

6

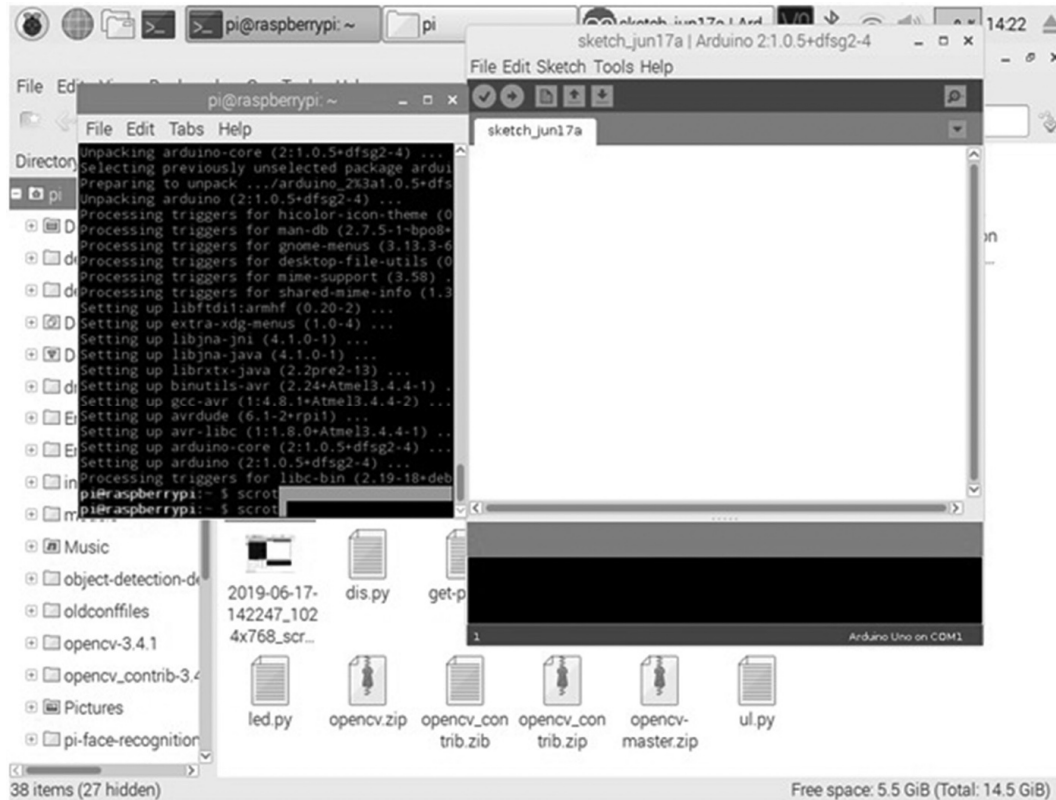
Raspberry Pi и Arduino

6.1 ប្រើប្រាស់ Arduino IDE ឬ Raspberry Pi

ข้อจำกัดของ Raspberry Pi คือไม่มีพอร์ตแอนด์ล็อกออนบอร์ด ซึ่งทำให้ไม่เหมาะสมสำหรับระบบที่จำเป็นต้องอ่านเซ็นเซอร์แอนด์ล็อก เพื่อเอาชนะข้อจำกัดนี้ สามารถติดตั้ง Arduino integrated development environment (IDE) บน Raspberry Pi ได้ เนื่องจาก Arduino มีพอร์ตแอนด์ล็อก ดังนั้นพอร์ตเหล่านี้จึงสามารถใช้เชื่อมต่อเซ็นเซอร์แอนด์ล็อกได้ การติดตั้ง Arduino IDE บน Raspberry Pi เป็นกระบวนการง่ายๆ ด้วยขั้นตอนง่ายๆ Arduino IDE พร้อมใช้งานสำหรับระบบปฏิบัติการส่วนใหญ่ แต่ที่นี่เราจะมาดูวิธีการติดตั้งบน Raspberry Pi3 รุ่น B โดยใช้ Raspbian Jessie ในส่วนติดต่อผู้ใช้แบบกราฟิก (GUI)

