Translated from English to Lao - www.onlinedoctranslator.com

6

Raspberry Pi ແລະ Arduino

6.1 ຕິດຕັ້ງ Arduino IDE ເທິງ Raspberry Pi

ຂ້ຈຳກັດກັບ Raspberry Pi ແມ່ນການບ໊ມີພອດອະນາລັອກເທິງເຮືອ, ເຊິ່ງເຮັດໃຫ້ມັນບ໊ເຫມາະສົມສຳ ລັບລະບົບທີ່ເຊັ່ນເຊີອະນາລັອກຕ້ອງການອ່ານ. ເພື່ອເອົາຊະນະຂ້ຈຳກັດນ້ີ, ສະພາບແວດລ້ອມການ ພັດທະນາປະສົມປະສານ Arduino (IDE) ສາມາດໄດ້ຮັບການຕິດຕັ້ງຢູ່ໃນ Raspberry Pi, ເນ່ືອງ ຈາກວ່າ Arduino ມີພອດອະນາລັອກດ້ັງນ້ັນພອດເຫຼ່ານ້ີສາມາດຖ້ກນ໋ໍາໃຊ້ເພ່ືອໂຕ້ຕອບເຊ້ັນເຊີ analog. ການຕິດຕ້ັງ Arduino IDE ໃນ Raspberry Pi ແມ່ນຂະບວນການທ່ີງ່າຍດາຍທ່ີມ້ີຂ້ັນ ຕອນງ່າຍໆ. Arduino IDE ແມ່ນມ້ີຢູ່ສ໋ໍາລັບລະບົບປະຕ້ິບັດການສ່ວນໃຫຍ່, ແຕ່ໃນທ່ີນ້ີພວກເຮ໋ໍາຈະ ເຫັນວ້ິທ້ີການຕ້ິດຕ້ັງມັນຢູ່ໃນ Raspberry Pi3 ຮູ້ບແບບ B ທ່ີມ້ີການແລ່ນ Raspbian Jessie ໃນ ການໂຕ້ຕອບຜູ້້ໃຊ້ແບບກາຟ້ກ (GUI).

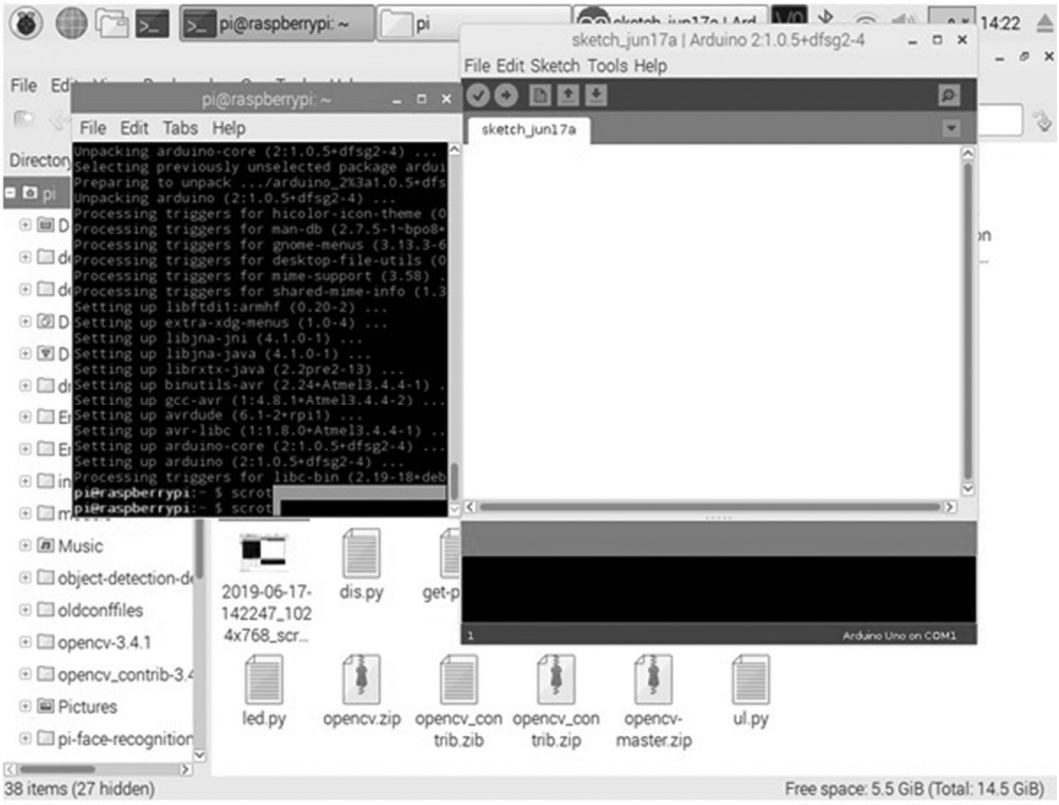
1. ຄວາມຕ້ອງການທ໋ໍາອິດແມ່ນການເຊ່ືອມຕ໊ໍອິນເຕີເນັດທ່ີໃຊ້ວຽກ.
2. ໜ້ າຈໍ, ແປ້ ນພິມ, ແລະເມ໋ໍາຕ້ ອງເຊ ື ອມຕ ໊ ໍ ກ ັ ບ Raspberry Pi.

