

แผนการจัดการเรียนรู้มุ่งเน้นสมรรถนะ

ชื่อวิชา พื้นฐานไมโครคอนโทรลเลอร์ รหัสวิชา 20127-2018 ทฤษฎี 1 ปฏิบัติ 3 หน่วยกิต 2 ✓ หลักสูตรประกาศนียบัตรวิชาชีพ □ หลักสูตรประกาศนียบัตรวิชาชีพชั้นสูง ประเภทวิชา อุตสาหกรรม สาขาวิชา เมคคาทรอนิกส์ สาขางาน เมคคาทรอนิกส์

จัดทำโดย

นายวิรุณ จิตต์บุญ

วิทยาลัยเทคนิคชลบุรี สำนักงานคณะกรรมการการอาชีวศึกษา



หลักสูตรรายวิชา

สาขางาน เมคคาทรอนิกส์

จุดประสงค์รายวิชา

- 1. เข้าใจโครงสร้างและหลักการทำงานของไมโครคอนโทรลเลอร์
- 2. สามารถต่อไมโครคอนโทรลเลอร์กับอุปกรณ์ภายนอก
- 3. สามารถเขียน ทดสอบโปรแกรมควบคุมด้วยไมโครคอนโทรลเลอร์
- 4. มีเจตคติและกิจนิสัยที่ดีในการทำงานด้วยความรับผิดชอบ รอบคอบปลอดภัย

สมรรถนะรายวิชา

- 1. แสดงความรู้เกี่ยวกับโครงสร้างและหลักการทำงานของไมโครคอนโทรลเลอร์
- 2. ต่อวงจรไมโครคอนโทรลเลอร์ควบคุมอุปกรณ์ภายนอก
- 3. เขียนโปรแกรมควบคุมด้วยไมโครคอนโทรลเลอร์ถูกต้องตามหลักการ
- 4. ทดสอบโปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์

คำอธิบายรายวิชา

ศึกษาและปฏิบัติเกี่ยวกับโครงสร้างและหน้าที่ส่วนต่าง ๆ ของไมโครคอนโทรลเลอร์ ชุดคำสั่ง การ เขียน โปรแกรมควบคุม การอ่านค่าแอนะล็อก การเชื่อมต่อผ่านพอร์ตอนุกรม I2C การเชื่อมต่อ ไมโครคอนโทรลเลอร์กับ อุปกรณ์ภายนอกต่าง ๆ ควบคุมหลอดไฟ LED ควบคุมหลอดไฟ 7-Segment ควบ คุมสเต็ปเปอร์มอเตอร์ (Stepper Motor) ควบคุมมอเตอร์ แสดงผล LCD ไอซีวัดอุณหภูมิ โมดูลตรวจจับ สัญญาณอินพุต การใช้งานระบบ IoT เบื้องต้น

ใบงานที่ 1

ส่วนประกอบและการใช้งานบอร์ด Arduino

จุดประสงค์การเรียนรู้

- 1. ศึกษาส่วนประกอบและการใช้งานบอร์ดไมโครคอนโทรลเลอร์ Arduino
- 2. อธิบายส่วนประกอบและการใช้งานบอร์ดไมโครคอนโทรลเลอร์ Arduino ได้

เครื่องมือและอุปกรณ์การทดลอง

- 1. เครื่องไมโครคอมพิวเตอร์
- 2. บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3
- 3. โปรแกรมการทดลอง
- 4. อุปกรณ์อิเล็กทรอนิกส์สำหรับทดลอง

แนะนำบอร์ดไมโครคอนโทรลเลอร์ Arduino

บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 เป็นบอร์ดวงจรไมโครคอนโทรลเลอร์ที่ใช้ไอซี ประมวลผล เบอร์ ATmega328 ออกแบบวงจรให้มีขาสัญญาณอินพุตหรือเอาต์พุตแบบดิจิตอลจำนวน 14 ขา และยังสามารถ ส่งสัญญาณเอาต์พุตเป็นแบบ PWM ได้จำนวน 6 ขา ขาสัญญาณอินพุตแบบ อนาล็อกมี 6 ขา ใช้วงจรกำเนิด สัญญาณนาฬิกามีความถี่ 16 เมกะเฮิรตซ์ มีพอร์ตเชื่อมต่อ USB, มีขั้ว ต่อสัญญาณแบบ ICSP และมีปุ่มรีเซต บอร์ด Arduino Uno เป็นบอร์ดที่สนับสนุนระบบไมโคร คอนโทรลเลอร์ ซึ่งสามารถเชื่อมต่อเข้ากับเครื่อง คอมพิวเตอร์ด้วยสาย USB หรือจ่ายไฟจากอะแดปเตอร์ หรือแบตเตอรี่



รูปที่ 1.8 วงจร PCB บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3

วิทยาลัยเทคนิคชลบุรี

การปฏิบัติงาน

1. จากวงจรรูปที่ 1.9 จงเขียนอธิบายส่วนประกอบของบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3



รูปที่ 1.9 ส่วนประกอบบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3

ใบงานที่ 10

โปรแกรม Arduino ควบคุมมอเตอร์ไฟฟ้ากระแสตรง

จุดประสงค์การเรียนรู้

- 1. ศึกษาการทำงานของโปรแกรม Arduino กับมอเตอร์ไฟฟ้ากระแสตรง
- 2. สามารถเขียนโปรแกรม Arduino เพื่อควบคุมมอเตอร์ไฟฟ้ากระแสตรงได้
- 3. อธิบายโปรแกรม Arduino เพื่อควบคุมมอเตอร์ไฟฟ้ากระแสตรงได้

เครื่องมือและอุปกรณ์การทดลอง

- 1. เครื่องไมโครคอมพิวเตอร์
- 2. บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3
- 3. โปรแกรมการทดลอง
- 4. อุปกรณ์อิเล็กทรอนิกส์สำหรับทดลอง

การทดลอง

โปรแกรมที่ 1 โปรแกรมควบคุมความเร็วมอเตอร์กระแสตรง

ประกอบวงจรตามรูปที่ 10.6 เป็นวงจรควบคุมความเร็วมอเตอร์ไฟฟ้ากระแสตรงด้วย Arduino โดยใช้ ทรานซิสเตอร์ชนิด NPN เบอร์ TIP120 โดยขาเบส (B) ของทรานซิสเตอร์นำมาต่อที่พอร์ตดิจิตอล ขา 2 ขาคอลเลกเตอร์ (C) ต่อกับมอเตอร์ และขาอีมิตเตอร์ (E) ต่อลงกราวนด์ ส่วนอินพุตต่อสวิตช์ SW1 กับพอร์ต ดิจิตอลที่ขา 3



รูปที่ 10.6 วงจรควบคุมความเร็วมอเตอร์กระแสตรงด้วย Arduino

รหัสวิชา 20127-2018	67 วิชา พื้นฐานไมโครคอนโทรลเลอร์
โปรแกรมการทดลอง	
int push Button = 3;	// กำหนดตัวแปร pushButton ต่อที่พอร์ตดิจิตอลขา 3
int motorControl = 2;	// กำหนดตัวแปร motorControl ต่อที่พอร์ตดิจิตอลขา 2
void setup() {	
pinMode(pushButton, INPUT);	//เซตพอร์ตดิจิตอลขา 3 ให้เป็นพอร์ตอินพุต .
pinMode(motorControl, OUTPUT);	//เซตพอร์ตดิจิตอลขา 2 ให้เป็นพอร์ตเอาต์พุต
}	
void loop() {	
if(digitalRead(pushButton) == LOW) {	// รอการกดสวิตช์ pushButton
for(int x = 0; x <= 255; x++) {	// กำหนดตัวแปร x มีค่า 0-255 เพิ่มค่า x คลั้งละ 1
analogWrite(motorCo	ntrol, x); // มอเตอร์หมุนจากช้าไปเร็ว
delay(50);	// หน่วงเวลา 0.05 วินาที
}	
for(int x = 255; x >= 0; x) {	// กำหนดตัวแปร x มีค่า 0-255 ลดค่า x คลั้งละ 1
analogWrite(motorCor	ntrol, X); // มอเตอร์หมุนจากเร็วมาช้า
delay(50);	// หน่วงเวลา 0.05 วินาที
}	
}	
delay(1);	// หน่วงเวลา 0.001 วินาที
}	

ผลการทดลอง

การทดลอง	อธิบายลักษณะการทำงานของมอเตอร์
ให้ทำการกดสวิตซ์ SW1	

โปรแกรมที่ 2 โปรแกรมควบคุมความเร็วมอเตอร์กระแสตรงด้วยไอซี L293D

ประกอบวงจรตามรูปที่ 10.7 เป็นวงจรการต่อใช้งาน Arduino กับไอซีควบคุมมอเตอร์เบอร์ L293D ที่ พอร์ตดิจิตอลขา 4 และ 5 ต่อที่ขาอินพุตขา IN1 และ IN2 ใช้สำหรับควบคุมทิศทางการหมุน ของมอเตอร์ ส่วน พอร์ตดิจิตอลขา 3 ต่อเข้าที่ขา EN1 ใช้สำหรับการปรับความเร็วของมอเตอร์ ขา 16 ต่อไฟเลี้ยง +5 โวลต์ร่วมกับ ไฟเลี้ยง Arduino ขา 8 ต่อไฟเลี้ยงของมอเตอร์ เช่น มอเตอร์ใช้แรงดัน ไฟฟ้า +6 โวลต์ VS นำไปต่อแรงดันไฟฟ้า +6 โวลต์ ขา 4, 5, 12, 13 ต่อลงกราวนด์ และขา OUT1, OUT2 ต่อกับมอเตอร์



รูปที่ 10.7 วงจรควบคุมความเร็วมอเตอร์กระแสตรงด้วยไอซี 12930

โปรแกรมการทดลอง

int speedPin = 3; int motor1APin = 4;

int motor2APin = 5;

int speed_value_motor1;

void setup() {

pinMode(speedPin, OUTPUT); pinMode(motor1APin, OUTPUT); pinMode(motor2APin, OUTPUT); // กำหนดตัวแปร speedPin ต่อที่พอร์ตดิจิตอลขา 3
 // กำหนดตัวแปร motor1APin ต่อที่พอร์ตดิจิตอลขา 4
 // กำหนดตัวแปร motor2APin ต่อที่พอร์ตดิจิตอลขา 5
 // กำหนดตัวแปร speed_value_motor1

// เซตพอร์ตดิจิตอลขา 3 ให้เป็นพอร์ตเอาต์พุต
// เซตพอร์ตดิจิตอลขา 4 ให้เป็นพอร์ตเอาต์พุต
// เซตพอร์ตดิจิตอลขา 5 ให้เป็นพอร์ตเอาต์พุต

}

void loop() {

digitalWrite(motor1APin, LOW);	// ส่งข้อมูลออกเอาต์พุตเป็นลอจิก "0"
digitalWrite(motor2APin, HIGH);	// ส่งข้อมูลออกเอาต์พุตเป็นลอจิก "1"
<pre>speed_value_motor1 = 200;</pre>	// กำหนดความเร็วมอเตอร์มีค่าตั้งแต่ 0 - 255
analogWrite(speedPin, speed_value_	_motor1); // มอเตอร์หมุนตามความเร็วที่กำหนด

}

หลังจากบันทึกผลการทดลองในตารางบันข	กึกผลช่องแรกเสร็จแล้ว จากคำสั่งบรรทัด
digitalWrite(motor1APin, LOW);	// ส่งข้อมูลออกเอาต์พุตเป็นลอจิก "0"
digitalWrite(motor2APin, HIGH);	// ส่งข้อมูลออกเอาต์พุตเป็นลอจิก" 1 "

ผลการทดลอง

คำสั่ง	สังเกตทิศทางการหมุน ของมอเตอร์	สังเกตความเร็วของมอเตอร์
digitalWrite(motor1APin, LOW);		
digitalWrite(motor2APin, HIGH);		
digitalWrite(motor1APin, HIGH);		
digitalWrite(motor2APin, LOW);		
digitalWrite(motor1APin, LOW);		
digitalWrite(motor2APin, LOW);		
digitalWrite(motor1APin, HIGH);		
digitalWrite(motor2APin, HIGH);		

จากโปรแกรมที่ 1 ในคำสั่ง speed value motor1 = 200; ให้เปลี่ยนค่า PWM ตามตาราง ที่กำหนดให้

ผลการทดลอง

speed_value_motor1 = 200;	สังเกตความเร็วของมอเตอร์
0	
64	
127	
191	
255	

สรุปผลการทดลอง

กิจกรรมตรวจสอบความเข้าใจใบงานที่ 10

1. จงเขียนโปรแกรมตามโจทย์ที่กำหนดให้คือ

จากวงจรรูปที่ 10.8 ให้เขียนโปรแกรมการรับค่าสวิตซ์และควบคุมมอเตอร์ไฟฟ้ากระแสตรง โดยมีเงื่อนไข

ดังนี้

- ถ้ากดสวิตช์ SW1 กำหนดให้มอเตอร์หมุนด้วยความเร็ว PWM = 0
- ถ้ากดสวิตซ์ SW2 กำหนดให้มอเตอร์หมุนด้วยความเร็ว PWM = 85
- ถ้ากดสวิตซ์ SW3 กำหนดให้มอเตอร์หมุนด้วยความเร็ว PWM = 170
- ถ้ากดสวิตซ์ SW4 กำหนดให้มอเตอร์หมุนด้วยความเร็ว PWM = 255



รูปที่ 10.8 วงจรรับค่าสวิตซ์และควบคุมมอเตอร์กระแสตรงด้วยไอซี L293D

โปรแกรม

ใบงานที่ 2

การติดตั้งและการใช้งานบอร์ด Arduino

จุดประสงค์การเรียนรู้

1. ศึกษาการติดตั้งโปรแกรม Arduino (DE)

2. อธิบายการเขียนโปรแกรมและการอัปโหลดโปรแกรมได้

เครื่องมือและอุปกรณ์การทดลอง

- 1. เครื่องไมโครคอมพิวเตอร์
- 2. บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3
- 3. โปรแกรมการทดลอง
- 4. อุปกรณ์อิเล็กทรอนิกส์สำหรับทดลอง

การปฏิบัติงาน

1. ให้นักเรียนแบ่งกลุ่มฝึกติดตั้ง Arduino (IDE) พร้อมอธิบายขั้นตอนการติดตั้งโปรแกรม ดังกล่าว

 ให้นักเรียนเขียนโปรแกรมเพื่อทดสอบการทำงานลงในโปรแกรม Arduino (IDE) จากนั้น อัปโหลดโปรแกรมที่ เขียนนี้ลงบอร์ด Arduino จากนั้นคลิกปุ่ม Serial Monitor เพื่อดูผลการทดลอง

void setup() {

Serial.begin(9600);

}

void loop() {

Serial.println("Welcome to Arduino");

Serial.println("Congratulations, you've done a great job!");

while(1);

