tiny.one/IoT-BSc facebook.com/somsacki



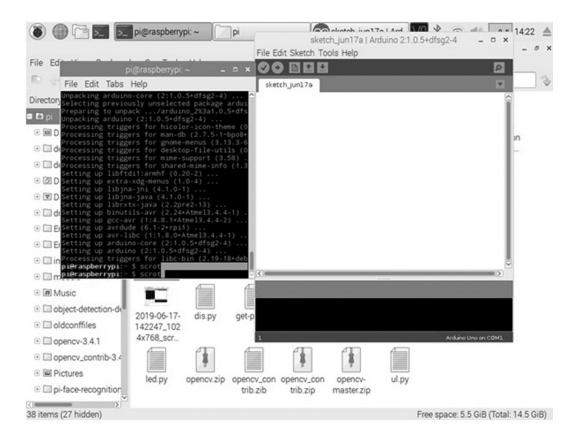
# Raspberry Pi และ Arduino

# 6.1 ติดตั้ง Arduino IDE uu Raspberry Pi

ข้อจำกัดของ Raspberry Pi คือไม่มีพอร์ตแอนะล็อกออนบอร์ด ซึ่งทำให้ไม่เหมาะสมสำหรับ ระบบที่จำเป็นต้องอ่านเซ็นเซอร์แอนะล็อก เพื่อเอาชนะข้อจำกัดนี้ สามารถติดตั้ง Arduino integrated development environment (IDE) บน Raspberry Pi ได้ เนื่องจาก Arduino มีพอร์ตแอนะล็อก ดังนั้นพอร์ตเหล่านี้จึงสามารถใช้เชื่อมต่อเซ็นเซอร์แอนะล็อกได้ การติดตั้ง Arduino IDE บน Raspberry Pi เป็นกระบวนการง่ายๆ ด้วยขั้นตอนง่ายๆ Arduino IDE พร้อมใช้งานสำหรับระบบปฏิบัติการส่วนใหญ่ แต่ที่นี่เราจะมาดูวิธีการติดตั้ง บน Raspberry Pi3 รุ่น B โดยใช้ Raspbian Jessie ในส่วนติดต่อผู้ใช้แบบกราฟิก (GUI)

- 1. ข้อกำหนดอันดับแรกคือการเชื่อมต่ออินเทอร์เน็ตที่ใช้งานได้
- 2. หน้าจอ คีย์บอร์ด และเมาส์ต้องเชื่อมต่อกับ Raspberry Pi
- ติดตั้ง Arduino IDE เวอร์ชันล่าสุดโดยใช้ apt: sudo apt-get update &&sudo apt-get upgrade sudo apt-get install arduino
- เชื่อมต่อบอร์ด Arduino กับ Raspberry Pi โดยใช้สายเคเบิลที่เหมาะสมแล้วดึง เมนูหลัก Raspbian และเลือก Arduino IDE ใต้หัว "Electronics" หน้าต่างว่าง จะเปิดขึ้นรูปที่ 6.1 แสดงหน้าต่างว่างสำหรับ Arduino IDE
- 5. คลิกที่ Tools > Board > และเลือกบอร์ด Arduino ที่เหมาะสม
- 6. ในการเลือกพอร์ตของ Arduino ที่เชื่อมต่อ ให้ตรวจสอบพอร์ตอนุกรมในเมนู " เครื่องมือ" ชื่อพอร์ตของ Arduino คือ: /dev/ ttyUSB0 หรือ /dev/ttyACM0.

# Internet of Things กับ Raspberry Pi และ Arduino



รูปที่ 6.1 หน้าต่างเปล่าของ Arduino IDE

### 6.2 เล่นกับเซนเซอร์ดิจิตอล

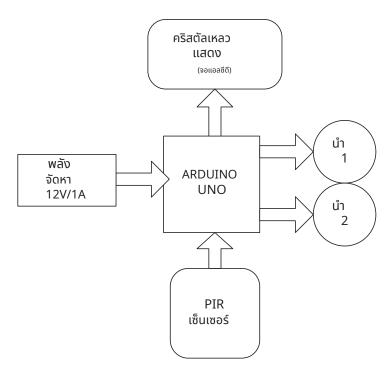
หลังจากติดตั้ง Arduino IDE บน Raspberry Pi แล้ว สามารถอ่านเซ็นเซอร์ที่เชื่อมต่อกับ Raspberry Pi และ Arduino ได้ Arduino สามารถทำหน้าที่เหมือนบอร์ด Arduino และ Raspberry Pi ทำหน้าที่เป็นคอมพิวเตอร์เมื่อเซ็นเซอร์เชื่อมต่อกับ Arduino สิ่งนี้สามารถ เข้าใจได้ด้วยความช่วยเหลือจากตัวอย่างบางส่วน

