	สาขาวิชา	อิเล็กทรอนิกส์	ใบงานการทดลองที่ 5	
	ชื่อวิชา	ไมโครคอนโทรลเลอร์		
	รหัสวิชา	2105-2105	หน้าที่	
	ชื่องาน	งานโปรแกรมอ่านค่าจากพอร์ตแอนาล็อกและการใช้ PWM	32	

คำชี้แจง ให้ผู้เรียนทุกคนทำการทดลองตามใบงานการทดลองที่ 4 เรื่องงานโปรแกรมอ่านค่าจากพอร์ตแอนาล็อกและการใช้ PWM ตามขั้นตอนการปฏิบัติงาน

จุดประสงค์ทั่วไป

เพื่อให้มีทักษะการปฏิบัติงานโปรแกรมอ่านค่าจากพอร์ตแอนาล็อกและการใช้ PWM

จุดประสงค์การเรียนรู้เชิงพฤติกรรม (เพื่อให้ผู้เรียน.....)


1. สามารถใช้โปรแกรม Arduino IDE ในการเขียนโปรแกรมภาษา C เบื้องต้นได้อย่างถูกต้อง
2. สามารถใช้งานไมโครคอนโทรลเลอร์ บอร์ด Arduino UNO R3 เบื้องต้นได้อย่างถูกต้อง
3. สามารถประกอบและทดสอบวงจรการอ่านค่าพอร์ตแอนาล็อกและการใช้ PWM ได้อย่างถูกต้อง
4. สามารถเขียนโปรแกรมใช้งานพอร์ตทำหน้าที่อินพุทพอร์ตเบื้องต้นได้อย่างถูกต้อง
5. สามารถประยุกต์ใช้งานไมโครคอนโทรลเลอร์บอร์ด Arduino UNO R3 เบื้องต้นได้อย่างถูกต้อง
6. มีทัศนคติในการแสวงหาความรู้เพิ่มเติม การทำงานด้วยความประณีต รอบคอบและปลอดภัย

เครื่องมือและอุปกรณ์

- | | | |
|--|---|---------|
| 1. โปรแกรม Arduino IDE 1.8.4 หรือสูงกว่า | 1 | โปรแกรม |
| 2. สาย USB สำหรับ Arduino Uno R3 | 1 | เส้น |
| 3. ชุดทดลอง Arduino Uno R3 พร้อมสายต่อวงจร | 1 | ชุด |
| 4. เครื่องคอมพิวเตอร์แบบพกพา | 1 | เครื่อง |
| 5. แผงต่อวงจร | 1 | ตัว |
| 6. มัลติมิเตอร์ | 1 | ตัว |
| 7. เครื่องมือประจำตัว | 1 | ชุด |