}

บันทึกผลการทดลอง

สรุปผลการทดลอง

ใบงานที่ 3

คำสั่งภาษาซี Arduino

จุดประสงค์การเรียนรู้

- 1. ศึกษาคำสั่งภาษาซีของ Arduino
- 2. อธิบายคำสั่งภาษาซีของ Arduino ได้

เครื่องมือและอุปกรณ์การทดลอง

- 1. เครื่องไมโครคอมพิวเตอร์
- 2. บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3
- 3. โปรแกรมการทดลอง
- 4. อุปกรณ์อิเล็กทรอนิกส์สำหรับทดลอง

การปฏิบัติงาน

 จงเขียนโปรแกรมไมโครคอนโทรลเลอร์ Arduino ตามโปรแกรมตัวอย่างที่ 1 เมื่อเขียน โปรแกรมเสร็จ ให้ทำการคอมไพล์และอัปโหลดโปรแกรมลงบอร์ด Arduino ให้สังเกตหลอด LED ที่ ต่อกับพอร์ตดิจิตอล ขาที่ 13 แล้วบันทึกผลการทดลอง



รูปที่ 3.3 วงจรไมโครคอนโทรลเลอร์ Arduino

```
โปรแกรมตัวอย่างที่ 1

const int ledin = 13,

void setup()

{

    pinMode(ledin, OUTPUT);

}

void loop()

{

    digitalWrite(ledin, HIGH);

    delay(1000);

    digitalWrite(ledin, LOW);

    delay(1000);

}

ผลการทดลอง
```

จงเขียนโปรแกรมไมโครคอนโทรลเลอร์ Arduino ตามโปรแกรมตัวอย่างที่ 2 และต่อวงจร ตามรูปที่
 3.4 เมื่อเขียนโปรแกรมเสร็จให้ทำการคอมไพล์และอัปโหลดโปรแกรมลงบอร์ด Arduino ให้ สังเกตหลอด LED1 ที่
 ต่อกับพอร์ตดิจิตอล ขาที่ 1 แล้วบันทึกผลการทดลอง



รูปที่ 3.4 วงจรไมโครคอนโทรลเลอร์ Arduino

```
โปรแกรมตัวอย่างที่ 2
const int led1 = 1;
void setup()
{
   pinMode(led1, OUTPUT);
}
void loop()
{
   digitalWrite(led1, HIGH);
   delay(500);
   digitalWrite(led1, LOW);
   delay(500);
}
ผลการทดลอง
_____
สรุปผลการทดลอง
.....
```

ใบงานที่ 4 ใบงานที่ 4.1 โปรแกรมไฟกะพริบ

จุดประสงค์การเรียนรู้

- 1. ศึกษาการทำงานของโปรแกรมไฟกะพริบ
- 2. สามารถเขียนโปรแกรมไฟกะพริบรูปแบบต่าง ๆ ได้

เครื่องมือและอุปกรณ์การทดลอง

- 1. เครื่องไมโครคอมพิวเตอร์
- 2. บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3
- 3. โปรแกรมการทดลอง
- 4. อุปกรณ์อิเล็กทรอนิกส์สำหรับทดลอง

การทดลอง

โปรแกรมที่ 1 โปรแกรมไฟกะพริบ 1 ดวงรูปแบบที่ 1

ในการทดลองเขียนโปรแกรมไฟกะพริบ 1 ดวงนี้ นำเอาคำสั่งการควบคุม ได้แก่ คำสั่ง if, คำสั่ง false, คำสั่ง for และคำสั่ง while มาทดสอบการทำงานเพื่อให้เข้าใจการทำงานของคำสั่งเหล่านี้ มากขึ้น ให้ต่อวงจรตาม รูปที่ 4.7



รูปที่ 4.7 วงจรสำหรับโปรแกรมไฟกะพริบ

โปรแกรมไฟกะพริบ 1 ดวงรูปแบบนี้เป็นการนำคำสั่งควบคุม คือ คำสั่ง if เพื่อใช้ในการ ตรวจสอบเงื่อนไข ในโปรแกรมไฟกะพริบนี้

const	int Led1 = 0;	// กำหนดตัวแปร Led1 ต่อที่พอร์ตดิจิตอลขา 0
void s	etup()	
{		
	pinMode(Ledi, OUTPUT);	// เซตพอร์ตดิจิตอลขา 0 ให้เป็นพอร์ตเอาต์พุต
}		
int de	layTime = 1000;	// กำหนดตัวแปร delayTime เป็นตัวแปรชนิด int
		// มีค่าเท่ากับ 1000
void la	oop()	
{		
	delayTime = delayTime - 100;	// นำค่าใน delayTime - 100 ผลลัพธ์เก็บ
		// ใน delayTime
	if(delay Time <= 0) {	// ถ้าตัวแปร delayTime น้อยกว่าหรือเท่ากับ 0
	delayTime = 1000;	// ทำให้ตัวแปร delayTime มีค่าเท่ากับ 1000
}		
	digitalWrite(Led1, HIGH);	// ส่งข้อมูลออกเอาต์พุตเป็นลอจิก "1"
	delay(delayTime);	// หน่วงเวลาตามค่าในตัวแปร delayTime
	digitalWrite(Led1, LOW);	// ส่งข้อมูลออกเอาต์พุตเป็นลอจิก "0"
	delay(delayTime);	// หน่วงเวลาตามค่าในตัวแปร delayTime

9

ผลการทดลอง

สังเกตการทำงานของ	อธิบายลักษณะการแสดงผลของ LED1
LED1	

โปรแกรมที่ 2 โปรแกรมไฟกะพริบ 1 ดวงรูปแบบที่ 2

```
้โปรแกรมไฟกะพริบ 1 ดวงรูปแบบนี้เป็นการนำคำสั่งควบคุม คือ คำสั่ง if..else เพื่อใช้ในการ ตรวจสอบ
เงื่อนไขในโปรแกรมไฟกะพริบนี้
                                                     //กำหนดตัวแปร Led1 ต่อที่พอร์ตดิจิตอลขา 0
const int Ledi = 0;
void setup()
{
                                                     // เซตพอร์ตดิจิตอลขา 0 ให้เป็นพอร์ตเอาต์พุต
       pinMode(Led1, OUTPUT);
}
                                                     // กำหนดตัวแปร delayTime เป็นตัวแปรชนิด int
int delayTime = 1000;
                                                     // มีค่าเท่ากับ 1000
void loop()
{
                                                     // ถ้าตัวแปร delayTime น้อยกว่าหรือเท่ากับ 100
       if (delayTime \leq 100) {
                                                     // ทำให้ตัวแปร delayTime มีค่าเท่ากับ 1000
               delayTime = 1000;
       }
else {
                                                     // นำค่าใน delayTime - 100 ผลลัพธ์เก็บ
       delayTime = delay Time - 100;
                                                     // ใน delayTime
       }
                                                     // ส่งข้อมูลออกเอาต์พุตเป็นลอจิก "1"
       digitalWrite(Led1, HIGH);
                                                     // หน่วงเวลาตามค่าในตัวแปร delayTime
       delay(delayTime);
                                                     // ส่งข้อมูลออกเอาต์พุตเป็นลอจิก "0"
       digitalWrite(Led1, LOW);
                                                     // หน่วงเวลาตามค่าในตัวแปร delayTime
       delay(delayTime);
```

}

ผลการทดลอง

สังเกตการทำงานของ	อธิบายลักษณะการแสดงผลของ LED1
LED1	

วิชา พื้นฐานไมโครคอนโทรลเลอร์

```
โปรแกรมที่ 3 โปรแกรมไฟกะพริบ 1 ดวงรูปแบบที่ 3
โปรแกรมไฟกะพริบ 1 ดวงรูปแบบนี้เป็นการนำคำสั่งควบคุม คือ คำสั่ง for เพื่อใช้ในการ ตรวจสอบเงื่อนไขใน
โปรแกรมไฟกะพริบนี้
                                                      //กำหนดตัวแปร Led1 ต่อที่พอร์ตดิจิตอลขา 0
const int Led1 = 0;
void setup()
{
                                                      // เซตพอร์ตดิจิตอลขา 0 ให้เป็นพอร์ตเอาต์พุต
        pinMode(Led1, OUTPUT);
}
void loop()
{
                                      //ตัวแปร i=0; ถ้า <4 วนอยู่ในลูป for ; เพิ่ม 1 ครั้งละ 1/รอบ
        for (int i = 0; i < 4, i++) {
                                                      // ส่งข้อมูลออกเอาต์พุตเป็นลอจิก "1"
               digitalWrite(Led1, HIGH);
                                                      // หน่วงเวลา 0.3 วินาที
               delay(300);
               digitalWrite(Led1, LOW);
                                                      // ส่งข้อมูลออกเอาต์พุตเป็นลอจิก "0"
                                                      // หน่วงเวลา 0.3 วินาที
               delay(300);
       }
                                                      // หน่วงเวลา 2 วินาที
        delay(2000);
```

11

```
}
```

ผลการทดลอง

สังเกตการทำงานของ	อธิบายลักษณะการแสดงผลของ LED1
LED1	

โปรแกรมที่ 4 โปรแกรมไฟกะพริบ 1 ดวงรูปแบบที่ 4

โปรแกรมไฟกะพริบ 1 ดวงรูปแบบนี้เป็นการนำคำสั่งควบคุม คือ คำสั่ง while เพื่อใช้ในการ ตรวจสอบเงื่อนไขใน โปรแกรมไฟกะพริบนี้

```
const int Led1 = 0;
```

// กำหนดตัวแปร Led1 ต่อที่พอร์ตดิจิตอลขา 0

วิทยาลัยเทคนิคชลบุรี

จัดทำโดย นายวิรุณ จิตต์บุญ

```
void setup()
{
                                                    // เซตพอร์ตดิจิตอลขา 0 ให้เป็นพอร์ตเอาต์พุต
       pinMode(Led1, OUTPUT);
}
                                             // ตัวแปร delayTime เป็นชนิด int มีค่าเท่ากับ 1000
int delayTime = 1000;
void loop()
{
                                             // ถ้า delayTime มากกว่า 0 ทำงานในลูป while นี้
       while (delayTime > 0) {
                                                    // ส่งข้อมูลออกเอาต์พุตเป็นลอจิก "1"
               digitalWrite(Led1, HIGH);
                                                    // หน่วงเวลาตามค่าในตัวแปร delayTime
               delay(delayTime);
                                                    // ส่งข้อมูลออกเอาต์พุตเป็นลอจิก "0"
               digitalWrite(Led 1, LOW);
                                                    // หน่วงเวลาตามค่าในตัวแปร delayTime
               delay(delayTime);
               delayTime = delayTime - 100;
       }
                                             // ถ้า delayTime น้อยกว่า 1000 ทำงานในลูป while นี้
       while (delayTime < 1000) {
       delayTime = delayTime + 100;
                                                    // ส่งข้อมูลออกเอาต์พุตเป็นลอจิก "1"
       digitalWrite(Led1, HIGH);
                                                    // หน่วงเวลาตามค่าในตัวแปร delayTime
       delay(delayTime);
                                                    // ส่งข้อมูลออกเอาต์พุตเป็นลอจิก "0"
       digitalWrite(Led1, LOW);
                                                     // หน่วงเวลาตามค่าในตัวแปร delayTime
       delay(delayTime);
       }
```

12

```
}
```

ผลการทดลอง

สังเกตการทำงานของ	อธิบายลักษณะการแสดงผลของ LED1
LED1	

สรุปผลการทดลอง	

กิจกรรมตรวจสอบความเข้าใจใบงานที่ 4.1

1. จากรูปที่ 4.7 จงเขียนโปรแกรมส่งสัญญาณรหัสมอร์ส SOS ดังนี้



รหัส S หลอดแอลอีดี กะพริบ 3 ครั้ง ช่วงแอลอีดีติด 0.15 วินาที ดับ 0.1 วินาที รหัส 0 หลอดแอลอีดี กะพริบ 3 ครั้ง ช่วงแอลอีดีติด 0.4 วินาที ดับ 0.1 วินาที โปรแกรม

ใบงานที่ 4.2 โปรแกรมไฟกะพริบ

จุดประสงค์การเรียนรู้

- 1. ศึกษาการทำงานของโปรแกรมไฟวิ่งหลายรูปแบบ
- 2. สามารถเขียนโปรแกรมไฟวิ่งหลายรูปแบบต่าง ๆ ได้

เครื่องมือและอุปกรณ์การทดลอง

- 1. เครื่องไมโครคอมพิวเตอร์
- 2. บอร์ตไมโครคอนโทรลเลอร์ Arduino Uno R3
- 3. โปรแกรมการทดลอง
- 4. อุปกรณ์อิเล็กทรอนิกส์สำหรับทดลอง

การทดลอง

โปรแกรมที่ 1 โปรแกรมไฟวิ่ง 4 ดวง

โปรแกรมนี้เป็นการเขียนโปรแกรมไฟวิ่ง 4 ดวง โดยนำบอร์ด Arduino มาต่อกับหลอดแอลอีดี จำนวน 4 ดวง เมื่อเริ่มโปรแกรมหลอดแอลอีดีติด 1 ดวง ติดนาน 0.2 วินาที จากนั้นหลอดแอลอีดีติด เพิ่มครั้งละ 1 ดวงจน ครบ 4 ดวง เมื่อครบแล้วหลอดแอลอีดีดับครั้งละ 1 ดวงจนดับหมด



รูปที่ 4.8 วงจรสำหรับโปรแกรมไฟวิ่ง 4 ดวง

```
//กำหนดตัวแปร ledCount มีค่าเท่ากับ 4 Const
const int ledCount = 4;
                                                      //กำหนดตัวแปร ledPins ต่อที่พอร์ตดิจิตอลขา 1-4
const int ledPins[ledCount] = {1,2,3,4};
void setup()
{
       for (int i = 0; i < ledCount; i++) { // ตัวแปร I=0; ถ้า |< ledCount; เพิ่ม 1 ครั้งละ 1/รอบ
                                                      // เซตพอร์ตติจิตอลขา 1-4 ให้เป็นพอร์ตเอาต์พุต
               pinMode(ledPins[i], OUTPUT);
       }
}
void loop()
{
                                              // ตัวแปร =0; ถ้า < ledCount; เพิ่ม ครั้งละ 1/รอบ
       for (int i = 0; i < ledCount; i++){
                                              // ส่งข้อมูลออกเอาต์พุตเป็นลอจิก "1" ตามตัวแปร 1
               digitalWrite(ledPinsl, HIGH);
                                              // หน่วงเวลา 0.2 วินาที
               delay(200);
       }
}
       for (int i = [edCount - 1; i>= 0; i-) { // ตัวแปร |=0; ถ้า >= (edCount; ลด 1 ครั้งละ 1/รอบ
               digitalWrite(ledPins[i], LOW); // ส่งข้อมูลออกเอาต์พุตเป็นลอจิก "0" ตามตัวแปร 1
                                              // หน่วงเวลา 0.2 วินาที
               delay(200);
       }
}
```