#### 6.2.1 เซ็นเซอร์ PIR

โมดลเซ็นเซอร์อินฟราเรดแบบไพโรอิเล็กทริก (PIR) ใช้สำหรับตรวจจับการเคลื่อนไหว มี ้ขนาดกะทัดรัดและใช้งานง่าย มีเลนส์ Fresnel และวงจรตรวจจับการเคลื่อนไหวซึ่งมีแรงดัน ้ไฟฟ้าหลากหลายที่ให้กระแสไฟน้อยกว่า มีความไวสูงและสัญญาณรบกวนต่ำ เอาต์พูตของ ้ เซ็นเซอร์คือสัญญาณทรานซิสเตอร์ทรานซิสเตอร์ตรรกะ (TTL) ต่ำ ตรวจจับการเคลื่อนไหว ้โดยการวัดการเปลี่ยนแปลงของระดับอินฟราเรดที่ปล่อยออกมาจากวัตถุที่อยู่รอบข้าง โมดูล นี้มีระยะการตรวจจับ 6 ม. และสามารถใช้กับสัญญาณกันขโมยและระบบควบคุมรูปที่ 6.2 ้ แสดงแผนภาพบล็อกของระบบที่ออกแบบมาเพื่อทำความเข้าใจการทำงานของเซ็นเซอร์ PIR ประกอบด้วย Arduino Uno,

#### Raspberry Pi และ Arduino

101



รูป 6.2 บล็อกไดอะแกรมสำหรับการเชื่อมต่อ PIR กับ Arduino

เซ็นเซอร์ PIR จอแสดงผลคริสตัลเหลว และ LED ระบบได้รับการออกแบบให้ "LED สีแดง" จะเป็น "ON" หากตรวจพบการเคลื่อนไหว มิฉะนั้น "BLUE LED" จะเป็น "ON"

#### 6.2.2 แผนภาพวงจร

เชื่อมต่อส่วนประกอบตามที่แสดงในร<mark>ูปที่6.3</mark>เพื่อตรวจสอบการทำงานของเซ็นเซอร์ PIR อัป โหลดโปรแกรมที่อธิบายไว้ในมาตรา 6.2.2และตรวจสอบการทำงาน

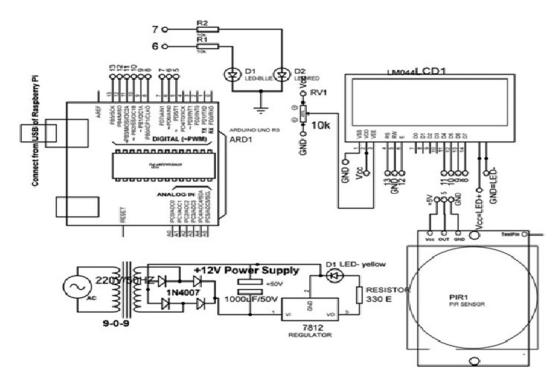
### การเชื่อมต่อเซ็นเซอร์ PIR

- เชื่อมต่อ Arduino GND กับโมดูล PIR GND
- เชื่อมต่อ Arduino +5 V กับโมดูล PIR +
- เชื่อมต่อขาดิจิตอล Arduino 2 เข้ากับขาสัญญาณดิจิตอลขาออกของโมดูล PIR

### การเชื่อมต่อ LCD

- เชื่อมต่อขาดิจิตอล Arduino (13) กับขา RS (4) ของ LCD
- เชื่อมต่อขาดิจิตอล Arduino (GND) กับขา RW (5) ของ LCD
- เชื่อมต่อขาดิจิตอล Arduino (12) กับขา E (6) ของ LCD
- เชื่อมต่อขาดิจิตอล Arduino (11) กับขา D4 (11) ของ LCD
- เชื่อมต่อขาดิจิตอล Arduino (10) กับขา D5 (12) ของ LCD

# Internet of Things กับ Raspberry Pi และ Arduino



รูปที่6.3 แผนภาพวงจรสำหรับการเชื่อมต่อ PIR กับ Arduino

- เชื่อมต่อขาดิจิตอล Arduino (9) กับขา D6 (13) ของ LCD
- เชื่อมต่อขาดิจิตอล Arduino (8) กับขา D7 (14) ของ LCD

# การเชื่อมต่อ LED

- เชื่อมต่อขาดิจิตอล Arduino 7 กับข้วบวกของ LED สีแดงผ่านตัวต้านทาน
- เชื่อมต่อขาดิจิตอล Arduino 6 กับข้วบวก BLUE-LED ผ่านตัวต้านทาน 330-โอห์ม
- เชื่อมต่อแคโทดของ LED ทั้งสองดวงกับกราวด์

#### 6.2.3 ร่าง

```
# รวม <LiquidCrystal.h> // รวมไลบรารีของ LCD
LiquidCrystallcd (13, 12, 11, 10,9, 8); // ติดขา LCD RS,E,D4,D5,D6,D7
  ไปยังหมุดที่กำหนด
int PIR_SENSOR_LOW=5; // กำหนดพิน 5 เป็น PIR_SENSOR_LOW int
RED_LED=7; // กำหนดพิน 7 เป็น RED_LED
int BLUE LED=6; // // กำหนดพิน 6 เป็นการตั้งค่าโมฆะ
BLUE_LED ()
{
```

tiny.one/IoT-BSc

facebook.com/somsacki

Raspberry Pi และ Arduino

```
โหมดพิน (PIR_SENSOR_LOW, INPUT_PULLUP); // congure pin5 เป็น
  อินพุตและเปิดใช้งานตัวต้านทานแบบดึงขึ้นภายใน
โหมดพิน (RED_LED, OUTPUT); // congure pin7 เป็นเอาต์พุต pinMode
(BLUE LED, OUTPUT); // conqure pin6 เป็นเอาต์พต lcd.begin(20, 4); //
้ตั้งค่าจำนวนคอลัมน์และแถวของ LCD lcd.setCursor(0, 0); // ตั้งค่าเคอร์เซอร์
เป็น column0 และ row1 lcd.print("MOTION SENSOR BASED"); // พิมพ์
ข้อความไปยัง LCD lcd.setCursor(0, 1); // ตั้งค่าเคอร์เซอร์เป็น column0 และ
row1
lcd.print("การตรวจจับการเคลื่อนไหว"); // พิมพ์ข้อความไปยัง LCD
lcd.setCursor(0, 2); // ตั้งค่าเคอร์เซอร์เป็น column0 และ row2
lcd.print("ระบบที่ LPU"); // พิมพ์ข้อความไปยัง LCD ล่าซ้า (1000);
}
วงเป็นโมฆะ ()
int PIR_SENSOR_LOW_READ = digitalRead (PIR_SENSOR_LOW);
  // อ่านค่า PIR เป็นตัวแปร
if (PIR SENSOR LOW READ == LOW) // อ่าน PIN 5 เป็น PIN ต่ำ
lcd.clear(); // ล้างเนื้อหาของ LCD lcd.setCursor(0, 3); // ตั้งค่าเคอร์เซอร์
เป็น column0 และ row2 lcd.print("MOTION DETECTED"); // พิมพ์
้ข้อความไปยัง LCD digitalWrite (RED_LED, สูง); // สร้าง pin7 เป็น
HIGH digitalWrite (BLUE_LED, LOW); // ทำให้ pin6 เป็น LOW ล่าซ้า
(20); // ล่าช้า 20 mS
 }
อื่น // มิฉะนั้น
 {
lcd.clear(); // ล้างเนื้อหาของ LCD lcd.setCursor(0, 3); // ตั้งค่าเคอร์เซอร์เป็น
column0 และ row3 lcd.print("ไม่พบการเคลือนไหว"); // พิมพ์ข้อความไปยัง
LCD digitalWrite (BLUE_LED สูง); // สร้างพิน 7 เป็น HIGH digitalWrite
(RED_LED, LOW); // pin6 ต่ำถึง LOW
ล่าช้า (20); // ล่าช้า 20 mS
 }
}
```

# Internet of Things กับ Raspberry Pi และ Arduino

# 6.3 เล่นด้วยเซนเซอร์อนาล็อก

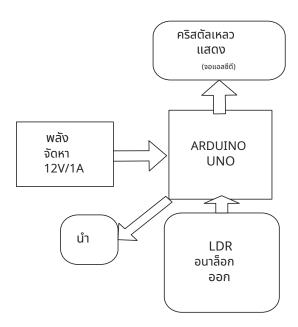
หากต้องการอ่านเซ็นเซอร์อะนาล็อกด้วย Arduino เพียงเชื่อมต่อเซ็นเซอร์กับพินอะนาล็อก ของบอร์ด เพื่อทำความเข้าใจการทำงานของเซ็นเซอร์แอนะล็อก จะมีการอธิบายตัวอย่างตัว ต้านทานแบบพึงพาแสง (LDR) ไว้ที่นี่ LDR มีเซลล์โฟโตคอนดักเตอร์แคดเมียมซัลไฟด์ (CdS) ที่มีการตอบสนองทางสเปกตรัม ความต้านทานของเซลล์ลดลงเมื่อความเข้มของแสง เพิ่มขึ้น LDR สามารถใช้ได้กับหลากหลายแอพพลิเคชั่น เช่น การตรวจจับควัน การควบคุม แสงอัตโนมัติ การนับแบทช์ และระบบสัญญาณกันขโมยรูปที่6.4แสดงแผนภาพบล็อกเพื่อ เชื่อมต่อ LDR กับ Arduino ประกอบด้วย Arduino Uno พาวเวอร์ซัพพลาย จอแสดงผล คริสตัลเหลว และ LDR ระบบถูกออกแบบมาเพื่อแสดงความเข้มแสงของ LCD