1. ข้อกำหนดอันดับแรกคือการเชื่อมต่ออินเทอร์เน็ทที่ใช้งานได้
2. หน้าจอ คีย์บอร์ด และเมาส์ต้องเชื่อมต่อกับ Raspberry Pi
3. ติดตั้ง Arduino IDE เวอร์ชันล่าสุดโดยใช้ apt:

```
sudo apt-get update &&sudo apt-get upgrade  
sudo apt-get install arduino
```
4. เชื่อมต่อบอร์ด Arduino กับ Raspberry Pi โดยใช้สายเคเบิลที่เหมาะสมแล้วดึงเมนูหลัก Raspbian และเลือก Arduino IDE ได้หัว "Electronics" หน้าต่างว่างจะเปิดขึ้น**รูปที่ 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.



รูปที่ 6.1

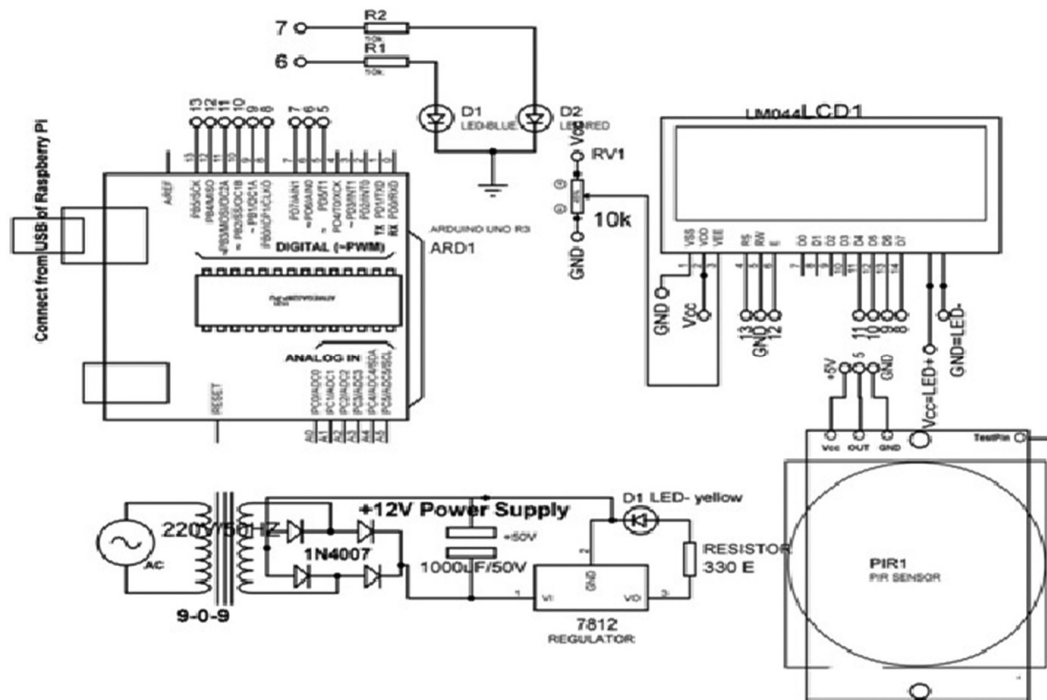
หน้าต่างเปล่าของ Arduino IDE

6.2 เล่นกับเซนเซอร์ดิจิทัล

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

6.2.1 ឃើនឆ័រ PIR

โมดูลเซ็นเซอร์อินฟราเรดแบบไพโรอิเล็กทริก (PIR) ใช้สำหรับตรวจจับการเคลื่อนไหว มีขนาดเล็กกระทัดรัดและใช้งานง่าย มีเลนส์ Fresnel และวงจรตรวจจับการเคลื่อนไหวซึ่งมีแรงดันไฟฟ้าหลากหลายที่ให้กระแสไฟน้อยกว่า มีความไวสูงและสัญญาณรบกวนต่ำ เอาต์พุตของเซ็นเซอร์คือสัญญาณทรานซิสเตอร์ทรานซิสเตอร์ตรรกะ (TTL) ต่ำ ตรวจจับการเคลื่อนไหวโดยการวัดการเปลี่ยนแปลงของระดับอินฟราเรดที่ปล่อยออกมาจากวัตถุที่อยู่รอบข้าง โมดูลนี้มีระยะเวลาตรวจจับ 6 ม. และสามารถใช้กับสัญญาณกันขโมยและระบบควบคุม [รูปที่ 6.2](#) แสดงแผนภาพบล็อกของระบบที่ออกแบบมาเพื่อทำความเข้าใจการทำงานของเซ็นเซอร์ PIR ประกอบด้วย Arduino Uno,



รูปที่ 6.3

แผนภาพวงจรสำหรับการเชื่อมต่อ PIR กับ Arduino

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

การเชื่อมต่อ LED

- เชื่อมต่อขาจิจิตอล Arduino 7 กับขั้วบวกของ LED สีแดงผ่านตัวต้านทาน 330 โอห์ม
- เชื่อมต่อขาจิจิตอล Arduino 6 กับขั้วบวก BLUE-LED ผ่านตัวต้านทาน 330-โอห์ม
- เชื่อมต่อแคโทดของ LED ทั้งสองดวงกับกราวด์

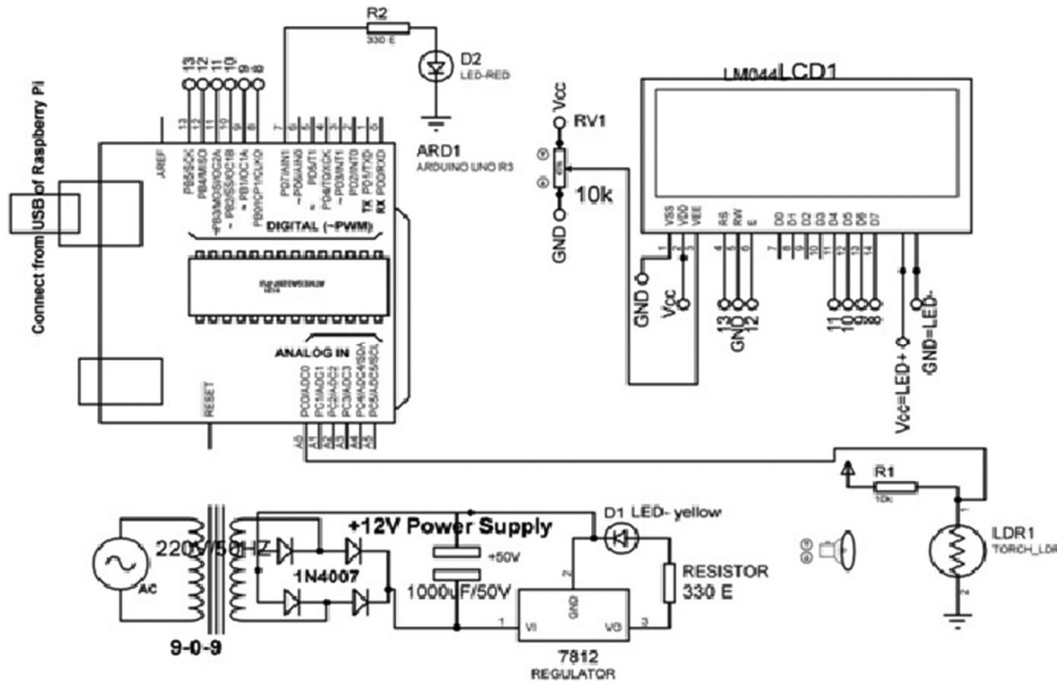
6.2.3 ກ່າວ

```
# รวม <LiquidCrystal.h> // รวมไลบรารีของ LCD
LiquidCrystal lcd (13, 12, 11, 10,9, 8); // ติดต่อ LCD RS,E,D4,D5,D6,D7
// ยังหมุดที่กำหนด

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 เป็นการตั้งค่าโมดูล
BLUE_LED ()

{
```