สังเกตการทำงานของ	อธิบายลักษณะการแสดงผลของ LED				
LED4-LED1	LED4	LED3	LED2	LED1	
ครั้งที่ 1	0	0	0	0	
ครั้งที่ 2	0	0	0	0	
ครั้งที่ 3	0	0	0	0	
ครั้งที่ 4	0	0	0	0	
ครั้งที่ 5	0	0	0	0	
ครั้งที่ 6	0	0	0	0	
ครั้งที่ 7	0	0	0	0	
ครั้งที่ 8	0	0	0	0	
ครั้งที่ 9	0	0	0	0	
ครั้งที่ 10	0	0	0	0	

ผลการทดลอง

โปรแกรมที่ 2 โปรแกรมไฟวิ่ง 8 ดวง

โปรแกรมนี้เป็นการนำหลอดแอลอีดีทั้งหมด 8 ตวง เชื่อมต่อกับบอร์ด Arduino ที่พอร์ตดิจิตอล ขา 1 ถึง 8 เมื่อเริ่มโปรแกรมหลอดแอลอีดีติดครั้งละ 1 ดวง จากพอร์ตดิจิตอล ขา 1, 2, 3, ไปหา ขา 8 โดยใช้คาบเวลาใน การติตกะพริบครั้งละ 0.3 วินาที เมื่อหลอดแอลอีดีที่ขา 8 ติดสว่างแล้วก็ติด ย้อนกลับมายังขา 7, 6, 5,ขา 1 วน เช่นนี้ไปเรื่อย ๆ ตลอดทั้งโปรแกรม



รูปที่ 4.9 วงจรสำหรับโปรแกรมไฟวิ่ง 8 ดวง

```
// กำหนดตัวแปร ledCount มีค่าเท่ากับ 8
int ledCount = 8;
                                              // กำหนดตัวแปร ledPins ต่อที่พอร์ตดิจิตอลขา 1-3
int iledPins] = {1, 2, 3, 4, 5, 6, 7, 8, J;
                                                      // กำหนดตัวแปร (edDelay มีค่าเท่ากับ 300
int delayTime = 300;
void setup()
{
                                            // ตัวแปร I=0; ถ้า < ledCount; เพิ่ม 1 ครั้งละ 1/รอบ
       for (int i = 0; i < ledCount; i++) {
                                                      // เซตพอร์ตดิจิตอลขา 1-8 ให้เป็นพอร์ตเอาต์พุต
               pinMode(ledPins[i], OUTPUT);
void loop()
{
       for (int i = 0; i < ledCount-1; i++) { // ตัวแปร I-0; ถ้า < ledCount; เพิ่ม 1 ครั้งละ 1/รอบ
                                                      // ส่งข้อมูลออกเอาต์พุตเป็นลอจิก "1" ตามตัวแปร
               digitalWrite(ledPins[i], HIGH);
                                                      // หน่วงเวลาตามค่าในตัวแปร delayTime
               delay(delayTime);
               digitalWrite(ledPins[i, LOW); // ส่งข้อมูลออกเอาต์พุตเป็นลอจิก "0" ตามตัวแปร i
       }
       for (int i = ledCount-1; i > 0; i--) {
                                              // ส่งข้อมูลออกเอาต์พุตเป็นลอจิก "1" ตามตัวแปร i
        digitalWrite(ledPins[i], HIGH);
                                              // หน่วงเวลาตามค่าในตัวแปร delayTime
        delay(delayTime);
                                               // ส่งข้อมูลออกเอาต์พุตเป็นลอจิก "D" ตามตัวแปร
        digitalWrite(ledPins[i], LOW);
       }
```

```
}
```

ผลการทดลอง

สังเกตการทำงานของ			อธิบายส	้กษณะกา	รแสดงผลข	เอง LED		
LED8-LED1	LED8	LED7	LED6	LED5	LED4	LED3	LED2	LED1
ครั้งที่ 1	0	Ο	0	Ο	0	0	0	0
ครั้งที่ 2	Ο	Ο	Ο	Ο	Ο	0	Ο	0
ครั้งที่ 3	Ο	Ο	Ο	Ο	Ο	0	Ο	0
ครั้งที่ 4	Ο	Ο	0	Ο	0	Ο	Ο	0
ครั้งที่ 5	Ο	Ο	0	Ο	0	Ο	Ο	0
ครั้งที่ 6	0	0	0	0	0	0	0	0

วิชา พื้นฐานไมโครคอนโทรลเลอร์

ครั้งที่ 7	Ο	Ο	0	Ο	0	0	0	0
ครั้งที่ 8	0	0	0	0	Ο	Ο	0	0
ครั้งที่ 9	0	Ο	Ο	Ο	Ο	Ο	Ο	0
ครั้งที่ 10	0	Ο	Ο	Ο	Ο	Ο	Ο	0
ครั้งที่ 11	0	Ο	Ο	Ο	Ο	Ο	Ο	0
ครั้งที่ 12	0	Ο	Ο	Ο	Ο	Ο	Ο	0
ครั้งที่ 13	0	Ο	Ο	Ο	Ο	Ο	Ο	0
ครั้งที่ 14	Ο	Ο	Ο	Ο	Ο	Ο	Ο	0
ครั้งที่ 15	0	Ο	0	Ο	Ο	0	0	0
ครั้งที่ 16	0	Ο	0	Ο	Ο	0	0	0
ครั้งที่ 17	Ο	Ο	Ο	Ο	Ο	Ο	Ο	0
ครั้งที่ 18	0	Ο	0	Ο	Ο	Ο	0	Ο

สรุปผลการทดลอง

โปรแกรมที่ 3 โปรแกรมไฟจราจร

โปรแกรมนี้เป็นการเขียนโปรแกรมไฟจราจร โดยนำบอร์ด Arduino มาต่อกับหลอดแอลอีดี จำนวน 3 ดวง มีหลอดแอลอีดี สีแดง สีเหลือง และสีเขียว



รูปที่ 4.10 วงจรสำหรับโปรแกรมไฟจราจร

รหัสวิชา 20127-2018

19

วิชา พื้นฐานไมโครคอนโทรลเลอร์

const int redLED= 1; const int yellow LED= 2; const int greenLED= 3; // กำหนดตัวแปร redLED ต่อที่พอร์ตดิจิตอลขา 1
 // กำหนดตัวแปร yellowLED ต่อที่พอร์ตดิจิตอลขา 2
 // กำหนดตัวแปร greenLED ต่อที่พอร์ตดิจิตอลขา 3

```
void setup() {
```

pinMode (greenLED, OUTPUT); pinMode (yellowLED, OUTPUT); pinMode (redLED, OUTPUT); // เซตพอร์ตดิจิตอลขา 1-3 ให้เป็นพอร์ตเอาต์พุต

```
}
```

```
void loop() {
```

digitalWrite (greenLED, HIGH);	// หลอด LED สีเขียวติด
digitalWrite (yellowLED, LOW);	// หลอด LED สีเหลืองดับ
digitalWrite (redLED, LOW);	// หลอด LED สีแดงดับ
delay(10000);	// หน่วงเวลา 10 วินาที
digitalWrite (greenLED, LOW);	// หลอด LED สีเขียวดับ
digitalWrite (yellowLED, HIGH);	// หลอด LED สีเหลืองติด
digitalWrite (redLED, LOW);	// หลอด LED สีแดงดับ
delay(2000)	// หน่วงเวลา 2 วินาที
digitalWrite (greenLED, LOW);	// หลอด LED สีเขียวดับ
digitalWrite (yellowL.ED, LOW);	// หลอด LED สีเหลืองดับ
digitalWrite (redLED, HIGH);	// หลอด LED สีแดงติด

delay(10000);

ผลการทดลอง

สังเกตการทำงาน	อธิบายลักษณะการแสดงผลของ LED			เวลาการติด
ของ LED3-LED1	LED-GREEN	LED-YELLOW	LED-RED	
ครั้งที่ 1	0	0	0	

// หน่วงเวลา 10 วินาที

ครั้งที่ 2	Ο	Ο	Ο	
ครั้งที่ 3	О	0	0	
ครั้งที่ 4	О	0	0	
ครั้งที่ 5	Ο	0	0	
ครั้งที่ 6	Ο	0	0	

สรุปผลการทดลอง

กิจกรรมตรวจสอบความเข้าใจใบงานที่ 4.2

 จงออกแบบวงจร และเขียนโปรแกรมต่อไปนี้ จากรูปที่ 4.11 ให้ออกแบบวงจรไฟจราจร ซึ่งมีทั้งหมด 4 แยกโดยทำการปล่อยรถครั้งละแยก



รูปที่ 4.11 สี่แยกไฟจราจร

ออกแบบวงจร

โปรแกรม

ใบงานที่ 5

โปรแกรม Arduino กับพอร์ตดิจิตอลอินพุต

จุดประสงค์การเรียนรู้

- 1. ศึกษาการทำงานของโปรแกรม Arduino กับพอร์ตดิจิตอลอินพุต
- 2. สามารถเขียนโปรแกรม Arduino กับพอร์ตดิจิตอลอินพุตได้

เครื่องมือและอุปกรณ์การทดลอง

- 1. เครื่องไมโครคอมพิวเตอร์
- 2. บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3
- 3. โปรแกรมการทดลอง
- 4. อุปกรณ์อิเล็กทรอนิกส์สำหรับทดลอง

การทดลอง

โปรแกรมที่ 1 โปรแกรมสวิตช์กดติดปล่อยดับ

ประกอบวงจรตามรูปที่ 5.6 ลงบนบอร์ดทดลองไมโครคอนโทรลเลอร์ Arduino โดยต่อพอร์ต ดิจิตอลเป็น พอร์ตอินพุตที่ขา 2 กับสวิตซ์ SW1 และต่อพอร์ตดิจิตอลเป็นพอร์ตเอาต์พุตที่ขา 9 กับ ตัวต้านทาน R1 และหลอด LED1



รูปที่ 5.6 วงจรการทดลองโปรแกรมรับข้อมูลพอร์ตดิจิตอลอินพุต

โปรแกรมนี้เริ่มต้นด้วยการกำหนดพอร์ตดิจิตอลขา 2 ให้เป็นพอร์ตอินพุต และกำหนดพอร์ต ดิจิตอลขา 9 ให้เป็นพอร์ตเอาต์พุต โปรแกรมทำการตรวจสอบการกดสวิตซ์ SW1 เมื่อสวิตซ์ SW1 ถูก กดส่งผลให้ LED1 ติด สว่าง แต่เมื่อปล่อยสวิตซ์ SW1 ส่งผลให้ LED1 ดับทันที

วิชา พื้นฐานไมโครคอนโทรลเลอร์

const int Sw1 = 2;	// กำหนดตัวแปร SW1 ต่อที่พอร์ตดิจิตอลขา 2
const int Led1 = 9;	// กำหนดตัวแปร Led1 ต่อที่พอร์ตดิจิตอลขา 9
void setup() {	
pinMode(Sw1, INPUT);	// เซตพอร์ตดิจิตอลขา 2 ให้เป็นพอร์ตอินพุต
pinMode(Led1, OUTPUT);	// เซตพอร์ตดิจิตอลขา 9 ให้เป็นพอร์ตเอาต์พุต
}	

```
void loop() {
                                              // ถ้าอ่านค่าอินพุตตัวแปร Swi เท่ากับ 0
       if(digitalRead(Sw1) == LOW) {
               digitalWrite(Led1, HIGH);
                                              // LED1 ติดสว่าง
       }
                                              // แต่ค่าอินพุตตัวแปร Sw1 ไม่เท่ากับ 0
       else {
               digitalWrite(Led1, LOW);
                                              // LED1 ดับ
       }
```

```
}
```

ผลการทดลอง

รหัสวิชา 20127-2018

สังเกตการทำงาน	อธิบายลักษณะการแสดงผลของ LED1
เมื่อกด SW1	
เมื่อปล่อย SW1	

โปรแกรมที่ 2 โปรแกรมสวิตซ์กดติด - กดดับ

้โปรแกรมนี้เริ่มต้นด้วยการกำหนดพอร์ตดิจิตอลขา 2 ให้เป็นพอร์ตอินพุต และกำหนดพอร์ต ดิจิตอลขา 9 ให้เป็นพอร์ตเอาต์พุต โปรแกรมทำการตรวจสอบการกดสวิตซ์ SW1 เมื่อทำการกดและ ปล่อยสวิตซ์ SW1 ส่งผล ให้ LED1 ติดสว่าง และเมื่อทำการกดและปล่อยสวิตช์ครั้งที่ 2 LED1 ดับทันที

รหัสวิชา 20127-2018	24	วิชา พื้นฐานไมโครคอนโทรลเลอร์		
int Sw1 = 2;	// กำหนดตัวแบ	Jร Swl ต่อที่พอร์ตดิจิตอลขา 2		
int Led1 = 9;	// กำหนดตัวแบ	ls Led1 ต่อที่พอร์ตดิจิตอลขา 9		
boolean lastState;	// กำหนดตัวแปร lastSt	ate เพื่อเก็บสถานะเอาต์พุตที่ผ่านมา		
boolean reading;	// กำหนดตัวแปร readii	ng เพื่อเก็บค่าจากอินพุต		
boolean state = LOW;	// กำหนดตัวแปร state	มีค่าสถานะเป็น LOW		
void setup() {				
pinMode (Sw1,INPUT);	// เซตพอร์ตดิจิตอลขา 2 ให้เป็นพอร์ตอินพุต			
pinMode (Led1,OUTPUT);	//เซตพอร์ตดิจิตอลขา 9 ให้เป็นพอร์ตเอาต์พุต			
}				
void loop() {				
reading = digitalRead(Sw1);	// อ่านค่าจากพ	อร์ตขา 2 มาเก็บไว้ในตัวแปร reading		
if (reading == LOW && lastSta	ate == HIGH) {	// ตรวจสอบการกดและปล่อยสวิตช์		
delay (10);		// หน่วงเวลา 0.01 วินาที		
if(digitalRead(Sw1) ==	LOW) state = !state;	//ทำการกลับค่าตัวแปร state		
}				
digitalWrite(Led1,state);	//ส่งข้อมูลออกเ	อาต์พุตตามข้อมูลในตัวแปร state		
lastState = reading;	// นำค่าในตัวแร	// นำค่าในตัวแปร reading เก็บไว้ในตัวแปร lastState		

}

ผลการทดลอง

สังเกตการทำงาน การกดและปล่อย SW1	อธิบายลักษณะการแสดงผลของ LED1
ครั้งที่ 1	
ครั้งที่ 2	
ครั้งที่ 3	
ครั้งที่ 4	
ครั้งที่ 5	
ครั้งที่ 6	

สรุปผลการทดลอง

เปรียบเทียบการทำงานของโปรแกรมที่ 1 กับโปรแกรมที่ 2

 •••••	•••••	••••••	••••••	••••••
 ••••••	••••••	•••••••••••••••••••••••••••••	• • • • • • • • • • • • • • • • • • • •	••••••