### 6.3.1 แผนภาพวงจร

เชื่อมต่อส่วนประกอบตามที่แสดงในร<mark>ูปที่6.5</mark>เพื่อตรวจสอบการทำงานของเซ็นเซอร์ LDR เป็นเซ็นเซอร์แอนะล็อกอย่างง่าย LDR นี้มีขัวต่อสามขัว: กราวด์, Vcc, OUT อัปโหลด โปรแกรมที่อธิบายไว้ในมาตรา 6.3.2และตรวจสอบการทำงาน

# การเชื่อมต่อเซ็นเซอร์แสง

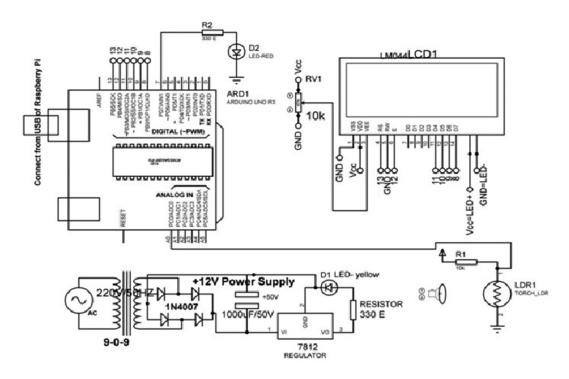
- เชื่อมต่อ Arduino GND กับโมดูล LDR GND
- เชื่อมต่อ Arduino +5 V กับโมดูล LDR +
- ต่อขา Arduino A0 เข้ากับขา OUT ของเซ็นเซอร์



รูป 6.4 บล็อกไดอะแกรมเพื่อเชื่อมต่อ LDR กับ Arduino

#### Raspberry Pi และ Arduino

105



รูปที่6.5 แผนภาพวงจรสำหรับเซ็นเซอร์ LDR ทีเชื่อมต่อกับ Arduino

### การเชื่อมต่อ LCD

- เชื่อมต่อขาดิจิตอล Arduino 13 กับขา RS (4) ของ LCD
- เชื่อมต่อ Arduino digital pin GND กับ RW pin (5) ของ LCD
- ต่อขาดิจิตอล Arduino 12 เข้ากับขา E (6) ของ LCD
- เชื่อมต่อขาดิจิตอล Arduino 11 กับขา D4 (11) ของ LCD
- เชื่อมต่อขาดิจิตอล Arduino 10 ถึง D5 ขา (12) ของ LCD
- เชื่อมต่อขาดิจิตอล Arduino 9 กับขา D6 (13) ของ LCD
- เชื่อมต่อขาดิจิตอล Arduino 8 กับขา D7 (14) ของ LCD

### 6.3.2 ร่าง

# รวม <LiquidCrystal.h> // รวมไลบรารีของ LCD

LiquidCrystallcd (13, 12, 11, 10, 9, 8); // ติดขา LCD RS,E,D4,D5,D6,D7 ไปยังหมุดที่กำหนด