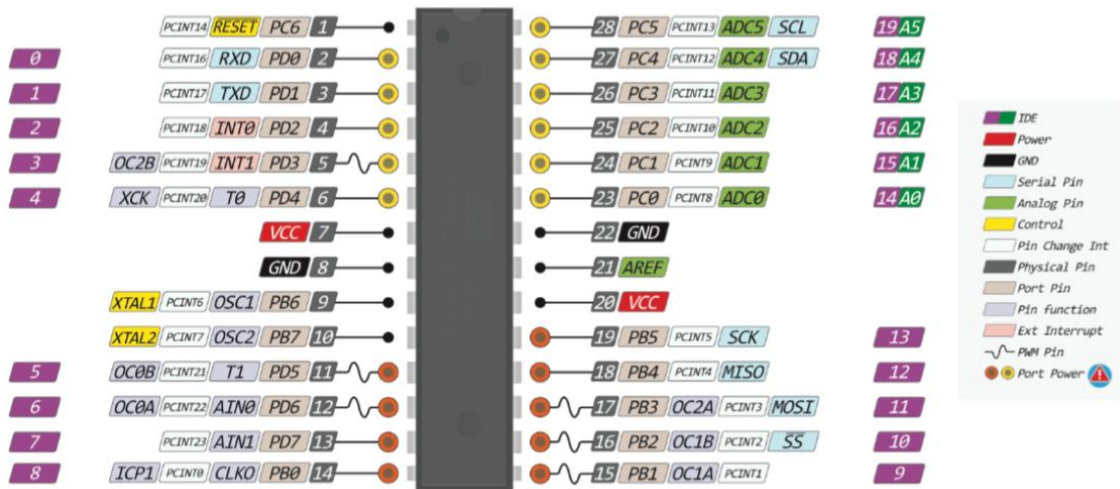
ข้อห้ามและข้อควรระวัง

1. ไม่เล่นและหยอกล้อกันในเวลาปฏิบัติงาน
2. ควรระวังไม่วางบอร์ด Arduino Uno R3 หรือซีลต่างๆ บนโต๊ะโลหะหรือที่วางที่เป็นโลหะเพราะอาจเกิดการลัดวงจรของภาคจ่ายไฟได้
3. ไม่ควรต่อสายต่อวงจรในบอร์ด Arduino Uno R3 ทิ้งไว้ ควรถอดสายต่อวงจรออกให้หมด เพราะผล การทดลองอาจเกิดการผิดพลาดไม่เป็นไปตามทฤษฎีได้
4. ไม่ควรถอดสายสายไหลต USB เข้าออกตลอดเวลา เพราะอาจทำให้ภาคจ่ายไฟของบอร์ด Arduino Uno R3 เสียหายได้
5. ควรระวังเครื่องมือและอุปกรณ์เสียหายจากการปฏิบัติงานไม่ถูกต้องตามขั้นตอนและไม่ปลอดภัย

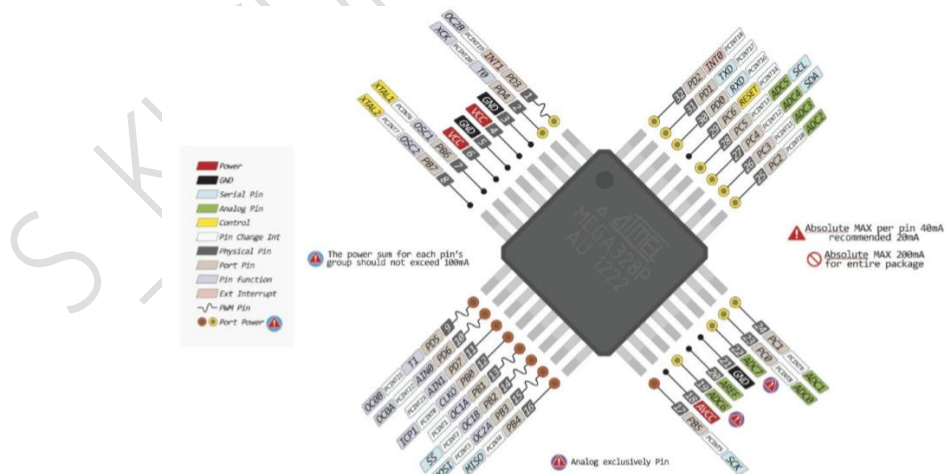
	สาขาวิชา	อิเล็กทรอนิกส์	ใบงานการทดลองที่ 5
	ชื่อวิชา	ไมโครคอนโทรลเลอร์	
	รหัสวิชา	2105-2105	หน้าที่
	ชื่องาน	งานโปรแกรมอ่านค่าจากพอร์ตแอนาล็อกและการใช้ PWM	33

ทฤษฎี


บอร์ด Arduino รุ่นที่ใช้ไอซีไมโครคอนโทรลเลอร์ ATMEGA328, ATMEGA168, ATMEGA8 มีพอร์ตที่สามารถรับสัญญาณเข้าที่เป็นสัญญาณแอนาล็อกได้ 6 ช่อง (6 ขา) A0, A1, ...A5 สำหรับบอร์ดที่ใช้ไอซีที่มีตัวถัง DIP (ตัวถังตั้งตะขาก) เช่นบอร์ดรุ่น UNO และสามารถรับสัญญาณได้ 8 ช่องสำหรับบอร์ดที่ใช้ชิพไอซีที่มีตัวถังแบบ SMD (อุปกรณ์ผิวหน้า) เช่น บอร์ดรุ่น Mini และ Nano ส่วนบอร์ดรุ่น Mega มีช่องสัญญาณที่สามารถสัญญาณแอนาล็อกได้ 16 ช่อง ภายในตัวชิพมีโมดูลแปลงสัญญาณแอนาล็อกเป็นดิจิทัลขนาด 10 บิต ดังนั้นเมื่ออ่านค่าเข้ามาชิพจะแปลงค่าได้เป็นค่าตั้งแต่ 0 จนถึง 1023 ที่ค่าแรงดันของสัญญาณแอนาล็อกที่รับเข้ามาอยู่ในช่วง 0 ถึง 5 โวลต์