โปรแกรมที่ 3 โปรแกรมรับค่าสวิตช์ 2 ตัว

ประกอบวงจรตามรูปที่ 5.7 ลงบนบอร์ดทดลองไมโครคอนโทรลเลอร์ Arduino โดยต่อพอร์ต ดิจิตอลเป็น พอร์ตอินพุตที่ขา 2 กับสวิตซ์ SW1 พอร์ตอินพุตที่ขา 3 กับสวิตซ์ SW2 และต่อพอร์ต ดิจิตอลเป็นพอร์ตเอาต์พุตที่ ขา 9 กับตัวต้านทาน R1 และหลอด LED1



รูปที่ 5.7 วงจรการทดลองโปรแกรมรับคาสวิตซ์ 2 ตัว

โปรแกรมนี้เริ่มต้นด้วยการกำหนดพอร์ตดิจิตอลขา 2, 3 ให้เป็นพอร์ตอินพุต และกำหนดพอร์ต ดิจิตอลขา 9 ให้เป็นพอร์ตเอาต์พุต โปรแกรมทำการตรวจสอบการกดสวิตซ์ SW1 เมื่อทำการกดและปล่อยสวิตซ์ SW1 ส่งผล ให้ LED1 ติดสว่าง และเมื่อทำการกดและปล่อยสวิตซ์ SW2 ส่งผลให้ LED1 ดับทันที

const int Sw1 = 2;	// กำหนดตัวแปร Sw1 ต่อที่พอร์ตดิจิตอลขา 2
const int Sw2 = 3;	// กำหนดตัวแปร Sw2 ต่อที่พอร์ตดิจิตอลขา 3
const int Ledi = 9;	// กำหนดตัวแปร Led1 ต่อที่พอร์ตดิจิตอลขา 9

void se	etup() {	
	pinMode(Sw1, INPUT);	// เซตพอร์ตดิจิตอลขา 2 ให้เป็นพอร์ตอินพุต
	pinMode(Sw2, INPUT);	// เซตพอร์ตดิจิตอลขา 3 ให้เป็นพอร์ตอินพุต
	pinMode(Led1, OUTPUT);	// เซตพอร์ตดิจิตอลขา 9 ให้เป็นพอร์ตเอาต์พุต
}		
void lo	op() {	
	if(digitalRead(Sw1) == LOW) {	// ถ้าอ่านค่าอินพุตตัวแปร Sw1 เท่ากับ 0
	digitalWrite(Led1,HIGH);	// LED1 ติดสว่าง
	}	
	else if(digitalRead(Sw2) == LOW) {	// ถ้าอ่านค่าอินพุตตัวแปร Sw2
	digitalWrite(Ledi,LOW);	// เท่ากับ 0 LED1 ดับ
	}	
	delay(10);	// หน่วงเวลา 0.01 วินาที

}

ผลการทดลอง

สังเกตการทำงาน	อธิบายลักษณะการแสดงผลของ LED1
เมื่อกด SW1	
เมื่อปล่อย SW2	

โปรแกรมที่ 4 โปรแกรมรับค่าสวิตซ์ 4 ตัว

ประกอบวงจรตามรูปที่ 5.8 ลงบนบอร์ดทดลองไมโครคอนโทรลเลอร์ Arduino โดยต่อพอร์ติ ดิจิตอลเป็น พอร์ตอินพุตที่ขา 2-5 กับสวิตซ์ SW1 - 4 ตามลำดับ และต่อพอร์ตดิจิตอลเป็นพอร์ต เอาต์พุตที่ขา 9 - 12 กับ หลอด LED1 - 4 ตามลำดับ



รูปที่ 5.8 วงจรการทดลองโปรแกรมรับค่าสวิตซ์ 4 ตัว

โปรแกรมนี้เริ่มต้นด้วยการกำหนดพอร์ตดิจิตอลขา 2 - 5 ให้เป็นพอร์ตอินพุต และกำหนด พอร์ตดิจิตอล ขา 9 - 12 ให้เป็นพอร์ตเอาต์พุต โปรแกรมทำการตรวจสอบการกดสวิตซ์ SW1 - SW4 โดยสวิตซ์แต่ละตัวทำการ ควบคุมหลอด LED1 - LED4 เรียงตามลำดับ เมื่อมีการกดสวิตช์ส่งผลให้ LED ติดสว่าง และเมื่อปล่อยสวิตซ์ส่งผล ให้ LED ดับ

int Sw1 = 2;	// กำหนดตัวแปร Sw1 ต่อที่พอร์ตดิจิตอลขา 2
int Sw2 = 3;	// กำหนดตัวแปร Sw2 ต่อที่พอร์ตดิจิตอลขา 3
int SW3 = 4;	// กำหนดตัวแปร Sw3 ต่อที่พอร์ตดิจิตอลขา 4
int Sw4 = 5;	// กำหนดตัวแปร Sw4 ต่อที่พอร์ตดิจิตอลขา 5
int Led1 = 9:	// กำหนดตัวแปร Led1 ต่อที่พอร์ตที่จิตอลขา 9
int Led2 = 10;	// กำหนดตัวแปร Led2 ต่อที่พอร์ตดิจิตอลขา 10
int Led3 = 11;	// กำหนดตัวแปร Led3 ต่อที่พอร์ตดิจิตอลขา 11
int Led4 = 12;	// กำหนดตัวแปร Led4 ต่อที่พอร์ตติจิตอลขา 12
void setup() {	

pinMode(Sw1, INPUT);	// เซตพอร์ตติจิตอลขา 2 ให้เป็นพอร์ตอินพุต
pinMode(Sw2, INPUT);	//เซตพอร์ตดิจิตอลขา 3 ให้เป็นพอร์ตอินพุต
pinMode(Sw3, INPUT);	//เซตพอร์ตดิจิตอลขา 4 ให้เป็นพอร์ตอินพุต
pinMode(Sw4, INPUT);	// เซตพอร์ตดิจิตอลขา 5 ให้เป็นพอร์ตอินพุต

วิชา พื้นฐานไมโครคอนโทรลเลอร์

	pinMode(Led1, OUTPUT);	// เซตพอร์ตดิจิตอลขา 9 ให้เป็นพอร์ตเอาต์พุต
	pinMode(Led2, OUTPUT);	//เซตพอร์ตดิจิตอลขา 10 ให้เป็นพอร์ตเอาต์พุต
	pinMode(Led3, OUTPUT);	//เซตพอร์ตดิจิตอลขา 11 ให้เป็นพอร์ตเอาต์พุต
	pinMode(Led4, OUTPUT);	// เซตพอร์ตดิจิตอลขา 12 ให้เป็นพอร์ตเอาต์พุต
}		
void lo	op) {	
	if(digitalRead(Sw1) == LOW)	// ถ้าอ่านค่าอินพุตตัวแปร Sw1 เท่ากับ 0
	digitalWrite(Led1, HIGH);	// LED1 ติดสว่าง
	else	
	digitalWrite(Led1, LOW);	// LED1 ดับ
	if(digitalRead(Sw2) == LOW)	// ถ้าอ่านค่าอินพุตตัวแปร Sw2 เท่ากับ 0
	digitalWrite(Led2, HIGH);	// LED2 ติดสว่าง
	else	
	digitalWrite(Led2, LOW);	// LED2 ดับ
	if(digitalRead(Sw3) == LOW)	// ถ้าอ่านค่าอินพุตตัวแปร SW3 เท่ากับ 0
	digitalWrite(Led3, HIGH);	// LED3 ติดสว่าง
	else	
	digitalWrite(Led3, LOW);	// LED3 ดับ
	if(digitalRead(Sw4) == LOW)	// ถ้าอ่านค่าอินพุตตัวแปร Sw4 เท่ากับ 0
	digitalWrite(Led4, HIGH);	// LED4 ติดสว่าง
	else	
	digitalWrite(Led4, LOW);	// LED4 ดับ
}		
ผลการทดลอง

สังเกตการทำงาน	อธิบายลักษณะการแสดงผลของ LED
เมื่อกด SW1	
เมื่อกด SW2	
เมื่อกด SW3	
เมื่อกด SW4	
เมื่อกด SW1,2 หรือ	
กด SW3,4 พร้อมกัน	
เมื่อกด SW1,2,3 หรือ	
กด SW2,3,4 พร้อมกัน	
เมื่อกด SW1,2,3,4 พร้อมกัน	

สรุปผลการทดลอง

29

กิจกรรมตรวจสอบความเข้าใจใบงานที่ 5 1

1. จงเขียนโปรแกรมตามโจทย์ที่กำหนดให้คือ

จากวงจรรูปที่ 5.9 กำหนดให้เขียนโปรแกรมรับข้อมูลจากสวิตช์ SW1 - SW4 เพื่อควบคุม การติด-ดับของ หลอด LED1 - LED4 ดังนี้

- กด SW1 เป็นสวิตซ์กดติด-กดดับของหลอด LED1
- กด SW2 เป็นสวิตช์กดติด-กดดับของหลอด LED2
- กด SW3 เป็นสวิตซ์กดติด-กดดับของหลอด LED3
- กด SW4 เป็นสวิตซ์กดติด-กดดับของหลอด LED4



รูปที่ 5.9 วงจรการทดลองโปรแกรมรับข้อมูลจากสวิตซ์ SW1 - Sw4

โปรแกรม

ใบงานที่ 6

โปรแกรม Arduino กับพอร์ตดิจิตอลอินพุต

จุดประสงค์การเรียนรู้

- 1. ศึกษาการทำงานของโปรแกรม Arduino ควบคุมแอลอีดีแสดงผล 7 ส่วน
- 2. สามารถเขียนโปรแกรม Arduino ควบคุมแอลอีดีแสดงผล 7 ส่วนได้
- 3. อธิบายโปรแกรม Arduino ควบคุมแอลอีดีแสดงผล 7 ส่วนได้

เครื่องมือและอุปกรณ์การทดลอง

- 1. เครื่องไมโครคอมพิวเตอร์
- 2. บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3
- 3. โปรแกรมการทดลอง
- 4. อุปกรณ์อิเล็กทรอนิกส์สำหรับทดลอง

การทดลอง

โปรแกรมที่ 1 วงจรนับ 0 ถึง 9

ประกอบวงจรตามรูปที่ 6.6 ลงบนบอร์ดทดลองไมโครคอนโทรลเลอร์ Arduino โดยต่อพอร์ต ดิจิตอลเป็น พอร์ตเอาต์พุตที่ขา 2, 3, 4, 5, 6, 7, 8 และ 9 ร่วมกับแอลอีดีแสดงผล 7 ส่วนเข้าที่ขา a, b, c, d, e, f, g และ dot ตามลำดับ จำนวน 1 หลัก โปรแกรมนี้ทำการแสดงตัวเลขตั้งแต่ 0 ถึง 9 แล้ววนซ้ำ โดยเว้นระยะเวลาในการ เปลี่ยนเป็นตัวเลขถัดไปเป็นเวลา 1 วินาที



รูปที่ 6.6 วงจรการทดลองโปรแกรมแสดงผลแอลอีดีแสดงผล 7 ส่วน

```
รหัสวิชา 20127-2018
```

32

```
// กำหนดตัวแปร Seven seg ต่อที่
const byte Seven seg[7] = {2, 3, 4, 5, 6, 7, 8};
                                                     // พอร์ตดิจิตอลขา 2-8 byte
                                                     // กำหนดตัวแปร Count ใช้สำหรับนับเลข
Count = 0;
                                                            // กำหนดตัวแปร numbers
const byte numbers] = { 0x3F, 0x06, 0x5B, 0x45, 0x66,
                                                            // ใช้แสดงตัวเลขต่าง ๆ
                      Ox6D, Ox7D, 0x07, 0xTE, Ox6F3;
void setup() {
                                 // ตัวแปร i=0; ถ้า |< 7; เพิ่ม i ครั้งละ 1/รอบ
       for (int i = 0; i < 7; i++) {
       pinMode( Seven seg [ ], OUTPUT); // เซตพอร์ตดิจิตอลขา 2-8 ให้เป็นพอร์ตเอาต์พุต
       }
}
void loop() {
       displayDigit( Count ); // ไปยังฟังก์ชัน displayDigit เพื่อแสดงตัวเลขแอลอีดี 7 ส่วน
       Count = (Count + 1) % 10; // ตัวแปร Count + 1 หาร 10 นำเศษที่ได้เป็นผลลัพธ์
                                     // หน่วงเวลา 0.5 วินาที
       delay(500);
}
                                             // กำหนดตัวแปร value
void displayDigit (byte value ) {
                                           // ตัวแปร value มีค่าระหว่าง 0 - 9
       if ( 0 <= value && Value < 10 ) {
               value = numbers [Count ]; // แปลงตัวเลขให้เป็น digit สถานะของ 7 Segment
                                             // ตัวแปร =0; ถ้า |< 8; เพิ่ม 1 ครั้งละ 1/รอบ
               for (int i = 0; i < 8; i++) {
                      digitalWrite(Seven seg[i], (value & 1));
                                             // นำค่าใน value ไปแสดงผลที่แอลอีดี 7 ส่วน
                                             // เลื่อนไปบิตของตัวแปร value บิตถัดไป
                      Value >>= 1;
               }
       }
```

}

สังเกตการทำงานของ	อธิบายลักษณะการแสดงผลตัวเลขของแอลอีดีแสดงผล 7 ส่วน
แอลอีดีแสดงผล 7 ส่วน	

โปรแกรมที่ 2 วงจรนับ 0 ถึง 9 โดยการกดสวิตช์

ประกอบวงจรตามรูปที่ 6.7 ลงบนบอร์ดทดลองไมโครคอนโทรลเลอร์ Arduino โดยต่อพอร์ต ดิจิตอลเป็น พอร์ตเอาต์พุตที่ขา 2, 3, 4, 5, 6, 7, 8 และ 9 ร่วมกับแอลอีดีแสดงผล 7 ส่วนเข้าที่ขา a, b, c, d, e, f, g และ dot ตามลำดับ จำนวน 1 หลัก ให้ทำการต่อสวิตซ์ SW1 เข้าที่พอร์ตอินพุตที่ ขา 10 เมื่อโปรแกรมเริ่มทำงาน แอลอีดีแสดงผล 7 ส่วนแสดงเลข 0 แต่ถ้ากดสวิตซ์ SW1 ตัวเลข จะนับขึ้นไปเรื่อย ๆ จนถึง 9 แล้ววนกลับมาที่เลข 0 เหมือนเดิม