3. ຕ ິ ດ ຕ ັ ້ ງ Arduino IDE ເ ວ ີ ຊ ັ ນ ຫ ຼ ້ າ ສ ຸ ດ ໂ ດ ຍ ໃ ຊ ້ apt:

   sudo apt-get update &&sudo apt-get upgrade
   sudo apt-get ຕ ິ ດ ຕ ັ ້ ງ arduino
4. ເ ຊ ື ອ ມ ຕ ໊ ໍ ກ ະ ດ ານ Arduino ກ ັ ບ Raspberry Pi ໂ ດ ຍ ໃ ຊ ້ ສ າ ຍ ທ ີ ເ ຫ ມ າ ະ ສ ົ ມ ແ ລ ະ ດ ຶ ງ ເ ມ ນ ູ ຫ ຼ ັ ກ Raspbian ລ ົ ງ ແ ລ ະ ເ ລ ື ອ ກ Arduino IDE ພ າ ຍ ໃ ຕ ້ ຫ ົ ວ "ເ ອ ເ ລ ັ ກ ໂ ຕ ຣ ນ ິ ກ". ປ ່ ອ ງ ຢ ້ ຽ ມ ເ ປ ົ ້ າ ຈ ະ ເ ປ ີ ດ .ຮ ູ ບ ທ ີ 6.1ສ ະ ແ ດ ງ ປ ່ ອ ງ ຢ ້ ຽ ມ ຫ ວ ່ າ ງ ເ ປ ົ ້ າ ສ ໍ າ ລ ັ ບ Arduino IDE.

5. ຄ ລ ິ ກ ທ ີ Tools > Board > ແ ລ ະ ເ ລ ື ອ ກ ກ ະ ດ ານ ທ ີ ເ ໝ າ ະ ສ ົ ມ ຂ ອ ງ Arduino.

6. ເ ພ ື ່ ອ ເ ລ ື ອ ກ ພ ອ ດ ຂ ອ ງ Arduino ທ ີ ເ ຊ ື ອ ມ ຕ ໊ ໍ , ໃ ຫ ້ ກ ວ ດ ເ ບ ິ ່ ງ ພ ອ ດ serial ພ າ ຍ ໃ ຕ ້ ເ ມ ນ ู “Tools”. ຊ ື ່ ພ ອ ດ ຂ ອ ງ Arduino ແ ມ ່ ນ : /dev/ ttyUSB0 ຫ ຼ ื /dev/ttyACM0.