รูปที่ 5.1 แสดงตำแหน่งขาและหน้าที่ของชิพยูนิตตัวถังแบบ DIP



รูปที่ 5.2 แสดงตำแหน่งขาและหน้าที่ของชิพยูนิตตัวถังแบบ SMD

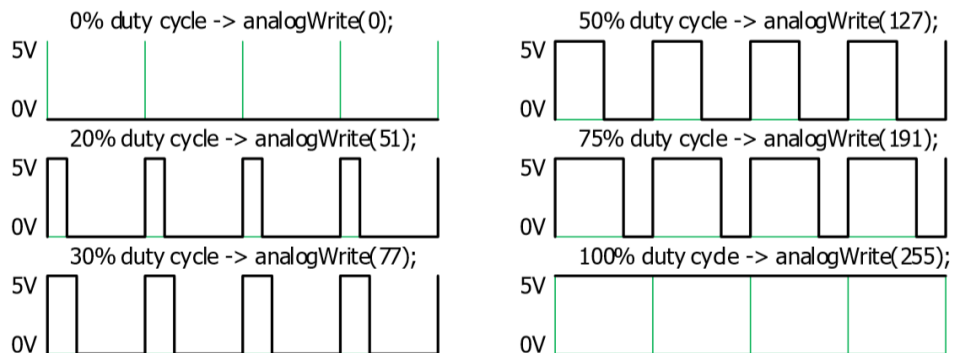
	สาขาวิชา	อิเล็กทรอนิกส์	ใบงานการทดลองที่ 5
	ชื่อวิชา	ไมโครคอนโทรลเลอร์	
	รหัสวิชา	2105-2105	หน้าที่
	ชื่องาน	งานโปรแกรมอ่านค่าจากพอร์ตแอนาล็อกและการใช้ PWM	34

ขาพอร์ตแอนาล็อกจะมีบางส่วนที่สามารถนำไปใช้งานแบบดิจิทัลได้ นั่นก็หมายความว่า สามารถใช้คำสั่ง `digitalWrite()`, `digitalRead()` ได้เหมือนกับขาดิจิทัลปกติ ขาพอร์ตแอนาล็อกที่นำมาใช้งานในโหมดดิจิทัลได้มีเพียง 6 ขาคือ A0, A1, ...A5 ส่วน A6 และ A7 ของบอร์ดรุ่น Mini และ Nano ใช้รับสัญญาณเข้าที่เป็นสัญญาณแอนาล็อกเพียงอย่างเดียว ดังนั้นเมื่อใช้งานในโหมดดิจิทัลการใช้งานจะใช้คำสั่งเดียวกันกับคำสั่งที่ควบคุมขาดิจิทัลเพียงแต่การระบุขาใช้งานจะต้องใส่ช่องขาสัญญาณว่า A นำหน้า เช่น

`pinMode(A0,OUTPUT);` หมายถึงให้ขา A0 ทำงานในโหมดดิจิทัลที่เป็นเอาต์พุตพอร์ต
`digitalWrite(A0,HIGH);` หมายถึงให้เขียนลอจิก 1 ไปที่ขาพอร์ต A0