รูปที่ 6.7 วงจรการทดลองโปรแกรมวงจรนับ 0 ถึง 9 โดยการกดสวิตช์

```
// กำหนดตัวแปร Seven seg ต่อที่
const byte Seven seg[7] = {2, 3, 4, 5, 6, 7, 8};
                                                    // พอร์ตดิจิตอลขา 2-8
                                                    // กำหนดตัวแปร Counter มีค่าเท่ากับ 0
int counter = 0;
                             // กำหนดตัวแปร SW1 ต่อที่พอร์ตดิจิตอลขา 10
const int Sw1 = 10;
                             // กำหนดตัวแปร lastState เพื่อเก็บสถานะเอาต์พุตที่ผ่านมา
boolean lastState:
                             // กำหนดตัวแปร reading เพื่อเก็บค่าจากอินพุต
boolean reading;
                             // กำหนดตัวแปร state มีค่าสถานะเป็น LOW
boolean state = LOW;
                                                           // กำหนดตัวแปร ทนmbers
byte numbers[10] = { Ox3F, 0x06, Ox5B, Ox4F, Ox66,
                      Ox6D, OX7D, 0x07, Ox7F, Ox6F };
void setup() {
       for(int i = 2; i <= 8; i++) { // ตัวแปร i=0; ถ้า i< 8: เพิ่ม 1 ครั้งละ 1/รอบ
       pinMode(Seven seg [ ], OUTPUT); // เซตพอร์ตดิจิตอลขา 2-8 ให้เป็นพอร์ตเอาต์พุต
       }
                                            // เซตพอร์ตดิจิตอลขา 10 ให้เป็นพอร์ตอินพุต
       pinMode(Sw1, INPUT);
}
void loop() {
       reading = digitalRead(Sw1); // อ่านค่าจากพอร์ตขา 10 มาเก็บไว้ในตัวแปร reading
       if (reading == LOW && tastState == HIGH ){ // ตรวจสอบการกดและปล่อยสวิตซ์
                                                    // หน่วงเวลา 0.01 วินาที
              delay (10);
                                                   // ตัวแปร counter บวก 1 หารเอาเศษ
              Counter = (Counter + 1) % 10;
              if (digitalRead(Sw1) == LOW) state = !state; // ทำการกลับค่าตัวแปร state
       }
                                    // ไปยังฟังก์ชัน displayDigit เพื่อแสดงตัวเลขแอลอีดี 7 ส่วน
       displayDigit(counter);
                                    // นำค่าในตัวแปร reading เก็บไว้ในตัวแปร lastState
       lastState = reading;
}
                                          // กำหนดตัวแปร value
void displayDigit (byte value ) {
       if ( 0 <= value && Value < 10 ) { // ตัวแปร value มีค่าระหว่าง 0 - 9
               value = numbers [ counter ]; // แปลงตัวเลขให้เป็น digit สถานะของ 7 Segment
              for (int i = 0; i < 8; i++) { // ตัวแปร i=0; ถ้า i< 8; เพิ่ม 1 ครั้งละ 1/รอบ
                      digitalWrite(Seven seg [], (value & 1));
```

// น้ำค่าใน value ไปแสดงผลที่แอลอีดี 7 ส่วน value >>= 1; // เลื่อนไปบิตของตัวแปร value บิตถัดไป }

35

}

ผลการทดลอง

}

สังเกตการทำงาน	อธิบายลักษณะการแสดงผลของแอลอีดีแสดงผล 7 ส่วน
การกดและปล่อย SW1	
ครั้งที่ 1	
ครั้งที่ 2	
ครั้งที่ 3	
ครั้งที่ 4	
ครั้งที่ 5	
ครั้งที่ 6	
ครั้งที่ 7	
ครั้งที่ 8	
ครั้งที่ 9	
ครั้งที่ 10	
ครั้งที่ 11	

้โปรแกรมที่ 3 วงจรนับขึ้น-ลง 0 ถึง 9 โดยการกดสวิตซ์

ประกอบวงจรตามรูปที่ 6.8 ลงบนบอร์ดทดลองไมโครคอนโทรลเลอร์ Arduino โดยต่อพอร์ต ดิจิตอลเป็น พอร์ตเอาต์พุตที่ขา 2, 3, 4, 5, 6, 7, 8 และ 9 ร่วมกับแอลอีดีแสดงผล 7 ส่วนเข้าที่ขา a, b, c, d, e, f, g และ dot ตามลำดับ จำนวน 1 หลัก ให้ทำการต่อสวิตช์ SW1 เข้าที่พอร์ตอินพุต ที่ขา 10 และต่อสวิตช์ SW2 เข้าที่ พอร์ตอินพุตที่ขา 11 เมื่อโปรแกรมเริ่มทำงานแอลอีดีแสดงผล 7 ส่วนแสดงเลข 0 แต่ถ้ากดสวิตซ์ SW1 ตัวเลขจะ นับขึ้นไปเรื่อย ๆ จนถึง 9 เท่านั้น แต่ถ้ากด SW2 ตัวเลขจะนับลงมาเรื่อย ๆ จนถึง 0 เท่านั้น



รูปที่ 6.8 วงจรการทดลองโปรแกรมวงจรนับขึ้นลง 0 ถึง 9 โดยการกดสวิตซ์

const int Sw1 = 10;		// กำหนดตัวแปร Sw1 ต่อที่พอร์ตดิจิตอลขา 10
const int Sw2 = 11;		// กำหนดตัวแปร Sw1 ต่อที่พอร์ตดิจิตอลขา 11
const byte Seven seg[7] = {2, 3, 4, 5,	6, 7, 8};	// กำหนดตัวแปร Seven seg ต่อที่
		// พอร์ตดิจิตอลขา 2-8
boolean lastState;	// กำหนดตัวแป	ร lastState เพื่อเก็บสถานะเอาต์พุตที่ผ่านมา
boolean lastState2;	// กำหนดตัวแป	ร lastState เพื่อเก็บสถานะเอาต์พุตที่ผ่านมา
boolean reading;	// กำหนดตัวแป	ร reading เพื่อเก็บค่าจากอินพุต
boolean reading2;	// กำหนดตัวแป	ร reading เพื่อเก็บค่าจากอินพุต
boolean state = LOW;	// กำหนดตัวแป	ร state มีค่าสถานะเป็น LOW
boolean state2 = LOW;	// กำหนดตัวแป	ร state มีค่าสถานะเป็น LOW
int Counter = 0;	// กำหนดตัวแป	ร Counter มีค่าเท่ากับ 0
byte numbers[10] = {0x3F, 0x06, 0x5	B, 0x4F, Ox66,	// กำหนดตัวแปร numbers
Ox6D, 0x7D, 0;	x07, 0x75, 0x6F	};
void setup() {		
for(int i = 2; i <= 8; i++) {		// ตัวแปร =0; ถ้า i<8; เพิ่ม 1 ครั้งละ 1/รอบ
pinMode(Seven seg []	, OUTPUT);	//เซตพอร์ตดิจิตอลขา 2-8 ให้เป็นพอร์ตเอาต์พุต
}		
pinMode(Sw1, INPUT);		// เซตพอร์ตดิจิตอลขา 10 ให้เป็นพอร์ตอินพุต

```
// เซตพอร์ตดิจิตอลขา 11 ให้เป็นพอร์ตอินพุต
       pinMode(Sw2, INPUT);
}
void loop() {
                                                  // อ่านค่าจากพอร์ตขา 2 มาเก็บไว้ในตัวแปรreding
       reading = digitalRead(Sw1);
       if (reading == LOW & LastState == HIGH ) { // ตรวจสอบการกดและปล่อยสวิตซ์
                                                  // หน่วงเวลา 0.01 วินาที
              delay (10);
              counter = (counter + 1); // ตัวแปร counter บวก 1
              if (counter>9) {
                                         // ถ้าตัวแปร Counter มากกว่า 9
                     counter = 9; // ให้ตัวแปร Counter เท่ากับ 9
              }
       if (digitalRead(Sw1) == LOW) state = !state; //ทำการกลับค่าตัวแปร state
       }
       reading2 = digitalRead(Sw2); // อ่านค่าจากพอร์ตขา 2 มาเก็บไว้ในตัวแปร reading
       if ( reading2 == LOW && lastState2 == HIGH ) { // ตรวจสอบการกดและปล่อยสวิตซ์
                                           // หน่วงเวลา 0.01 วินาที
              delay (10);
              counter = (counter - 1);
                                           // ตัวแปร counter ลบ 1
                                           // ถ้าตัวแปร Counter น้อยกว่า 0
              if (counter <0) {
                                         // ให้ตัวแปร counter เท่ากับ 0
                     counter = 0;
              }
       if (digitalRead(Sw2) == LOW) state2 = !state2; //ทำการกลับค่าตัวแปร state
       }
                                           // ไปที่โปรแกรมย่อย displayDigit
       display Digit(counter);
                                           // นำค่าในตัวแปร readingเก็บไว้ในตัวแปร lastState
       lastState = reading;
                                           // น้ำค่าในตัวแปร reading เก็บไว้ในตัวแปร lastState
       lastState2 = reading2;
                               // กำหนดตัวแปร
void displayDigit (byte value ) {
       value if ( 0 = value && value < 10) { // ตัวแปร value มีค่าระหว่าง 0 - 9
              value = numbers [ counter]; // แปลงตัวเลขให้เป็น digit สถานะของ 7 Segment
              for (int i = 0; i < 8; i++) { // ตัวแปร i=0; ถ้า i< 8; เพิ่ม 1 ครั้งละ 1/รอบ
                      digitalWrite(Seven seg [i], (value & 1));
```

37

}

// นำค่าใน value ไปแสดงผลที่แอลอีดี 7 ส่วน value >>= 1; // เลื่อนไปบิตของตัวแปร value บิตถัดไป } }

}

สังเกตการทำงาน	อธิบายลักษณะการแสดงผลของ LED1
กดสวิตช์ SW1	
กดสวิตช์ SW2	

สรุปผลการทดลอง

กิจกรรมตรวจสอบความเข้าใจใบงานที่ 6

1. จงเขียนโปรแกรมตามโจทย์ที่กำหนดให้คือ

จากวงจรรูปที่ 6.9 กำหนดให้เขียนโปรแกรมรับข้อมูลจากสวิตซ์ SW1 - SW4 เพื่อควบคุม การแสดงผลที่ แอลอีดีแสดงผล 7 ส่วน ดังนี้ เมื่อโปรแกรมทำงานแล้วให้แอลอีดีแสดงผล 7 ส่วน แสดง ตัวเลข 0 เมื่อทำการกด สวิตซ์ให้แสดงผลตัวเลขตามสวิตซ์ที่ถูกกด

- กด SW1 แสดงเลข 1
- กด SW2 แสดงเลข 2
- กด SW3 แสดงเลข 3
- กด SW4 แสดงเลข 4



รูปที่ 6.9 วงจรโปรแกรมแสดงผลแอลอีดีแสดงผล 7 ส่วน

โปรแกรม

ใบงานที่ 7

โปรแกรม Arduino ควบคุมโมดูลแสดงผล LCD

จุดประสงค์การเรียนรู้

- 1. ศึกษาคำสั่งการทำงานเพื่อควบคุมโมดูลแสดงผล LCD
- 2. สามารถเขียนโปรแกรม Arduino ควบคุมโมดูลแสดงผล LCD ได้
- 3. อธิบายโปรแกรม Arduino ควบคุมโมดูลแสดงผล LCD ได้

เครื่องมือและอุปกรณ์การทดลอง

- 1. เครื่องไมโครคอมพิวเตอร์
- 2. บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3
- 3. โปรแกรมการทดลอง
- 4. อุปกรณ์อิเล็กทรอนิกส์สำหรับทดลอง

การทดลอง

โปรแกรมที่ 1 แสดงข้อความ "HELLO!"

ในโปรแกรมนี้เป็นการเขียนโปรแกรมแสดงผลที่โมดูลแสดงผล LCD โดยแสดงข้อความ "HELLO!" ใน บรรทัดที่ 1 และ "MR.ARDUINO" ในบรรทัดที่ 2 เป็นเวลา 2 วินาที จากนั้นข้อความ ทั้งบรรทัดจะหายเป็นเวลา อีก 2 วินาที สลับกันไปเรื่อย ๆ



รูปที่ 7.5 วงจรโปรแกรมแสดงข้อความบนโมดูลแสดงผล LCD

#include <liquidcrystal.h></liquidcrystal.h>	// ประกาศใช้ไลบรารี LiquidCrystalth
LiquidCrystal Lcd(12, 11, 4, 5, 6, 7)	// เป็นการกำหนดขาที่ใช้ในการเชื่อมต่อ LCD
void setup() {	
lcd.begin(16, 2);	// กำหนดขนาดโมดูล LCD ขนาด 16x2
lcd.setCursor(0,0);	// กำหนดเคอร์เซอร์ตำแหน่งที่ 1 ในบรรทัดที่ 1
lcd.print("HELLO!");	// แสดงข้อความที่ต้องการแสดงผล
lcd.setCursor(0,1);	//กำหนดเคอร์เซอร์ตำแหน่งที่ 1 ในบรรทัดที่ 2
lcd.print("MR.ARDUINO ");	// แสดงข้อความที่ต้องการแสดงผล
}	
void loop() {	
lcd.display();	// เปิดการแสดงข้อความ
delay(2000);	// หน่วงเวลา 2 วินาที
lcd.noDisplay();	// ปิดการแสดงข้อความ
delay(2000);	// หน่วงเวลา 2 วินาที

ผลการทดลอง

สังเกตการทำงานของ	อธิบายลักษณะการแสดงผลตัวเลขของ LCD
LCD	

เมื่อทำการทดลองโปแกรมที่ 1 เสร็จแล้ว ให้ทำการแก้ไขโปรแกรมในบรรทัดต่อไปนี้

lcd.setCursor(0,0); ເປັ້ນ lcd.setCursor(4,0);

lcd.setCursor(0,1); ເປັ້ນ lcd.setCursor(3,1);

เมื่อแก้ไขเสร็จแล้วทำการทดลองใหม่อีกครั้ง และบันทึกผลการทดลองลงในตารางด้านล่าง

ผลการทดลอง

สังเกตการทำงานของ	อธิบายลักษณะการแสดงผลตัวเลขของ LCD
LCD	

โปรแกรมที่ 2 โปรแกรมนับขึ้น 0-100

โปรแกรมนี้เป็นการเขียนโปรแกรมแสดงผลที่โมดูลแสดงผล LCD โดยแสดงข้อความ "COUNTER" ใน บรรทัดที่ 1 และในบรรทัดที่ 2 แสดงตัวเลขโดยเริ่มนับตั้งแต่เลข 0 ไปจนถึง 100