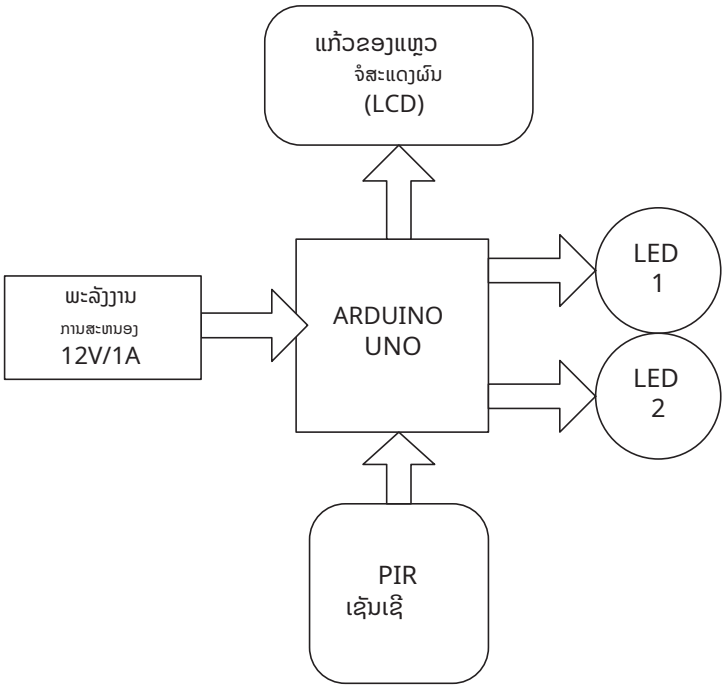
ຮູບ 6.1
ປ່ອງຢ້ຽມເປົ້າຂອງ Arduino IDE.

6.2 ຫຼີ້ນກັບເຊັນເຊີດິຈິຕອນ

ຫຼັງຈາກການຕິດຕັ້ງ Arduino IDE ໃນ Raspberry Pi, ເຊັນເຊີທີເຊື່ອມຕໍ່ກັບ Raspberry Pi ແລະ Arduino ສາມາດອ່ານໄດ້. Arduino ພຽງແຕ່ສາມາດປະຕິບັດຄືກັບກະດານ Arduino, ແລະ Raspberry Pi ເຮັດຫນ້າທີເປັນຄອມພິວເຕີເມື່ອເຊັນເຊີເຊື່ອມຕໍ່ກັບ Arduino. ນີ້ສາມາດເຂົ້າໃຈໄດ້ ໂດຍການຊ່ວຍເຫຼືອຂອງຕົວຢ່າງຈໍານວນຫນ້ອຍຫນຶ່ງ.

6.2.1 ເຊັນເຊີ PIR

ໂມດູນເຊັນເຊີ pyroelectric infrared (PIR) ຖືກໃຊ້ເພື່ອກວດຫາການເຄື່ອນໄຫວ. ມັນຫນາແຫນ້ ນແລະງ່າຍຕໍ່ການນໍາໃຊ້. ມັນມີເລນ Fresnel ແລະວົງຈອນກວດຈັບການເຄື່ອນໄຫວ, ເຊິ່ງມີລະດັບຄວາມ ກວ້າງຂອງແຮງດັນທີສະຫນອງການລະບາຍນ້ໍາຫນ້ອຍລົງ. ມັນມີຄວາມອ່ອນໄຫວສູງແລະສຽງຕ່ໍາ. ຜົນ ຜະລິດຂອງເຊັນເຊີແມ່ນສັນຍານ transistortransistor logic (TTL) ຕ່ໍາການເຄື່ອນໄຫວ. ມັນ ກວດພົບການເຄື່ອນໄຫວໂດຍການວັດແທກການປ່ຽນແປງໃນລະດັບອິນຟາເຣດທີປ່ອຍອອກມາໂດຍ ວັດຖຸໃນສິ່ງອ້ອມຂ້າງ. ໂມດູນນີ້ມີລະດັບການກວດພົບຂອງ 6 m ແລະສາມາດຖືກນໍາໃຊ້ໃນສັນຍານເຕືອນ ໄພ burglar ແລະລະບົບການຄວບຄຸມ.ຮູບທີ 6.2ສະແດງແຜນວາດຂອງລະບົບທີອອກແບບມາເພື່ອ ເຂົ້າໃຈການເຮັດວຽກຂອງເຊັນເຊີ PIR. ມັນປະກອບດ້ວຍ Arduino Uno,



ຮູບທີ 6.2
ແຜນວາດບລັອກສໍາລັບການໂຕ້ຕອບ PIR ກັບ Arduino.