intLDR\_sensor\_Pin = A0; // เลือกพินอินพุตสำหรับโพเทนชิออมิเตอร์

intLDR\_sensor\_ADC\_Value = 0; // ตัวแปรเก็บค่าที่กำลังมา จากเซ็นเซอร์

int RED\_LED=7; // กำหนดพิน 7 ให้กับ RED\_LED

tiny.one/IoT-BSc facebook.com/somsacki

106

# Internet of Things กับ Raspberry Pi และ Arduino

```
การตั้งค่าเป็นโมฆะ ()
lcd.begin(20, 4); // เริ่มต้นโหมดพิน LCD ขนาด 20*4 (RED_LED,
OUTPUT); // ใช้ RED_LED เป็นเอาต์พุต lcd.setCursor(0, 0); // ตั้ง
ค่าเคอร์เซอร์ของ LCD ที่ column0 และ Row0 lcd.print ("ไฟตาม
LDR"); // พิมพ์สตริงบน LCD lcd.setCursor(0, 1); // ตั้งเคอร์เซอร์บน
LCD
lcd.print("การตรวจสอบความเข้ม"); // พิมพ์สตริงบน LCD
lcd.setCursor(0, 2); // ตั้งเคอร์เซอร์บน LCD
lcd.print("ระบบที่ LPU"); // พิมพ์สตริงบน LCD ล่าซ้า
(1000); // หน่วงเวลา 1,000 mS
lcd.clear(); // ล้างเนื้อหาของ LCD }
วงเป็นโมฆะ ()
{
LDR_sensor_ADC_Value = analogRead (LDR_sensor_Pin); // อ่าน
  ค่าจากเซ็นเซอร์
lcd.setCursor(0,2); // ตั้งค่าเคอร์เซอร์บน LCD lcd.print("ADC
LEVEL+LDR:"); // พิมพ์สตริงบน LCD lcd.setCursor(17,2); //
ตั้งเคอร์เซอร์บน LCD
lcd.print (LDR_sensor_ADC_Value); // // พิมพ์ค่าบน LCD ถ้า
(LDR_sensor_ADC_Value>=100)
digitalWrite (RED_LED สูง); // ทำให้ pin7 เป็นความล่าช้าสูง
(20); // ล่าช้า 20 mS
 }
อื่น
digitalWrite (RED_LED ต่ำ); // ทำให้ pin7 เป็นความล่าช้าสูง
(20); // ล่าช้า 20 mS
 }
}
```

# 6.4 เล่นกับตัวกระตุ้น

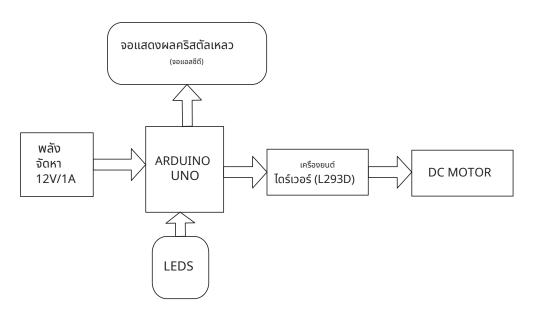
แอคทูเอเตอร์เป็นส่วนประกอบของเครื่องจักรที่ใช้สำหรับการเคลื่อนย้ายและควบคุมกลไก หรือระบบ มอเตอร์กระแสตรง สเต็ปเปอร์มอเตอร์ และเซอร์โวมอเตอร์มักใช้แอคทูเอเตอร์ใน ระบบ

#### 6.4.1 มอเตอร์กระแสตรง

โดยทั่วไปแล้ว มอเตอร์เกียร์ DC ที่มี 100 รอบต่อนาที่ 12 V มักใช้สำหรับการใช้งานด้าน วิทยาการหุ่นยนต์ ใช้งานง่ายมาก พวกเขามีน็อตและเกลียวบนเพลาเพื่อให้เชื่อมต่อได้ง่าย และเพลาเกลียวในเพื่อให้เชื่อมต่อกับล้อได้ง่ายรูปที่6.6แสดงบล็อกไดอะแกรมเพื่อเชื่อมต่อ มอเตอร์กระแสตรงกับ Arduino ประกอบด้วย Arduino Uno, แหล่งจ่ายไฟ, จอแสดงผล คริสตัลเหลว, ไดรเวอร์มอเตอร์ (L293D) และมอเตอร์กระแสตรงสองตัว

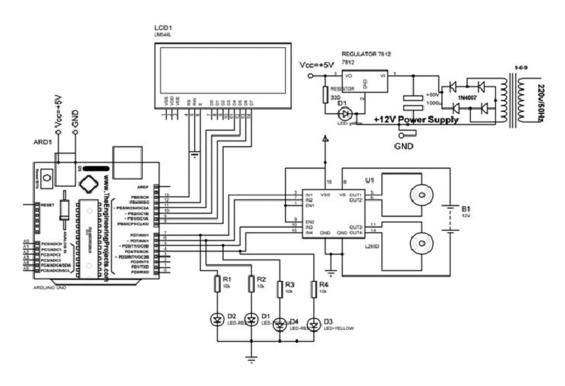
#### 6.4.1.1 แผนภาพวงจร

เชื่อมต่อส่วนประกอบตามที่แสดงในรู<mark>ปที่6.7</mark>เพื่อตรวจสอบการทำงานของมอเตอร์ กระแสตรง อัปโหลดโปรแกรมที่อธิบายไว้ในมาตรา 6.4.1.2และตรวจสอบการทำงาน