#include <liquidcrystal.h></liquidcrystal.h>	// ประกาศใช้ไลบรารี LiquidCrystal.h
LiquidCrystal Lcd(12, 11, 4, 5, 6, 7);	// เป็นการกำหนดขาที่ใช้ในการเชื่อมต่อ LCD
void setup() {	
lcd.begin(16,2);	// กำหนดขนาดโมดูล LCD ขนาด 16x2
lcd.setCursor(5,0);	// กำหนดเคอร์เซอร์ตำแหน่งที่ 5 ในบรรทัดที่ 1
lcd.print("COUNTER");	// แสดงข้อความที่ต้องการแสดงผล
delay(100);	// หน่วงเวลา 0.1 วินาที
int a=0;	// กำหนดตัวแปร a = 0
lcd.setCursor(7,1);	// กำหนดเคอร์เซอร์ตำแหน่งที่ 7 ในบรรทัดที่ 2
lcd.print(a);	// แสดงค่าตัวเลขในตัวแปร a
while(a<=100) {	// ตรวจสอบเงื่อนไขตัวแปร 2 ถ้า 2 มีค่าน้อยกว่า
	// หรือเท่ากับ 100 ให้ทำตามคำสั่งในลูป while
a=a+1;	// ตัวแปร a+1 ผลลัพธ์เก็บไว้ที่ 2
delay(500);	// หน่วงเวลา 0.5 วินาที
lcd.setCursor(7,1);	// กำหนดเคอร์เซอร์ตำแหน่งที่ 7 ในบรรทัดที่ 2
lcd.print(a);	// แสดงค่าตัวเลขในตัวแปร a
}	

}

void loop() {

lcd.clear();

// ลบข้อความบนหน้าจอ LCD ทั้งหมด

}

ผลการทดลอง

สังเกตการทำงานของ	อธิบายลักษณะการแสดงผลตัวเลขของ LCD
LCD	

โปรแกรมที่ 3 โปรแกรมนับขึ้น-ลงโดยการกดสวิตซ์

สำหรับโปรแกรมนี้เป็นการเขียนโปรแกรมแสดงผลที่โมดูลแสดงผล LCD โดยแสดงข้อความ "Up-Down Counter" ในบรรทัดที่ 1 และในบรรทัดที่ 2 แสดงตัวเลขโดยเริ่มนับตั้งแต่เลข 0 ขึ้นไปเรื่อย ๆ แต่ตัวเลขสามารถ นับขึ้นได้ต้องทำการกดสวิตซ์ Sw1 ถ้าต้องการให้ตัวเลขนับลง ต้องทำการกดสวิตซ์ Sw2



รูปที่ 7.6 วงจรโปรแกรมนับขึ้น-ลงโดยการกดสวิตซ์

43

#include <Liquid Crystal.h> LiquidCrystal lcd(12, 11, 4, 5, 6, 7); int sw1 = 1; int sw2 = 2; int count = 0; void setup() { lcd.begin(16, 2); lcd.setCursor(0,0); lcd.print("Up-Down Counter"); lcd.setCursor(2,1); lcd.print(count); pinMode(sw1,INPUT); pinMode(SW2, INPUT); void loop() { if(digitalRead(sw1) == LOW) { count++; lcd.setCursor(0,0); lcd.print("Up-Down Counter"); lcd.setCursor(2,1); lcd.print(count); delay(400); } if(digitalRead(sw2) == LOW) {

count--; if(count <0) count = 0; lcd.clear(); lcd.setCursor(0,0); lcd.print("Up-Down Counter"); lcd.setCursor(2,1); // ประกาศใช้ไลบรารี LiquidCrystal.h
// เป็นการกำหนดขาที่ใช้ในการเชื่อมต่อ LCD
// กำหนดตัวแปร SW1 ต่อที่พอร์ตดิจิตอลขา 1
// กำหนดตัวแปร Sw2 ต่อที่พอร์ตดิจิตอลขา 2
// กำหนดตัวแปร Count = 0

// กำหนดขนาดโมดูล LCD ขนาด 16x2
// กำหนดเคอร์เซอร์ตำแหน่งที่ 1 ในบรรทัดที่ 1
// แสดงข้อความที่ต้องการแสดงผล
// กำหนดเคอร์เซอร์ตำแหน่งที่ 2 ในบรรทัดที่ 2
// แสดงค่าตัวเลขในตัวแปร Count
// เซตพอร์ตดิจิตอลขา 1 ให้เป็นพอร์ตอินพุต
// ถ้าอ่านค่าอินพุตตัวแปร SW1 เท่ากับ 0
// ตัวแปร Count + 1
// กำหนดเคอร์เซอร์ตำแหน่งที่ 1 ในบรรทัดที่ 1
// แสดงข้อความที่ต้องการแสดงผล
// กำหนดเคอร์เซอร์ตำแหน่งที่ 2 ในบรรทัดที่ 2
// แสดงข้อความที่ต้องการแสดงผล
// กำหนดเคอร์เซอร์ตำแหน่งที่ 2 ในบรรทัดที่ 2
// แสดงข้อความที่ต้องการแสดงผล
// กำหนดเคอร์เซอร์ตำแหน่งที่ 2 ในบรรทัดที่ 2
// แสดงค่าตัวเลขในตัวแปร Count
// หน่วงเวลา 0.4 วินาที

//ถ้าอ่านค่าอินพุตตัวแปร sw2 เท่ากับ 0 // ตัวแปร count - 1 // ถ้าตัวแปร Count น้อยกว่า 0 // ให้ตัวแปร Count = 0 // ลบข้อความบนหน้าจอ LCD ทั้งหมด // กำหนดเคอร์เซอร์ตำแหน่งที่ 1 ในบรรทัดที่ 1 // แสดงข้อความที่ต้องการแสดงผล // กำหนดเคอร์เซอร์ตำแหน่งที่ 2 ในบรรทัดที่ 2 lcd.print(count); delay(400); // แสดงค่าตัวเลขในตัวแปร count// หน่วงเวลา 0.4 วินาที

ผลการทดลอง

}

}

สังเกตการทำงานของ	อธิบายลักษณะการแสดงผลตัวเลขของ LCD
LCD	

สรุปผลการทดลอง

กิจกรรมตรวจสอบความเข้าใจใบงานที่ 7

1. จงเขียนโปรแกรมตามโจทย์ที่กำหนดให้คือ

จากวงจรรูปที่ 7.7 กำหนดให้เขียนโปรแกรมรับข้อมูลจากสวิตซ์ SW1 - SW4 เพื่อ ควบคุมการแสดงผลที่ โมดูลแสดงผล LCD ดังนี้ เมื่อโปรแกรมทำงานแล้วให้จอ LCD บรรทัดแรก แสดงประโยคว่า "Program =" และ บรรทัดที่ 2 แสดงตัวเลขตามสวิตซ์ที่ถูกกด

- กด SW1 แสดงเลข 1

- กด SW2 แสดงเลข 2

วิทยาลัยเทคนิคชลบุรี

จัดทำโดย นายวิรุณ จิตต์บุญ



- กด SW4 แสดงเลข 4



รูปที่ 7.7 วงจรโปรแกรมแสดงข้อความบนโมดูลแสดงผล LCD สวิตซ์กด 4 ตัว

โปรแกรม

ใบงานที่ 8

โปรแกรม Arduino กับอนาล็อกอินพุต

จุดประสงค์การเรียนรู้

- 1. ศึกษาการทำงานของโปรแกรม Arduino กับอนาล็อกอินพุต
- 2. สามารถเขียนโปรแกรม Arduino กับอนาล็อกอินพุตได้
- 3. อธิบายโปรแกรม Arduino กับอนาล็อกอินพุตได้

เครื่องมือและอุปกรณ์การทดลอง

- 1. เครื่องไมโครคอมพิวเตอร์
- 2. บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3
- 3. โปรแกรมการทดลอง
- 4. อุปกรณ์อิเล็กทรอนิกส์สำหรับทดลอง

การทดลอง

โปรแกรมที่ 1 โปรแกรมการอ่านค่าอนาล็อกอินพุต

สำหรับโปรแกรมนี้เป็นการแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล โดยการอ่านค่าแรงดัน ที่ได้จาก การปรับค่าที่โพเทนทิโอมิเตอร์ ซึ่งถูกเชื่อมต่อกับขาอนาล็อกของไมโครคอนโทรลเลอร์ Arduino โปรแกรมนี้เป็น การทดสอบที่พอร์ต A0 ต่อกับโพเทนที่โอมิเตอร์เพื่อรับค่าสัญญาณอนาล็อก อินพุต และนำค่าที่อ่านได้ไปแสดงผล ที่หน้าจอคอมพิวเตอร์ เมื่ออัปโหลดโปรแกรมที่เขียนนี้ลงบอร์ด Arduino จากนั้นคลิกปุ่ม Serial Monitor เพื่อดู ผลการทดลอง



รูปที่ 8.5 วงจรโปรแกรมการอานค่าอนาล็อกอินพุต

int potPin = 0;	// กำหนดตัวแปร potPin ต่อที่พอร์ตอนาล็อกขา O
int val = 0;	// กำหนดตัวแปร val = 0
void setup() {	
Serial.begin(9600);	// เซตค่าการติดต่อสื่อสารแบบอนุกรม
}	
void loop() {	
val = analogRead(potPin);	// อ่านค่าอนาล็อกจากโพเทนทิโอมิเตอร์
Serial.println(val);	// น้ำค่าตัวแปร vat แสดงบนจอคอมพิวเตอร์
delay(500);	// หน่วงเวลา 0.5 วินาที

48

}

ผลการทดลอง

รหัสวิชา 20127-2018

ทดลองปรับค่า VR	อธิบายลักษณะการแสดงผลที่จอคอมพิวเตอร์
ทดลองโดยการปรับค่า VR1 ให้	
แรงดันน้อยไปหาแรงดันมาก(ใช้มัลติ	
มิเตอร์วัดแรงดันที่พอร์ต A0 ปรับ	
VR1 ให้ได้แรงดัน 0 โวลต์)	

โปรแกรมที่ 2 ไฟกะพริบตามค่าอนาล็อกอินพุต

โปรแกรมนี้เป็นการทดสอบที่พอร์ต A0 ต่อกับโพเทนทิโอมิเตอร์เพื่อรับค่าสัญญาณอนาล็อก อินพุต และ นำค่าที่อ่านได้ไปเป็นค่าของการหน่วงเวลาในโปรแกรม เพื่อให้หลอดแอลอีดีที่ต่อกับพอร์ต ดิจิตอลขา 13 ติด กะพริบช้าหรือเร็วตามค่าอนาล็อกที่อ่านได้จากโพเทนที่โอมิเตอร์



รูปที่ 8.6 วงจรโปรแกรมไฟกะพริบตามควอนาล็อกอินพต

int potPin = 0; int ledPin = 13; int val = 0; // กำหนดตัวแปร potPin ต่อที่พอร์ตอนาล็อกขา 0
 // กำหนดตัวแปร ledPin ต่อที่พอร์ตดิจิตอลขา 13
 // กำหนดตัวแปร val = 0

```
void loop() {
```

```
val = analogRead(potPin);
digitalWrite(ledPin, HIGH);
delay(val);
digitalWrite(ledPin, LOW);
delay(val);
```

// อ่านค่าอนาล็อกจากโพเทนทิโอมิเตอร์
 // ส่งข้อมูลออกเอาต์พุตเป็นลอจิก "1"
 // หน่วงเวลาเท่ากับค่าที่อ่านได้จากโพเทนทิโอมิเตอร์
 // ส่งข้อมูลออกเอาต์พุตเป็นลอจิก "0"

// หน่วงเวลาเท่ากับค่าที่อ่านได้จากโพเทนที่โอมิเตอร์

}

ผลการทดลอง

ทดลองปรับค่า VR	อธิบายลักษณะการแสดงผลที่จอคอมพิวเตอร์
ทดลองโดยการปรับค่า VR1 ให้	
แรงดันน้อยไปหาแรงดันมาก(ใช้มัลติ	
มิเตอร์วัดแรงดันที่พอร์ต A0 ปรับ	
VR1 ให้ได้แรงดัน 0 โวลต์)	

โปรแกรมที่ 3 โปรแกรมการอ่านค่าอนาล็อกแสดงผลที่หลอดแอลอีดี

โปรแกรมนี้เป็นการทดสอบที่พอร์ต A0 ต่อกับโพเทนที่โอมิเตอร์เพื่อรับค่าสัญญาณอนาล็อก อินพุต และ นำค่าที่อ่านได้ไปแสดงผลออกทางหลอดแอลอีดีจำนวน 10 ดวง โดยต่อกับพอร์ตดิจิตอล ขา 2 ถึง 11 เมื่อ โปรแกรมอ่านค่าอนาล็อกจากพอร์ต A0 แล้วทำการแปลงค่าเพื่อนำไปแสดงผล ที่หลอดแอลอีดีทั้ง 10 ดวง โดยนำ ค่าอนาล็อกขนาด 10 บิตซึ่งมีค่าตั้งแต่ 0-1023 มาแบ่งเป็นช่วง ความห่างเท่า ๆ กันจำนวน 10 ช่วง คือ เมื่อค่าที่ แปลงดังกล่าวถึงค่าสูงสุดในช่วงนั้นจะทำให้หลอด แอลอีดีในช่วงนั้นติดทันที เช่น ถ้าปรับโพเทนทิโอมิเตอร์ไว้ ตำแหน่งต่ำสุดให้มีค่าอนาล็อกเท่ากับ 0 แล้วค่อย ๆ ปรับเพิ่มค่าอนาล็อก ถ้าค่าอนาล็อกมีค่า 102 จะทำให้หลอด แอลอีดีที่ต่อที่พอร์ตดิจิตอล ขา 2 ติดทันที



รูปที่ 5.7 วงจรการทดลองโปรแกรมการอ่านค่าอนาล็อกแสดงผลที่หลอดแอลอีดี

const int analogPin = A0;	// กำหนดตัวแปร analogPin ต่อที่พอร์ตอนาล็อกขา 0
const int ledCount = 10;	// กำหนดตัวแปร ledCount = 10
int ledPins] = {2, 3, 4, 5, 6, 7, 8, 9, 10, 11};	// กำหนดตัวแปร ledPins ต่อที่
	// พอร์ตดิจิตอลขา 2-11

```
void setup() {
```

```
for (int i = 0; i < (edCount; i++) { // ตัวแปร I=0, ถ้า |< ledCount; เพิ่ม 1 ครั้งละ 1/รอบ
pinMode(ledPins[], OUTPUT); // เซตพอร์ตดิจิตอลขา 2-11 ให้เป็นพอร์ตเอาต์พุต
}
```

} void loop() { //อ่านค่าอนาล็อกจากโพเทนที่โอมิเตอร์ int SensorReading = analogRead(analogPin); int ledLevel = map (sensorReading, 0, 1023, 0, ledCount); // กำหนดระดับช่วงความห่าง for (int i = 0; i < ledCount; 1++) { // ตัวแปร |=0; ถ้า |< ledCount: เพิ่ม 1 ครั้งละ 1/รอบ // ถ้า i< ledCount if (i <ledLevel) { // ส่งข้อมูลออกเอาต์พุตเป็นลอจิก "1" digitalWrite(ledPins[i], HIGH); } // นอกเหนือจากเงื่อนไข else { // ส่งข้อมูลออกเอาต์พุตเป็นลอจิก "0" digitalWrite(ledPins[], LOW); } } }

ผลการทดลอง

ทดลองปรับค่า VR	อธิบายลักษณะการแสดงผลของ LED
ทดลองโดยการปรับค่า VR1 ให้	
แรงดันน้อยไปหาแรงดันมาก(ใช้มัลติ	
มิเตอร์วัดแรงดันที่พอร์ต A0 ปรับ	
VR1 ให้ได้แรงดัน 0 โวลต์)	