Arduino มีขาพอร์ตที่สามารถส่งสัญญาณออกแบบผสมสัญญาณทางความกว้างของพัลส์ได้ หรือ ที่เรียกว่า PWM โดยบอร์ดที่ใช้ซีพียูเบอร์ Atmega168, Atmega328 (บอร์ดรุ่น UNO, Mini, Nano) มีขาที่สามารถส่งสัญญาณออกได้ 6 ขาคือ D3, D5, D6, D9, D10, และ D11 ความถี่ของสัญญาณ PWM จะมีค่า โดยประมาณที่ 490 Hz ส่วนค่าตัวตึ้ไซเคิลสามารถใส่ค่าได้ตั้งแต่ 0-255 ใช้สำหรับกำหนดค่าตัวตึ้ไซเคิล ระหว่าง 0-100% ดังรูป

Pulse Width Modulation




รูปที่ 5.3 แสดงรูปคลื่น PWM จากฟังก์ชัน `analogWrite`

ฟังก์ชันที่ส่งสัญญาณออกยังขาเหล่านี้ใช้คำสั่ง `analogWrite(3, 100);` หมายถึง ส่งสัญญาณ PWM ออกจากขา D3 โดยให้สัญญาณที่ส่งออกนั้นขนาดตัวตึ้ไซเคิลที่ 100

ฟังก์ชัน Arduino ที่ใช้งานในใบงานการทดลอง

1. ฟังก์ชันกำหนดโหมดการทำงานให้กับขาพอร์ต โดยสามารถกำหนดได้ทั้งขาดิจิทัลโดยใส่เพียงตัวเลขของขา (0, 1, 2,...13) และขาแอนาล็อกที่ต้องการให้ทำงานในโหมดดิจิทัลแต่ การใส่ขาต้องใส่ A นำ หน้าซึ่งใช้ได้เฉพาะ A0, A1,...A5 ส่วนขา A6 และ A7 ไม่สามารถใช้งานในโหมดดิจิทัลได้ รูปแบบของฟังก์ชันเป็นดังนี้

	สาขาวิชา	อิเล็กทรอนิกส์	ใบงานการทดลองที่ 5	
	ชื่อวิชา	ไมโครคอนโทรลเลอร์		
	รหัสวิชา	2105-2105	หน้าที่	
	ชื่องาน	งานโปรแกรมอ่านค่าจากพอร์ตแอนาล็อกและการใช้ PWM	35	

`pinMode(pin,mode);`

pin : หมายเลขขาที่ต้องการเซตโหมด, mode : INPUT, OUTPUT, INPUT_PULLUP

2. ฟังก์ชันส่งค่าลอจิกดิจิทัลไปยังขาพอร์ต ค่า HIGH เป็นการส่งลอจิก 1 และค่า LOW เป็นการส่งลอจิก 0 ออกไปยังขาพอร์ต ฟังก์ชันนี้จะทำงานได้ต้องมีการใช้ฟังก์ชัน `pinMode` ก่อน รูปแบบของฟังก์ชันเป็นดังนี้

`digitalWrite(pin,value);`

pin : หมายเลขขาที่ต้องการเขียนลอจิกออกพอร์ต ,value : HIGH หรือ LOW

3. ฟังก์ชันอ่านค่าลอจิกดิจิทัลที่ขาพอร์ต เป็นการอ่านค่าเข้ามาซึ่งอาจนำมาเก็บไว้ในตัวแปรไว้ตรวจสอบลอจิกที่หลังหรือจะตรวจสอบลอจิกแบบทันทีก็ได้ ฟังก์ชันนี้จะทำงานได้ต้องมี การใช้ฟังก์ชัน `pinMode` ก่อน รูปแบบของฟังก์ชันเป็นดังนี้

`digitalRead(PIN);` pin : หมายเลขขาพอร์ตที่ต้องการอ่านลอจิก

ตัวอย่างเช่น `value=digitalRead(2);` หมายถึง อ่านค่าลอจิกที่ขา D2 มาเก็บไว้ในตัวแปร `value` `if(digitalRead(2)==LOW)` หมายถึง ตรวจสอบขา D2 ว่าเป็นลอจิก 0 หรือไม่

4. ฟังก์ชันหน่วงเวลาหรือฟังก์ชันหยุดค้าง การใช้งานสามารถกำหนดตัวเลขของเวลาที่ ต้องการหยุดค้าง ตัวเลขที่ใส่เป็นตัวเลขของเวลาหน่วยเป็นมิลลิวินาที ตัวเลขของเวลาที่ใส่ ได้สูงสุดคือ 4,294,967,295 ซึ่งเป็นขนาดของตัวแปร `unsigned long` รูปแบบของฟังก์ชันเป็นดังนี้