รูป 6.6 บล็อกไดอะแกรมของมอเตอร์กระแสตรงที่เชื่อมต่อกับ Arduino

# Internet of Things กับ Raspberry Pi และ Arduino



รูป 6.7 แผนภาพวงจรของมอเตอร์กระแสตรงทีเชื่อมต่อกับ Arduino

# การเชื่อมต่อมอเตอร์ L293D และ DC

- ต่อ L293D pin 3 กับ +ve pin ของ DC motor1.
- ต่อ L293D pin 6 เข้ากับ –ve pin ของ DC motor1.
- ต่อ L293D pin 11 ถึง +ve pin ของ DC motor2.
- ต่อ L293D pin 14 to +ve pin ของ DC motor2. การเชื่อมต่อ L293D
  - เชื่อมต่อ Arduino GND กับพิน 4, 5, 12, 13 ของ L293D
  - เชื่อมต่อ Arduino +5 V กับพิน 1, 9, 16 ของ L293D
  - ต่อ Arduino pin 7 กับ pin 2 ของ L293D
  - ต่อ Arduino pin 6 กับ pin 7 ของ L293D
  - ต่อ Arduino pin 5 กับ pin 10 ของ L293D
  - ต่อ Arduino pin 4 กับ pin 15 ของ L293D
- เชื่อมต่อ L293D pin 8 กับ +ve ของแบตเตอรี่ 12V การเชื่อมต่อ LED
  - เชื่อมต่อ Arduino pin 7 กับข้วบวกของ LED1
  - เชื่อมต่อ Arduino pin 6 กับข้วบวกของ LED2
  - เชื่อมต่อ Arduino pin 5 กับข้วบวกของ LED3

Raspberry Pi และ Arduino

- เชื่อมต่อ Arduino pin 4 กับข้วบวกของ LED4
- เชื่อมต่อแคโทดของ LED ทั้งหมดกับกราวด์

### การเชื่อมต่อ LCD

- เชื่อมต่อขาดิจิตอล Arduino 13 กับขา RS (4) ของ LCD
- เชื่อมต่อ Arduino digital pin GND กับ RW pin (5) ของ LCD
- ต่อขาดิจิตอล Arduino 12 เข้ากับขา E (6) ของ LCD
- เชื่อมต่อขาดิจิตอล Arduino 11 กับขา D4 (11) ของ LCD
- เชื่อมต่อขาดิจิตอล Arduino 10 ถึง D5 ขา (12) ของ LCD
- เชื่อมต่อขาดิจิตอล Arduino 9 กับขา D6 (13) ของ LCD
- เชื่อมต่อขาดิจิตอล Arduino 8 กับขา D7 (14) ของ LCD

### 6.4.1.2 ร่าง

```
# รวม <LiquidCrystal.h> // รวมไลบรารีของ LCD
LiquidCrystallcd (13, 12, 11, 10,9, 8); // ติดขา LCD RS, E, D4, D5, D6,
   D7 ไปยังหมุดที่กำหนด
int MPIN1= 7; // กำหนดพิน 7 เป็น MPIN1
int MPIN2= 6; // กำหนดพิน 6 เป็น MPIN2
int MPIN3= 5; // กำหนดพิน 5 เป็น MPIN3
int MPIN4= 4; // กำหนดพิน 4 เป็น MPIN4
การตั้งค่าเป็นโมฆะ ()
์โหมดพิน (MPIN1, เอาต์พุต); // ทำให้ MPIN1 เป็นเอาต์พุต
pinMode (MPIN2, OUTPUT); // ทำให้ MPIN2 เป็นเอาต์พุต
pinMode (MPIN3, OUTPUT); // ทำให้ MPIN3 เป็นเอาต์พูต
pinMode (MPIN4, OUTPUT); // ทำให้ MPIN4 เป็นเอาต์พุต
lcd.begin(20,4); // ເริ່ມຕໍ່u LCD
lcd.setCursor(0, 0); // ตั้งเคอร์เซอร์บน LCD lcd.print ("ทิศทาง
มอเตอร์ DC"); // พิมพ์สตริงบน LCD lcd.setCursor(0, 1); // ตั้ง
เคอร์เซอร์บน LCD
lcd.print("ระบบควบคุม..."); // พิมพ์สตริงบน LCD ล่าช้า
(1000); // หน่วงเวลา 1,000 mS
lcd.clear(); // ล้างเนื้อหาของ LCD }
วงเป็นโมฆะ () // innite loop {
digitalWrite (MPIN1, สูง); // ทำให้ MPIN1 เป็น HIGH
```