โปรแกรมที่ 4 โปรแกรมแปลงอนาล็อกเป็นดิจิตอลแสดงผลจอ LCD

โปรแกรมนี้เป็นการนำค่าสัญญาณอนาล็อกอินพุตที่พอร์ต A0 ต่อกับโพเทนทิโอมิเตอร์ และนำค่าที่อ่านได้ ไปแสดงผลออกทางจอ LCD ตามวงจรรูปที่ 8.8 เมื่อโปรแกรมอ่านค่าอนาล็อก จากพอร์ต AO แล้วทำการแปลงค่า เพื่อนำไปแสดงผลทางจอ LCD โดยมีค่าตั้งแต่ 0-1023

51



รูปที่ 8.8 วงจรกวรทดลองโปรแกรมแปลงอนาล็อกเป็นดิจิตอลแสดงผลจอ LCD

#include <liquidcrystal.h></liquidcrystal.h>	// ประกาศใช้ไลบรารี
LiquidCrystal Lcd(12, 11, 4, 5, 6, 7);	// เป็นการกำหนดขาที่ใช้ในการเชื่อมต่อ LCD
int potPin = 0;	// กำหนดตัวแปร potPin ต่อที่พอร์ตอนาล็อกขา 0
int val = 0;	// กำหนดตัวแปร val = 0

```
lcd.begin(16,2); // กำหนดขนาดโมดูล LCD ขนาด 16x2
lcd.setCursor(6,0); // กำหนดเคอร์เซอร์ตำแหน่งที่ 6 ในบรรทัดที่ 1
lcd.print("Pot ="); // แสดงข้อความที่ต้องการแสดงผล
```

}

void loop() {

void setup() {

lcd.setCursor(6,0);	// กำหนดเคอร์เซอร์ตำแหน่งที่ 5 ในบรรทัดที่ 1
lcd.print("Pot =");	// แสดงข้อความที่ต้องการแสดงผล
val = analogRead(potPin);	// อ่านค่าอนาล็อกจากโพเทนทิโอมิเตอร์
lcd.setCursor(7,1);	// กำหนดเคอร์เซอร์ตำแหน่งที่ 7 ในบรรทัดที่ 2
lcd.print(val);	// แสดงค่าตัวเลขในตัวแปร val
delay(100);	// หน่วงเวลา 0.1 วินาที

lcd.clear();

// ลบข้อความบนหน้าจอ LCD ทั้งหมด

}

ผลการทดลอง

ทดลองปรับค่า VR	อธิบายลักษณะการแสดงผลของ LCD
ทดลองโดยการปรับค่า VR1 ให้	
แรงดันน้อยไปหาแรงดันมาก(ใช้มัลติ	
มิเตอร์วัดแรงดันที่พอร์ต A0 ปรับ	
VR1 ให้ได้แรงดัน 0 โวลต์)	

สรุปผลการทดลอง

กิจกรรมตรวจสอบความเข้าใจใบงานที่ 8 1. จงเขียนผังงาน และโปรแกรมตามโจทย์ที่กำหนดให้คือ จากวงจรรูปที่ 8.8 ให้เขียนโปรแกรมแสดงค่าสัญญาณอนาล็อกออกทางจอ LCD โดย มีเงื่อนไขดังนี้ จอ LCD บรรทัดที่ 1 แสดงข้อความ PROGRAM จอ LCD บรรทัดที่ 2 แสดงข้อความตามเงื่อนไขดังนี้ - ถ้าค่าสัญญาณอนาล็อกมีค่าตั้งแต่ 0 - 255 ให้ LCD แสดงเลข 1 - ถ้าค่าสัญญาณอนาล็อกมีค่าตั้งแต่ 256 – 511 ให้ LCD แสดงเลข 2 - ถ้าค่าสัญญาณอนาล็อกมีค่าตั้งแต่ 512 - 767 ให้ LCD แสดงเลข 3 - ถ้าค่าสัญญาณอนาล็อกมีค่าตั้งแต่ 768 - 1023 ให้ LCD แสดงเลข 4 โปรแกรม

ใบงานที่ 9

โปรแกรม Arduino ควบคุมสวิตช์เมทริกซ์

จุดประสงค์การเรียนรู้

- 1. ศึกษาการทำงานของโปรแกรม Arduino ควบคุมกับสวิตซ์เมทริกซ์
- 2. สามารถเขียนโปรแกรม Arduino ควบคุมสวิตช์เมทริกซ์ได้
- 3. อธิบายโปรแกรม Arduino ควบคุมสวิตซ์เมทริกซ์ได้

เครื่องมือและอุปกรณ์การทดลอง

- 1. เครื่องไมโครคอมพิวเตอร์
- 2. บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3
- 3. โปรแกรมการทดลอง
- 4. อุปกรณ์อิเล็กทรอนิกส์สำหรับทดลอง

การทดลอง

โปรแกรมที่ 1 โปรแกรมการรับค่าสวิตซ์เมทริกซ์

โปรแกรมนี้ต่อวงจรสวิตซ์เมทริกซ์ โดยการต่อวงจรสวิตซ์เมทริกซ์ให้นำสายสัญญาณแนวนอน (Row) ตั้งแต่แถว Row1- Row3 ต่อเข้าที่พอร์ตดิจิตอลขา 5, 4, 3, 2 และต่อสายสัญญาณแนวตั้ง (Column) ตั้งแต่แถว Col1- Col3 ต่อเข้าที่พอร์ตดิจิตอลขา 6, 7, 8 ในการทดสอบโปรแกรมให้เปิด หน้าต่างจอภาพคอมพิวเตอร์แบบ อนุกรมเพื่อดูผลการทำงานของโปรแกรม



รูปที่ 9.3 วงจรการทดลองโปรแกรมการรับควบคุมสวิตซ์เมทริกซ์

โปรแกรมการทดลอง

int r1 = 5;	// กำหนดตัวแปร r1 = 5
int r2 = 4;	// กำหนดตัวแปร r2 = 4
int r3 = 3;	// กำหนดตัวแปร r3 = 3
int r4 = 2;	// กำหนดตัวแปร r4 = 2
int c1 = 6;	// กำหนดตัวแปร c1 = 6
int c2 = 7,	// กำหนดตัวแปร C2 = 7
int c3 = 8;	// กำหนดตัวแปร c3 = 8
int colm1;	// กำหนดตัวแปร colm1
int colm2;	// กำหนดตัวแปร colm2
int colm3;	// กำหนดตัวแปร colm3

void setup() {

pinMode(r1,OUTPUT);
pinMode(r2,OUTPUT);
pinMode(r3,OUTPUT);
pinMode(r4,OUTPUT);
pinMode(c1,INPUT);
pinMode(c2,NPUT);
pinMode(c3,INPUT);
Serial.begin(9600);
digitalWrite(c1,HIGH);
digitalWrite(c2,HIGH);
digitalWrite(c3,HIGH);

}

void loop() {

digitalWrite(1,LOW);
digitalWrite(r2,HIGH);

- digitalWrite(r3,HIGH);
- digitalWrite(rd,HIGH);

// เซตพอร์ตดิจิตอลขา 5 ให้เป็นพอร์ตเอาต์พุต
// เซตพอร์ตดิจิตอลขา 4 ให้เป็นพอร์ตเอาต์พุต
// เซตพอร์ตดิจิตอลขา 3 ให้เป็นพอร์ตเอาต์พุต
// เซตพอร์ตดิจิตอลขา 2 ให้เป็นพอร์ตเอาต์พุต
// เซตพอร์ตดิจิตอลขา 6 ให้เป็นพอร์ตอินพุต
// เซตพอร์ตดิจิตอลขา 7 ให้เป็นพอร์ตอินพุต
// เซตพอร์ตดิจิตอลขา 8 ให้เป็นพอร์ตอินพุต
// เซตค่าการติดต่อสื่อสารแบบอนุกรม
// ส่งข้อมูลออกพอร์ตดิจิตอลขา 6 เอาต์พุตเป็นลอจิก "1"
// ส่งข้อมูลออกพอร์ตติจิตอลขา 7 เอาต์พุตเป็นลอจิก "1"
// ส่งข้อมูลออกพอร์ตดิจิตอลขา 8 เอาต์พุตเป็นลอจิก "1"

- // ส่งข้อมูลออกพอร์ตดิจิตอลขา 5 เอาต์พุตเป็นลอจิก "0"
- // ส่งข้อมูลออกพอร์ตดิจิตอลขา 4 เอาต์พุตเป็นลอจิก "1"
- // ส่งข้อมูลออกพอร์ตดิจิตอลขา 3 เอาต์พุตเป็นลอจิก "1"
- // ส่งข้อมูลออกพอร์ตดิจิตอลขา 2 เอาต์พุตเป็นลอจิก "1"

วิทยาลัยเทคนิคชลบุรี

```
// อ่านค่าพอร์ตดิจิตอลขา 6 เก็บค่าที่ตัวแปร colm1
colm1=digitalRead(c1);
                               // อ่านค่าพอร์ตดิจิตอลขา 7 เก็บค่าที่ตัวแปร cotm2
colm2=digitalRead(C2);
                               // อ่านค่าพอร์ตดิจิตอลขา 8 เก็บค่าที่ตัวแปร Colm3
colm3=digitalRead(C3);
                               // ถ้าตัวแปร com1 เท่ากับ 0
if(colm1==LOW) {
                               // แสดงเลข 1 บนจอคอมพิวเตอร์
        Serial.println("1"),
        delay(200);
                               // หน่วงเวลา 0.2 วินาที
        }
else {
                               // ถ้าตัวแปร coin2 เท่ากับ 0
if(colm2==LOW) {
                               // แสดงเลข 2 บนจอคอมพิวเตอร์
        Serial.println("2");
                               // หน่วงเวลา 0.2 วินาที
        delay(200);
        }
else {
                               // ถ้าตัวแปร colm3 เท่ากับ 0
if(colm3==LOW) {
                               // แสดงเลข 3 บนจอคอมพิวเตอร์
        Serial.println("3");
                               // หน่วงเวลา 0.2 วินาที
        delay(200);
        }
}
                               // ส่งข้อมูลออกพอร์ตดิจิตอลขา 5 เอาต์พุตเป็นลอจิก "1"
digitalWrite(r1,HIGH);
                               // ส่งข้อมูลออกพอร์ตดิจิตอลขา 4 เอาต์พุตเป็นลอจิก "0"
digitalWrite(r2,LOW);
                               // ส่งข้อมูลออกพอร์ตดิจิตอลขา 3 เอาต์พูตเป็นลอจิก "1"
digitalWrite(r3,HIGH);
                               // ส่งข้อมูลออกพอร์ตดิจิตอลขา 2 เอาต์พูตเป็นลอจิก "1"
digitalWrite(r4,HIGH);
                               // อ่านค่าพอร์ตดิจิตอลขา 6 เก็บค่าที่ตัวแปร cotm1
colm1=digitalRead(c1);
                               // อ่านค่าพอร์ตดิจิตอลขา 7 เก็บค่าที่ตัวแปร colm2
colm2=digitalRead(c2);
                               // อ่านค่าพอร์ตดิจิตอลขา 8 เก็บค่าที่ตัวแปร colm3
colm3=digitalRead(c3);
                               // ถ้าตัวแปร colnn1 เท่ากับ 0
if(colm1==LOW) {
                               // แสดงเลข 4 บนจอคอมพิวเตอร์
        Serial.println("4");
                               // หน่วงเวลา 0.2 วินาที
        delay(200);
```

}

วิทยาลัยเทคนิคชลบุรี

วิชา พื้นฐานไมโครคอนโทรลเลอร์

```
else {
                               // ถ้าตัวแปร colm2 เท่ากับ 0
if(colm2==LOW) {
                               // แสดงเลข 5 บนจอคอมพิวเตอร์
        Serial.println("5");
                               // หน่วงเวลา 0.2 วินาที
       delay(200);
       }
else {
                               // ถ้าตัวแปร colm3 เท่ากับ 0
if(colm3==LOW) {
                               // แสดงเลข 6 บนจอคอมพิวเตอร์
        Serial.println("6");
                               // หน่วงเวลา 0.2 วินาที
       delay(200);
       }
}
                               // ส่งข้อมูลออกพอร์ตดิจิตอลขา 5 เอาต์พุตเป็นลอจิก "1"
digitalWrite(r1,HIGH);
                               // ส่งข้อมูลออกพอร์ตดิจิตอลขา 4 เอาต์พุตเป็นลอจิก "1"
digitalWrite(r2,HIGH);
                               // ส่งข้อมูลออกพอร์ตดิจิตอลขา 3 เอาต์พุตเป็นลอจิก "0"
digitalWrite(r3,LOW);
                               // ส่งข้อมูลออกพอร์ตดิจิตอลขา 2 เอาต์พุตเป็นลอจิก "1"
digitalWrite(r4, HIGH);
                               // อ่านค่าพอร์ตดิจิตอลขา 6 เก็บค่าที่ตัวแปร com1
colm1=digitalRead(c1);
                               // อ่านค่าพอร์ตดิจิตอลขา 7 เก็บค่าที่ตัวแปร colm2
colm2=digitalRead(c2);
                               // อ่านค่าพอร์ตดิจิตอลขา 8 เก็บค่าที่ตัวแปร colm3
colm3=digitalRead(c3);
                               // ถ้าตัวแปร Cotm1 เท่ากับ 0
if(colm1==LOW) {
                               // แสดงเลข 7 บนจอคอมพิวเตอร์
        Serial.println("7");
                               // หน่วงเวลา 0.2 วินาที
       delay(200);
       }
else {
                               // ถ้าตัวแปร colm2 เท่ากับ 0
if(colm2==LOW) {
        Serial.println("8");
                               // แสดงเลข 8 บนจอคอมพิวเตอร์
                               // หน่วงเวลา 0.2 วินาที
       delay(200);
       }
else {
                               // ถ้าตัวแปร colm3 เท่ากับ 0
if(colm3==LOW) {
                               // แสดงเลข 9 บนจอคอมพิวเตอร์
        Serial.println("9");
```

