 microprocessor can be used in teaching effectively.

keywords : Laboratory, Microprocessor , PIC16F877