Internet of Things กับ Raspberry Pi และ Arduino

#### 110

digitalWrite (MPIN2, ต่ำ); // ทำให้ MPIN2 เป็น LOW digitalWrite (MPIN3, สูง); // ทำให้ MPIN3 เป็น high digitalWrite (MPIN4, LOW); // ทำให้ MPIN4 เป็น LOW lcd.setCursor(0, 2); // ตั้งเคอร์เซอร์บน LCD lcd.print("ตามเข็มนาฬิกา"); // พิมพ์สตริงบน LCD ล่าช้า (2000); // ล่าช้า 2 วินาที lcd.clear(); // ล้างเนื้อหาของ LCD digitalWrite (MPIN1, LOW); // ทำให้ MPIN1 เป็น LOW digitalWrite (MPIN2, สูง); // ทำให้ MPIN2 เป็น high digitalWrite (MPIN3, LOW); // ทำให้ MPIN3 เป็น LOW digitalWrite (MPIN4, สูง); //ทำให้ MPIN4 เป็น HIGH lcd.setCursor(0, 2); // ตั้ง เคอร์เซอร์บน LCD lcd.print("ทวนเข็มนาฬิกา"); // พิมพ์สตริงบน LCD ล่าช้า (2000); // ล่าช้า 2 วินาที lcd.clear(); // ล้างเนื้อหาของ LCD digitalWrite (MPIN1, LOW); // ทำให้ MPIN1 เป็น LOW digitalWrite (MPIN2, LOW); // ทำให้ MPIN2 เป็น LOW digitalWrite (MPIN3, สูง); // ทำให้ MPIN3 เป็น digitalWrite สูง (MPIN4, LOW); //ทำให้ MPIN4 เป็น LOW lcd.setCursor(0, 2); // ตั้งเคอร์เซอร์บน LCD lcd.print("ซ้าย"); // พิมพ์สตริงบน LCD ล่าช้า (2000); // ล่าช้า 2 วินาที lcd.clear(); // ล้างเนื้อหาของ LCD digitalWrite (MPIN1, สูง); // ทำให้ MPIN1 เป็น high digitalWrite (MPIN2, LOW); // ทำให้ MPIN2 เป็น LOW digitalWrite (MPIN3, LOW); // ทำให้ MPIN3 เป็น LOW digitalWrite (MPIN4, LOW); //ทำให้ MPIN4 เป็น LOW lcd.setCursor(0, 2); // ตั้งเคอร์เซอร์บน LCD lcd.print("ขวา"); // พิมพ์สตริงบน LCD ล่าช้า (2000); // ล่าช้า 2 วินาที lcd.clear(); // ล้างเนื้อหาของ LCD

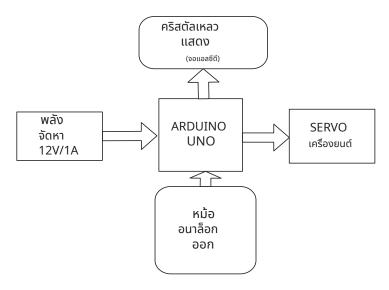
#### 6.4.2 เซอร์โวมอเตอร์

}

เซอร์โวมอเตอร์คือแอคชูเอเตอร์แบบโรตารีที่ใช้สำหรับการควบคุมตำแหน่งเชิงมุมอย่าง แม่นยำ ประกอบด้วยมอเตอร์ควบคู่กับเซ็นเซอร์สำหรับ

#### Raspberry Pi และ Arduino

111



รูป 6.8 บล็อกไดอะแกรมเพื่อเชื่อมต่อเซอร์โวมอเตอร์กับ Arduino

ข้อเสนอแนะตำแหน่ง นอกจากนี้ยังต้องใช้ไดรฟ์เซอร์โว ไดรฟ์ใช้เซ็นเซอร์ป้อนกลับเพื่อ ควบคุมตำแหน่งการหมุนของมอเตอร์อย่างแม่นยำ สิ่งนี้เรียกว่าการดำเนินการแบบวงปิด เซอร์โวมอเตอร์มาตรฐานแรงบิดสูงพร้อมเฟืองโลหะและ360°การหมุนสามารถให้ 11 กก./ ซม. ที่ 4.8 V, 13.5 กก./ซม. ที่ 6 V และ 16 กก./ซม. ที่ 7.2 V.รูปที่6.8แสดงบล็อก ไดอะแกรมเพื่อเชื่อมต่อเซอร์โวมอเตอร์กับ Arduino ประกอบด้วย Arduino Uno, แหล่ง จ่ายไฟ, จอแสดงผลคริสตัลเหลว, โพเทนชิออมิเตอร์ (POT) และเซอร์โวมอเตอร์ ระบบถูก ออกแบบมาเพื่อควบคุมมุมของเซอร์โวมอเตอร์ด้วยโพเทนชิออมิเตอร์