ເຊັນເຊີ PIR, ຈໍສະແດງຜົນຜລຶກຂອງແຫຼວ, ແລະໄຟ LED. ລະບົບໄດ້ຖືກອອກແບບເຊັ່ນວ່າ "LED ສີ ແດງ" ຈະ "ON" ຖ້າການເຄື່ອນໄຫວຖືກກວດພົບ; ຖ້າບໍ່ດັ່ງນັ້ນ, "ໄຟ LED ສີຟ້າ" ຈະ "ເປີດ."

6.2.2 ແຜນວາດວົງຈອນ

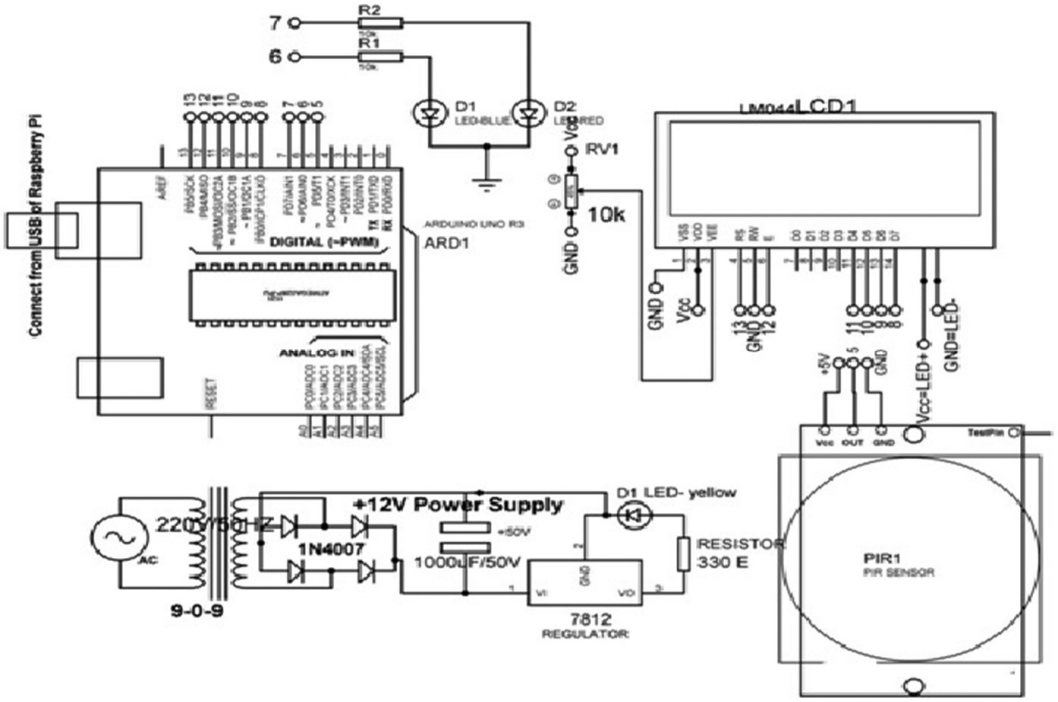
ເຊື່ອມຕໍ່ອົງປະກອບຕາມທີ່ສະແດງຢູ່ໃນຮູບທີ 6.3ເພື່ອກວດກາເບິ່ງການເຮັດວຽກຂອງເຊັນເຊີ PIR. ອັບ ໂຫລດໂຄງການທີ່ອະທິບາຍໄວ້ໃນພາກທີ 6.2.2ແລະ ກວດ ສອບ ການ ເຮັດ ວຽກ .

ການເຊື່ອມຕໍ່ເຊັນເຊີ PIR

- •ເຊື່ອມຕໍ່ Arduino GND ກັບ PIR Module GND.
- •ເຊື່ອມຕໍ່ Arduino +5 V ກັບ PIR Module +.
- • ເຊື່ອມຕໍ່ Arduino digital pin 2 ກັບ PIR Module digital out pin.

ການເຊື່ອມຕໍ່ LCD

- • ເຊື່ອມຕໍ່ Arduino digital pin (13) ກັບ RS pin(4) ຂອງ LCD.
- • ເຊື່ອມຕໍ່ Arduino digital pin (GND) ກັບ RW pin(5) ຂອງ LCD.
- • ເຊື່ອມຕໍ່ Arduino digital pin (12) ກັບ E pin(6) ຂອງ LCD.
- • ເຊື່ອມຕໍ່ Arduino digital pin (11) ກັບ D4 pin(11) ຂອງ LCD.
- • ເຊື່ອມຕໍ່ Arduino digital pin (10) ກັບ D5 pin(12) ຂອງ LCD.