`Delay(ms);` ms : ตัวเลขที่หยุดค้างของเวลาหน่วยมิลลิวินาที (`unsigned long`)

5. ฟังก์ชันกำหนดความเร็วในการสื่อสารทางพอร์ตอนุกรม รูปแบบของฟังก์ชันเป็นดังนี้


`Serial.begin(speed);` speed: ตัวเลขของอัตราเร็วในการสื่อสารผ่านพอร์ตอนุกรม

6. ฟังก์ชันส่งข้อมูลออกพอร์ต เป็นฟังก์ชันที่ใช้ในการส่งข้อมูลออกทางพอร์ตอนุกรมหรือพิมพ์ข้อมูลออกทางพอร์ตเพื่อแสดงผลที่จอคอมพิวเตอร์ เมื่อพิมพ์เสร็จตัวเคอร์เซอร์จะรออยู่ที่ท้ายสิ่งพิมพ์นั้น ๆ รูปแบบของฟังก์ชันเป็นดังนี้

`Serial.print(val); Serial.print(val, format);`

7. ฟังก์ชันส่งข้อมูลออกพอร์ต คล้ายกับฟังก์ชัน `Serial.print` ต่างกันตรงที่เมื่อพิมพ์เสร็จตัวเคอร์เซอร์จะขึ้นบรรทัดใหม่ ดังนั้นเมื่อส่งพิมพ์ครั้งถัดไปข้อมูลปรากฏจะอยู่ที่บรรทัดใหม่ แทนที่จะต่อท้ายเหมือนกับฟังก์ชัน `Serial.print` รูปแบบของฟังก์ชันเป็นดังนี้

`Serial.println(val); Serial.println(val, format);`

	สาขาวิชา	อิเล็กทรอนิกส์	ใบงานการทดลองที่ 5
	ชื่อวิชา	ไมโครคอนโทรลเลอร์	
	รหัสวิชา	2105-2105	หน้าที่
	ชื่องาน	งานโปรแกรมอ่านค่าจากพอร์ตแอนาล็อกและการใช้ PWM	36

8. ฟังก์ชันให้ขาพอร์ตส่งสัญญาณ PWM เป็นฟังก์ชันที่ให้ขาพอร์ตดิจิทัลขา 3,5,6,9,10 และ 11 (ขาที่ส่งสัญญาณ PWM ได้) ส่งสัญญาณ PWM ออกตามค่าตัวเลขที่เลือกที่กำหนดด้วยความถี่ 490 Hz

`analogWrite(pin, value);`

pin: ขาพอร์ตดิจิทัลที่ต้องการส่งสัญญาณ PWM , value: ค่าตัวเลขที่เลือกที่อยู่ระหว่าง 0 ถึง 255 ตัวอย่างเช่น `analogWrite(3, 100);` หมายถึง ส่งสัญญาณ PWM ออกทางขา D3 โดยให้สัญญาณที่ส่งออกนั้นขนาดตัวเลขที่เลือกที่ 100 (ประมาณ 39.216 %)