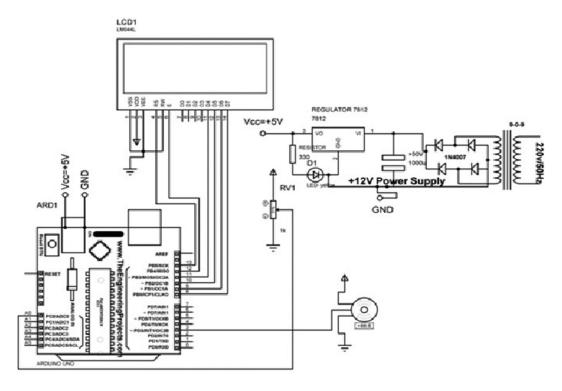
### 6.4.2.1 แผนภาพวงจร

เชื่อมต่อส่วนประกอบตามที่แสดงในร<mark>ูปที่ 6.9</mark>เพื่อตรวจสอบการทำงานของเซอร์โวมอเตอร์ อัปโหลดโปรแกรมที่อธิบายไว้ในมาตรา 6.4.2.2และตรวจสอบการทำงาน

### การเชื่อมต่อเซอร์โว

- เชื่อมต่อ Arduino GND กับขา GND ของเซอร์โวมอเตอร์
- ต่อ Arduino +5 V กับข้ว "+" ของเซอร์โวมอเตอร์
- ต่อขา Arduino (3) เข้ากับขา PWM ของเซอร์โวมอเตอร์ การเชื่อมต่อหม้อ
  - เชื่อมต่อ Arduino GND กับพิน GND ของ POT
  - ต่อ Arduino +5 V กับข้ว "+" ของ POT
  - ต่อขา Arduino A0 เข้ากับขา data out ของ POT

# Internet of Things กับ Raspberry Pi และ Arduino



รูป 6.9 แผนภาพวงจรเพื่อเชื่อมต่อเซอร์โวมอเตอร์กับ Arduino

# การเชื่อมต่อ LCD

- เชื่อมต่อขาดิจิตอล Arduino (13) กับขา RS (4) ของ LCD
- เชื่อมต่อขาดิจิตอล Arduino (GND) กับขา RW (5) ของ LCD
- เชื่อมต่อขาดิจิตอล Arduino (12) กับขา E (6) ของ LCD
- เชื่อมต่อขาดิจิตอล Arduino (11) กับขา D4 (11) ของ LCD
- เชื่อมต่อขาดิจิตอล Arduino (10) กับขา D5 (12) ของ LCD
- เชื่อมต่อขาดิจิตอล Arduino (9) กับขา D6 (13) ของ LCD
- เชื่อมต่อขาดิจิตอล Arduino (8) กับขา D7 (14) ของ LCD

# 6.4.2.2 ร่าง

# รวม <LiquidCrystal.h> // รวมไลบรารีของ LCD

LiquidCrystallcd (13, 12, 11, 10, 9, 8); // ติดขา LCD RS,E,D4,D5,D6,D7 ไปยังหมุดที่กำหนด

เซอร์โว myservo; // สร้างวัตถุเซอร์โวเพื่อควบคุมเซอร์โว int POT\_PIN = A0; // พินแอนะล็อกที่ใช้ต่อโพเทนชิออมิเตอร์ ใน POT\_PIN\_ADC\_LEVEL; // ตัวแปรที่จะอ่านค่าจาก เข็มอนาล็อก

# Raspberry Pi และ Arduino

```
การตั้งค่าเป็นโมฆะ ()
myservo.attach(3); // ติดเซอร์โวบนพิน 9 กับวัตถุเซอร์โว
lcd.begin(20,4); // เริ่มต้น LCD
lcd.setCursor(0, 0); // ตั้งค่าเคอร์เซอร์บน LCD lcd.print("Servo
ANALOG write"); // พิมพ์สตริงบน LCD lcd.setCursor(0, 1); //
ตั้งค่าเคอร์เซอร์บน LCD
lcd.print("system at LPU...."); // พิมพ์สตริงบน LCD }
วงเป็นโมฆะ ()
{
POT_PIN_ADC_LEVEL = analogRead (POT_PIN); // อ่านค่า POT
  ในรูปแบบของระดับ
POT_PIN_ADC_LEVEL = แผนที่ (POT_PIN_ADC_LEVEL, 0, 1023, 0, 179);
  // แมปค่า // ระหว่าง 0 ถึง 180 องศาสำหรับเซอร์โว
myservo.write(POT_PIN_ADC_LEVEL); // กำหนดตำแหน่งเซอร์โว
  ตามค่ามาตราส่วน
lcd.setCursor(0, 2); // ตั้งค่าเคอร์เซอร์บน LCD
lcd.print("ANGLE:"); // พิมพ์สตริงบน LCD lcd.print
(POT_PIN_ADC_LEVEL); // พิมพ์ค่าบน LCD ล่าซ้า (15); // ล่าซ้า
15 mSec
}
```