Internet of Things กับการ Raspberry Pi และ Arduino



รูปที่ 6.5

แผนภาพวงจรสำหรับเซ็นเซอร์ LDR ที่เชื่อมต่อกับ Arduino

การเชื่อมต่อ LCD

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

6.3.2 ກ່າວ

```
# รวม <LiquidCrystal.h> // รวมไลบรารีของ LCD
```

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

```
intLDR_sensor_Pin = A0; // เลือกพินอินพุตสำหรับโพเทนชิอมิเตอร์
```

```
intLDR_sensor_ADC_Value = 0; // ตัวแปรเก็บค่าที่กำลังมา  
จากเซ็นเซอร์
```

```
int RED_LED=7; // กำหนดพิน 7 ให้กับ RED_LED
```

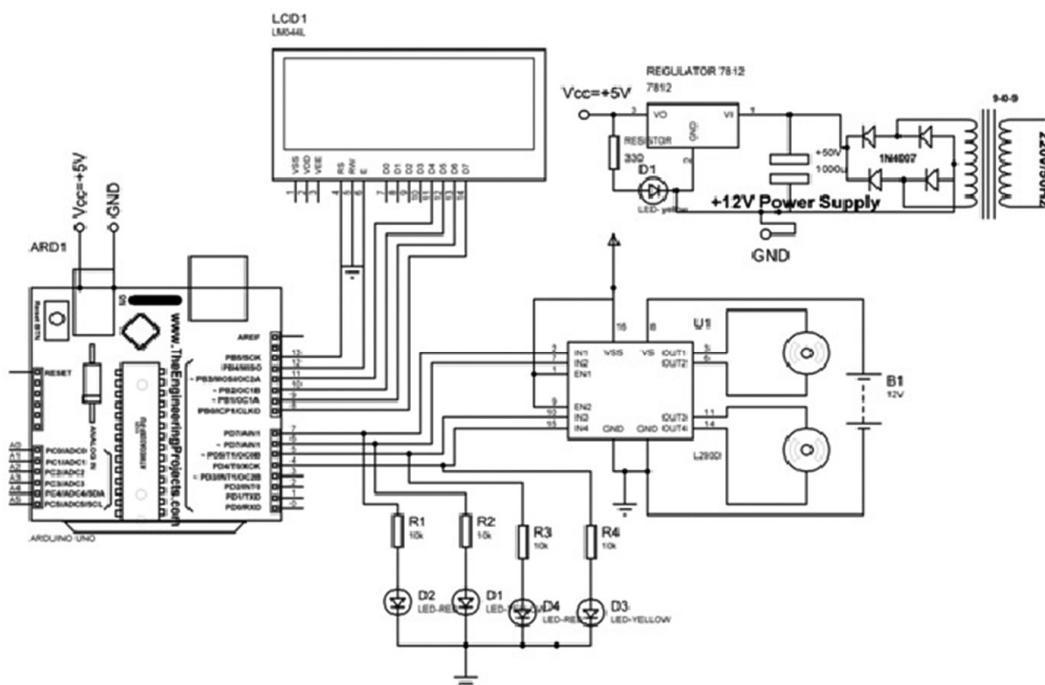
การตั้งค่าเป็นโมฆะ ()

```
{
  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 ลำช้า
  (1000); // หน่วงเวลา 1,000 ms
  lcd.clear(); // ล้างเนื้อหาของ LCD }
```

ຈະເປັນໂມឌະ ()

```
{
  LDR_sensor_ADC_Value = analogRead (LDR_sensor_Pin); // อ่าน
  ค่าจากเซ็นเซอร์
  lcd.setCursor(0,2); // ตั้งค่าเคอร์เซอร์บน LCD lcd.print("ADC
  LEVEL+LDR:"); // พิมพ์ตรึงบน 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
  }
}
```