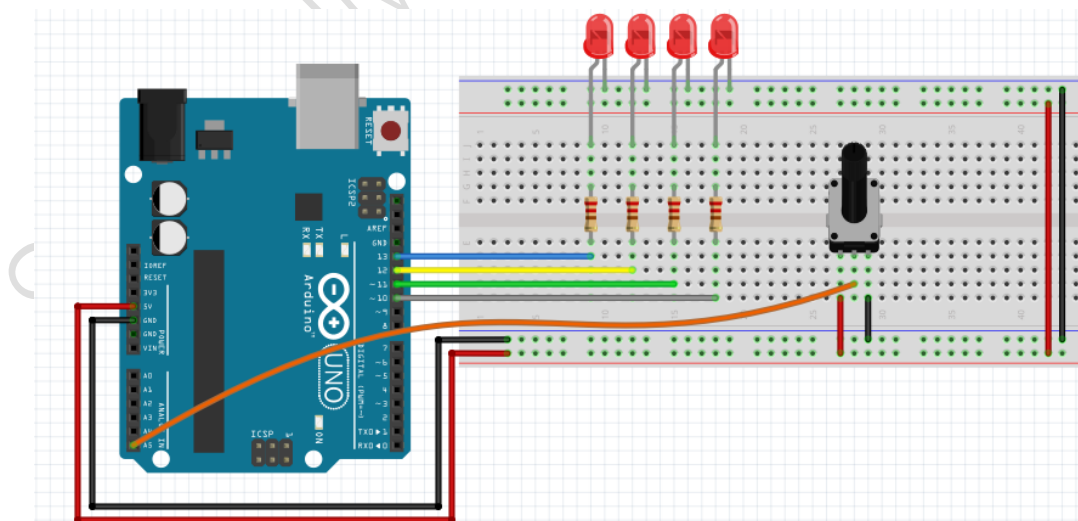
[ที่มา:ครูประภาส สุวรรณเพชร,เอกสารประกอบการอบรม เรียนรู้และลองเล่น Arduino เบื้องต้น (ฉบับปรับปรุงครั้งที่ 1) ,หน้าที่ 94-97.]

ลำดับขั้นตอนการทดลอง


ตอนที่ 1 โปรแกรมการควบคุมความสว่าง LED ด้วยโพเทนติโอมิเตอร์

แนวคิดการเรียนรู้ คือ เขียนโปรแกรมควบคุมระดับความสว่างของ LED ที่ต่ออยู่ที่ขาพอร์ตดิจิทัล D10 ซึ่งเป็นขา ที่สามารถส่งสัญญาณ PWM ออกมาได้ด้วยคำสั่ง `analogWrite(pin,value)` โดยควบคุมความสว่างด้วย โพเทนติโอมิเตอร์ที่ต่ออยู่ที่ขาพอร์ตแอนาล็อก A5 ค่าที่อ่านได้จะมีค่าอยู่ระหว่าง 0-1023 แต่ค่าที่ใช้ ควบคุมตัวเลขที่เลือกอยู่ระหว่าง 0-255 การดำเนินการลดค่าให้ใช้ค่าที่อ่านได้หารด้วย 4 ก่อนเอาไปใช้งาน โดยมีขั้นตอนดังนี้

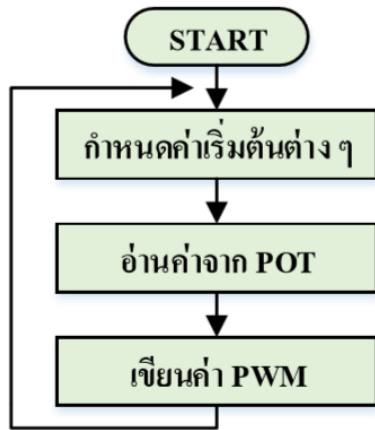
1. ประกอบวงจรการควบคุม LED 2 ตัว ด้วยสวิทซ์ 4 ตัวโดยใช้บอร์ด Arduino UNO R3 ดังรูปที่ 4.1



รูปที่ 5.1 แสดงการต่อวงจรควบคุมความสว่าง LED ด้วยโพเทนติโอมิเตอร์

	สาขาวิชา	อิเล็กทรอนิกส์	ใบงานการทดลองที่ 5
	ชื่อวิชา	ไมโครคอนโทรลเลอร์	
	รหัสวิชา	2105-2105	หน้าที่
	ชื่องาน	งานโปรแกรมอ่านค่าจากพอร์ตแอนาล็อกและการใช้ PWM	37

- เปิดโปรแกรม Arduino IDE จากนั้นพิมพ์โค้ดโปรแกรมการควบคุมความสว่าง LED ด้วยโพเทนติโอมิเตอร์ตามรูปที่ 5.2 ดังต่อไปนี้



(ก) ผังงาน

```

Lab5-1 §
1 #define LED 10
2 #define POT 5
3 void setup()
4 {
5   pinMode(LED, OUTPUT);
6   Serial.begin(9600);
7 }
8 void loop()
9 {
10  int adc=analogRead(POT);
11  analogWrite(LED, adc/4);
12 }
  