วิทยาลัยเทคนิคชลบุรี

จัดทำโดย นายวิรุณ จิตต์บุญ

วิชา พื้นฐานไมโครคอนโทรลเลอร์

```
// หน่วงเวลา 0.2 วินาที
        delay(200);
        }
}
                               // ส่งข้อมูลออกพอร์ตดิจิตอลขา 5 เอาต์พุตเป็นลอจิก "1"
digitalWrite(r1,HIGH);
                               // ส่งข้อมูลออกพอร์ตดิจิตอลขา 4 เอาต์พุตเป็นลอจิก "1"
digitalWrite(r2,HIGH);
                               // ส่งข้อมูลออกพอร์ตดิจิตอลขา 3 เอาต์พุตเป็นลอจิก "1"
digitalWrite(r3,HIGH);
                               // ส่งข้อมูลออกพอร์ตดิจิตอลขา 2 เอาต์พุตเป็นลอจิก "0"
digitalWrite(r4,LOW);
                               // อ่านค่าพอร์ตดิจิตอลขา 6 เก็บค่าที่ตัวแปร colm1
colm1=digitalRead(c1);
                               // อ่านค่าพอร์ตดิจิตอลขา 7 เก็บค่าที่ตัวแปร cotm2
colm2=digitalRead(c2);
                               // อ่านค่าพอร์ตดิจิตอลขา 8 เก็บค่าที่ตัวแปร colm3
colm3=digitalRead(C3);
                               // ถ้าตัวแปร colm1 เท่ากับ 0
if(colm1==LOW) {
        Serial.println("*");
                               // แสดงเลข * บนจอคอมพิวเตอร์
                               // หน่วงเวลา 0.2 วินาที
        delay(200);
        }
else {
                               // ถ้าตัวแปร Colm2 เท่ากับ 0
if(colm2==LOW) {
                               // แสดงเลข 0 บนจอคอมพิวเตอร์
        Serial.println("0");
                               // หน่วงเวลา 0.2 วินาที
        delay(200);
        }
else {
                               // ถ้าตัวแปร colm3 เท่ากับ 0
if(colm3==LOW) {
        Serial.println("#");
                               // แสดงเลข # บนจอคอมพิวเตอร์
                               // หน่วงเวลา 0.2 วินาที
        delay(200);}
        }
}
```

}

58

ผลการทดลอง

การทดสอบสวิตช์เมทริกซ์	อธิบายลักษณะการแสดงผลที่จอคอมพิวเตอร์
ให้ทดลองกดสวิตช์เมทริกซ์	
หมายเลขต่าง ๆ	

โปรแกรมที่ 2 โปรแกรมการรับค่าสวิตซ์เมทริกซ์แสดงผลที่ LCD

ในโปรแกรมนี้ต่อวงจรสวิตช์เมทริกซ์ โดยการต่อวงจรใช้ขาสัญญาณเช่นเดียวกับโปรแกรมที่ 1 แต่ในส่วน ของการแสดงผลการทำงานของโปรแกรมใช้จอ LCD เพื่อดูผลการทำงานของโปรแกรม ให้ทำการต่อวงจรตามรูปที่ 9.4



รูปที่ 9.4 วงจรการทดลองโปรแกรมการรับค่าสวิตซ์เมทริกซ์แสดงผลที่ LCD

โปรแกรมการทดลอง

#include <LiquidCrystal.h> // ประกาศLibrary LiquidCrystal lcd(1, 13, 9, 10, 11, 12); // ฟังก์ชันแรกกำหนด Pins ที่ใช้ในการเชื่อมต่อ int r1 = 5; // กำหนดตัวแปร r1 = 5

วิทยาลัยเทคนิคชลบุรี

จัดทำโดย นายวิรุณ จิตต์บุญ

59

int r2 = 4;	// กำหนดตัวแปร r2 = 4
int r3 = 3;	// กำหนดตัวแปร r3 = 3
int r4 = 2	// กำหนดตัวแปร r4 = 2
int c1 = 6;	// กำหนดตัวแปร C1 = 6
int c2 = 7;	// กำหนดตัวแปร c2 = 7
int c3 = 8;	// กำหนดตัวแปร c3 = 8
int colm1;	// กำหนดตัวแปร colm1
int colm2;	// กำหนดตัวแปร colm2
int colm3;	// กำหนดตัวแปร colm3

void setup() {

// กำหนดขนาดโมดูล LCD ขนาด 16x2
// เซตพอร์ตดิจิตอลขา 5 ให้เป็นพอร์ตเอาต์พุต
// เซตพอร์ตดิจิตอลขา 4 ให้เป็นพอร์ตเอาต์พุต
// เซตพอร์ตดิจิตอลขา 3 ให้เป็นพอร์ตเอาต์พุต
// เชตพอร์ตดิจิตอลขา 2 ให้เป็นพอร์ตเอาต์พุต
// เซตพอร์ตดิจิตอลขา 6 ให้เป็นพอร์ตอินพุต
// เซตพอร์ตดิจิตอลขา 7 ให้เป็นพอร์ตอินพุต
// เซตพอร์ตดิจิตอลขา 8 ให้เป็นพอร์ตอินพุต
//ส่งข้อมูลออกพอร์ตดิจิตอลขา 6 เอาต์พุตเป็นลอจิก "1"
// ส่งข้อมูลออกพอร์ตดิจิตอลขา 7 เอาต์พุตเป็นลอจิก "1"
// ส่งข้อมูลออกพอร์ตดิจิตอลขา 8 เอาต์พุตเป็นลอจิก "1"
// กำหนดข้อความที่ต้องการแสดงผล
// กำหนดตำแหน่ง Cursor
// กำหนดข้อความที่ต้องการแสดงผล

} void loop () {

```
digitalWrite(r1,LOW);
digitalWrite(r2,HIGH);
digitalWrite(r3,HIGH);
```

```
// ส่งข้อมูลออกพอร์ตดิจิตอลขา 5 เอาต์พุตเป็นลอจิก "0"// ส่งข้อมูลออกพอร์ตดิจิตอลขา 4 เอาต์พุตเป็นลอจิก "1"
```

```
// ส่งข้อมูลออกพอร์ตดิจิตอลขา 3 เอาต์พุตเป็นลอจิก "1"
```

```
// ส่งข้อมูลออกพอร์ตดิจิตอลขา 2 เอาต์พุตเป็นลอจิก "1"
digitalWrite(r4,HIGH);
                               // อ่านค่าพอร์ตดิจิตอลขา 6 เก็บค่าที่ตัวแปร colm1
colm1=digitalRead(c1);
                               // อ่านค่าพอร์ตดิจิตอลขา 7 เก็บค่าที่ตัวแปร colm2
colm2=digitalRead(c2);
                               // อ่านค่าพอร์ตดิจิตอลขา 8 เก็บค่าที่ตัวแปร com3
colm3=digitalRead(c3);
                               // ถ้าตัวแปร colm1 เท่ากับ 0
if(colm1==LOW) {
                              // ลบข้อความบนหน้าจอ LCD ทั้งหมด
       lcd.clear();
       lcd.print("you press"); // กำหนดข้อความที่ต้องการแสดงผล
                              // กำหนดตำแหน่ง Cursor
       lcd.setCursor(0, 1);
                              // กำหนดข้อความที่ต้องการแสดงผล
       lcd.print("- 1 -");
       }
else {
                              // ถ้าตัวแปร colm2 เท่ากับ 0
if(colm2==LOW) {
                               // ลบข้อความบนหน้าจอ LCD ทั้งหมด
       lcd.clear();
       lcd.print("you press"); // กำหนดข้อความที่ต้องการแสดงผล
        lcd.setCursor(0, 1);
                              // กำหนดตำแหน่ง Cursor
                              // กำหนดข้อความที่ต้องการแสดงผล
       lcd.print("- 2 -");
       }
else {
                              // ถ้าตัวแปร colm3 เท่ากับ 0
if(colm3==LOW) {
                              // ลบข้อความบนหน้าจอ LCD ทั้งหมด
       lcd.clear();
       lcd.print("you press"); // กำหนดข้อความที่ต้องการแสดงผล
                              // กำหนดตำแหน่ง Cursor
       lcd.setCursor(0, 1);
                               // กำหนดข้อความที่ต้องการแสดงผล
       lcd.print("- 3 -");
       }
}
                               // ส่งข้อมูลออกพอร์ตดิจิตอลขา 5 เอาต์พุตเป็นลอจิก "1"
digitalWrite(r1, HIGH);
                               // ส่งข้อมูลออกพอร์ตดิจิตอลขา 4 เอาต์พุตเป็นลอจิก "0"
digitalWrite(r2,LOW);
                               // ส่งข้อมูลออกพอร์ตดิจิตอลขา 3 เอาต์พุตเป็นลอจิก "1"
digitalWrite(r3,HIGH):
                               // ส่งข้อมูลออกพอร์ตดิจิตอลขา 2 เอาต์พุตเป็นลอจิก "1"
digitalWrite(r4,HIGH);
```

วิทยาลัยเทคนิคชลบุรี

จัดทำโดย นายวิรุณ จิตต์บุญ

```
// อ่านค่าพอร์ตดิจิตอลขา 6 เก็บค่าที่ตัวแปร colm1
colm1=digitalRead(c1);
                              // อ่านค่าพอร์ตดิจิตอลขา 7 เก็บค่าที่ตัวแปร colm2
colm2=digitalRead(c2);
                              // อ่านค่าพอร์ตดิจิตอลขา 8 เก็บค่าที่ตัวแปร colm3
colm3=digitalRead(3);
                              // ถ้าตัวแปร colm1 เท่ากับ 0
if(colm1 = LOW) 
                              // ลบข้อความบนหน้าจอ LCD ทั้งหมด
       lcd.clear():
       lcd.print("you press"); // กำหนดข้อความที่ต้องการแสดงผล
                              // กำหนดตำแหน่ง Cursor
       lcd.setCursor(0, 1);
                              // กำหนดข้อความที่ต้องการแสดงผล
       lcd.print("- 4 -");
       }
else {
                              // ถ้าตัวแปร colm2 เท่ากับ 0
if(colm2==LOW) {
                              // ลบข้อความบนหน้าจอ LCD ทั้งหมด
       lcd.clear():
       lcd.print("you press"); // กำหนดข้อความที่ต้องการแสดงผล
        lcd.setCursor(0, 1); // กำหนดตำแหน่ง Cursor
                              // กำหนดข้อความที่ต้องการแสดงผล
        lcd.print("- 5 -");
       }
else {
                              // ถ้าตัวแปร colm3 เท่ากับ 0
if(colm3==LOW) {
                              // ลบข้อความบนหน้าจอ LCD ทั้งหมด
       lcd.clear();
       lcd.print("you press"); // กำหนดข้อความที่ต้องการแสดงผล
                              // กำหนดตำแหน่ง Cursor
       lcd.setCursor(0.1);
                              // กำหนดข้อความที่ต้องการแสดงผล
       lcd.print("- 6- ");
       }
}
                              // ส่งข้อมูลออกพอร์ตดิจิตอลขา 5 เอาต์พุตเป็นลอจิก "1"
digitalWrite(r1,HIGH);
                              // ส่งข้อมูลออกพอร์ตดิจิตอลขา 4 เอาต์พุตเป็นลอจิก "1"
digitalWrite(r2, HIGH);
                              // ส่งข้อมูลออกพอร์ตดิจิตอลขา 3 เอาต์พูตเป็นลอจิก "0"
digitalWrite(13,LOW);
                              // ส่งข้อมูลออกพอร์ติติจิตอลขา 2 เอาต์พุตเป็นลอจิก "1"
digitalWrite(r4,HIGH);
                              // อ่านค่าพอร์ตดิจิตอลขา 6 เก็บค่าที่ตัวแปร colm1
colm1=digitalRead(c1);
                               // อ่านค่าพอร์ตดิจิตอลขา 7 เก็บค่าที่ตัวแปร com2
colm2=digitalRead(c2);
```

วิทยาลัยเทคนิคชลบุรี

```
// อ่านค่าพอร์ตดิจิตอลขา 8 เก็บค่าที่ตัวแปร colm3
colm3=digitalRead(c3);
                             // ถ้าตัวแปร colm1 เท่ากับ 0
if(colm1==LOW) {
                             // ลบข้อความบนหน้าจอ LCD ทั้งหมด
       lcd.clear();
       lcd.print("you press"); // กำหนดข้อความที่ต้องการแสดงผล
                             // !กำหนดตำแหน่ง Cursor
       lcd.setCursor(0, 1);
                             // กำหนดข้อความที่ต้องการแสงผล
       lcd.print(" - 7 -");
       }
else {
                             // ถ้าตัวแปร colm2 เท่ากับ 0
if(colm2==LOW) {
                             // ลบข้อความบนหน้าจอ LCD ทั้งหมด
       lcd.clear();
       lcd.print("you press"); // กำหนดข้อความที่ต้องการแสดงผล
                             // กำหนดตำแหน่ง Cursor
       lcd.setCursor(0, 1):
                             // กำหนดข้อความที่ต้องการแสดงผล
       lcd.print("- 8 -");}
       }
else {
                             // ถ้าตัวแปร colm2 เท่ากับ 0
if(colm2==LOW) {
                             // ลบข้อความบนหน้าจอ LCD ทั้งหมด
       lcd.clear();
       lcd.print("you press"); // กำหนดข้อความที่ต้องการแสดงผล
                             // กำหนดตำแหน่ง Cursor
       lcd.setCursor(0, 1);
                             // กำหนดข้อความที่ต้องการแสดงผล
       (cd.print("- 0 -");
       }
else {
                             // ถ้าตัวแปร colm3 เท่ากับ 0
if(colm3==LOW) {
                             // ลบข้อความบนหน้าจอ LCD ทั้งหมด
       led.clear();
       lcd.print("you press"); // กำหนดข้อความที่ต้องการแสดงผล
                             // กำหนดตำแหน่ง Cursor
       lcd.setCursor(0, 1);
                             // กำหนดข้อความที่ต้องการแสดงผล
       lcd.print(" - # -");}
       }
}
```

วิทยาลัยเทคนิคชลบุรี

}

ผลการทดลอง

ทดสอบกดคีย์สวิตซ์	อธิบายลักษณะการทำงานของ LCD
กดสวิตช์เลข 0	
กดสวิตช์เลข 1	
กดสวิตช์เลข 2	
กดสวิตช์เลข 3	
กดสวิตช์เลข 4	
กดสวิตช์เลข 5	
กดสวิตช์เลข 6	
กดสวิตช์เลข 7	
กดสวิตช์เลข 8	
กดสวิตช์เลข 9	
กดสวิตช์เลข *	
กดสวิตช์เลข #	

สรุปผลการทดลอง

กิจกรรมตรวจสอบความเข้าใจใบงานที่ 9

- 1. จงเขียนโปรแกรมตามโจทย์ที่กำหนดให้คือ
- จากวงจรรูปที่ 9.5 ให้เขียนโปรแกรมการรับค่าคีย์สวิตซ์แบบเมทริกซ์โดยมีเงื่อนไข ดังนี้
 - ถ้ากดสวิตซ์เลข 0 ถูกกด ให้ LED1 ดับทั้งหมด
 - ถ้ากดสวิตซ์เลข 1 ถูกกด ให้ LED1 ติด
 - ถ้ากดสวิตซ์เลข 2 ถูกกด ให้ LED2 ติด
 - ถ้ากดสวิตช์เลข 3 ถูกกด ให้ LED2 ติด
 - ถ้ากดสวิตซ์เลข 4 ถูกกด ให้ LED4 ติด

วิทยาลัยเทคนิคชลบุรี


รูปที่ 9.5 วงจรการทดลองโปรแกรมการรับค่าสวิตซ์เมทริกซ์แสดงผลที่ LED

โปรแกรม