รูป 6.6
บล็อกไอโตะแกรมของมอเตอร์กระแสตรงที่เชื่อมต่อกับ Arduino



SU 6.7

แผนภาพวงจรของมอเตอร์กระแสตรงที่เชื่อมต่อกับ 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 กับขั้วบวกของ LED1
- เชื่อมต่อ Arduino pin 6 กับขั้วบวกของ LED2
- เชื่อมต่อ Arduino pin 5 กับขั้วบวกของ LED3


```
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

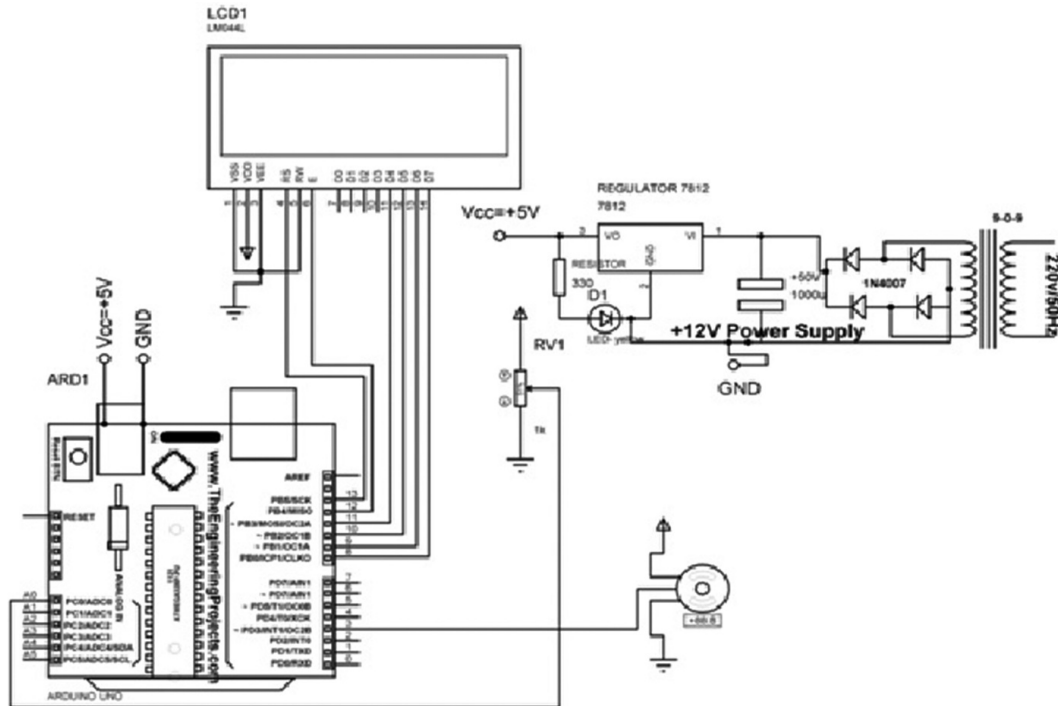
lcd.clear(); // ล้างเนื้อหาของ LCD

}
```

6.4.2 ឆេក្រូម៉ូតូរ

เซอร์โวมอเตอร์คือแอคชูเอเตอร์แบบโรตารีที่ใช้สำหรับการควบคุมตำแหน่งเชิงมุมอย่างแม่นยำ ประกอบด้วยมอเตอร์ควบคู่กับเซ็นเซอร์สำหรับ

- เชื่อมต่อ Arduino GND กับพื้น GND ของ POT
- ต่อ Arduino +5 V กับขั้ว “+” ของ POT
- ต่อขา Arduino A0 เข้ากับขา data out ของ POT



§ 6.9

แผนภาพวงจรเพื่อเชื่อมต่อเซอร์โวมอเตอร์กับ Arduino

การเชื่อมต่อ LCD

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

6.4.2.2 ຮ່າງ

```
# รวม <LiquidCrystal.h> // รวมไลบรารีของ LCD
```

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

เซอร์โว myservo; // สร้างวัตถุเซอร์โวเพื่อควบคุมเซอร์โว

```
int POT_PIN = A0; // พินแฉะล็อกที่ใช้ต่อโพเทนชิออมิเตอร์
```

ใน POT_PIN_ADC_LEVEL; // ตัวแปรที่จะอ่านค่าจาก
 เซ็นเซอร์