```

(ข) โค้ดโปรแกรม

รูปที่ 5.2 แสดงโปรแกรมการควบคุมความสว่าง LED ด้วยโพเทนติโอมิเตอร์

- บันทึกไฟล์โค้ด ชื่อ Lab5-1
- ทำการ Compile โค้ด Lab5-1
- เชื่อมต่อสาย USB กับ บอร์ด Arduino Uno R3
- Upload โปรแกรม Lab5-1 ลงบอร์ด Arduino UNO R3
- สังเกตวงจรการทำงานและบันทึกผลการทดลอง

.....

.....

.....


.....

.....

.....

.....

.....

	สาขาวิชา	อิเล็กทรอนิกส์	ใบงานการทดลองที่ 5
	ชื่อวิชา	ไมโครคอนโทรลเลอร์	
	รหัสวิชา	2105-2105	หน้าที่
	ชื่องาน	งานโปรแกรมอ่านค่าจากพอร์ตแอนาล็อกและการใช้ PWM	38

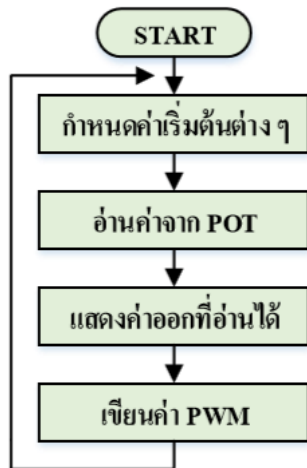
8. คำถามท้ายการทดลองตอนที่ 1 จากโค้ดโปรแกรม Lab5-1 จงตอบคำถามต่อไปนี้

- 8.1. บรรทัดที่ 2 ทำหน้าที่.....
- 8.2. บรรทัดที่ 5 ทำหน้าที่.....
- 8.3. บรรทัดที่ 6 ทำหน้าที่.....
- 8.4. บรรทัดที่ 10 ทำหน้าที่.....
- 8.5. บรรทัดที่ 11 ทำหน้าที่.....

ตอนที่ 2 โปรแกรมการควบคุม LED 2 ตัว ด้วยโพเทนติโอมิเตอร์

แนวคิดการเรียนรู้ คือ เขียนโปรแกรมควบคุมความสว่างแก่ LED จำนวน 2 ตัวที่อยู่ขา D10, D11 ซึ่งเป็นขาที่สามารถส่งสัญญาณ PWM ได้ โดยใช้โพเทนติโอมิเตอร์เป็นตัวปรับความสว่างโดยให้ LED ทั้งสองตัวมีความสว่างที่ตรงข้ามกัน โดยให้มีการแสดงค่าสัญญาณแอนาล็อกที่อ่านได้จากขาที่ต่อโพเทนติโอมิเตอร์ ออกทางพอร์ตอนุกรมแสดงผลที่จอคอมพิวเตอร์โดยมีขั้นตอนดังนี้

9. พิมพ์โค้ดโปรแกรมการควบคุม LED 2 ตัว ด้วยโพเทนติโอมิเตอร์ตามรูปที่ 5.3 ดังต่อไปนี้



(ก) ผังงาน

```

Lab5-2 §
1 #define LED1 10
2 #define LED2 11
3 #define POT 5
4 void setup()
5 {
6   pinMode(LED1, OUTPUT);
7   pinMode(LED2, OUTPUT);
8   Serial.begin(9600);
9 }
10 void loop()
11 {
12   int adc=analogRead(POT);
13   Serial.print("Analog Read value : ");
14   Serial.print(adc);
15   adc=map(adc, 0, 1023, 0, 255);
16   Serial.print(" map value to : ");
17   Serial.println(adc);
18   analogWrite(LED1, adc);
19   analogWrite(LED2, 255-adc);
20   delay(500);
21 }
  
```

(ข) โค้ดโปรแกรมการควบคุม LED 3 ตัว ด้วยสวิตช์ 4 ตัว

รูปที่ 5.3 แสดงการเขียนโปรแกรมการควบคุม LED 3 ตัว ด้วยสวิตช์ 4 ตัว