ຮູບທີ 6.3
ແຜນວາດວົງຈອນສໍາລັບການໂຕ້ຕອບ PIR ກັບ Arduino.

- ເຊື່ອມຕໍ Arduino digital pin (9) ກັບ D6 pin(13) ຂອງ LCD.
- ເຊື່ອມຕໍ Arduino digital pin (8) ກັບ D7 pin(14) ຂອງ LCD.

ການເຊື່ອມຕໍ LED

- ເຊື່ອມຕໍ Arduino digital pin 7 ກັບ anode ຂອງ RED-LED ຜ່ານຕົວຕ້ານທານ 330-ohm.
- ເຊື່ອມຕໍ Arduino digital pin 6 ກັບ anode BLUE-LED ຜ່ານຕົວຕ້ານທານ 330-ohm.
- ເຊື່ອມຕໍ cathode ຂອງທັງສອງ LEDs ກັບດິນ.

6.2.3 ການແຕ້ມຮູບ

```
# ປະກອບມີ <LiquidCrystal.h> // ປະກອບມີຫ້ອງສະຫມຸດ LCD
LiquidCrystalcd(13, 12, 11, 10,9, 8); // ຕິດ LCD PIN RS,E,D4,D5,D6,D7
   ກັບ pins ທີ່ໃຫ້
int PIR_SENSOR_LOW=5; // ກໍານົດ PIN 5 ເປັນ PIR_SENOR_LOW int
RED_LED=7; // ກໍານົດ PIN 7 ເປັນ RED_LED
int BLUE_LED=6; // // ກໍານົດ PIN 6 ເປັນ BLUE_LED
void setup()
{
```

```
pinMode(PIR_SENSOR_LOW, INPUT_PULLUP); // congure pin5 as
    ວັດສະດຸປ້ອນ ແລະເປີດໃຊ້ຕົວຕ້ານການດຶງຂຶ້ນພາຍໃນ
pinMode(RED_LED,OUTPUT); // congure pin7 ເປັນ output
pinMode(BLUE_LED,OUTPUT); // congure pin6 ເປັນຜົນຜະລິດ
lcd.begin(20, 4); // ຕັ້ງ ຄ່າ LCD ຂອງ ຖັນ ແລະ ແຖວ lcd.setCursor(0, 0); // ຕັ້ງ
ຕົວກະພິບເປັນຖັນ 0 ແລະ row1 lcd.print("MOTION SENSOR BASED"); // ພິມ
ຂໍ້ຄວາມໃສ່ LCD. lcd.setCursor(0, 1); // ຕັ້ງຕົວກະພິບເປັນຖັນ 0 ແລະແຖວ 1

lcd.print("MOTION DETECTION"); // ພິມຂໍ້ຄວາມໃສ່ LCD.
lcd.setCursor(0, 2); // ຕັ້ງຕົວກະພິບເປັນຖັນ 0 ແລະແຖວ 2
lcd.print("ລະບົບຢູ LPU"); // ພິມຂໍ້ຄວາມໃສ່ LCD. ຊັກຊ້າ(1000);

}
void loop()
{
int PIR_SENSOR_LOW_READ = digitalRead(PIR_SENSOR_LOW);
    // ອ່ານຄ່າ PIR ເປັນຕົວແປ
ຖ້າ (PIR_SENSOR_LOW_READ == LOW) // ອ່ານ PIN 5 ເປັນ PIN ຕ່ໍາ
  {
lcd.clear(); // ລ້າງເນື້ອໃນຂອງ LCD lcd.setCursor(0, 3); // ຕັ້ງຕົວກະພິບເປັນ
ຖັນ 0 ແລະ row2 lcd.print("MOTION DETECTED"); // ພິມຂໍ້ຄວາມໃສ່
LCD. digitalWrite(RED_LED, HIGH); // ເຮັດໃຫ້ pin7 ເປັນ HIGH
digitalWrite(BLUE_LED, LOW); // ເຮັດໃຫ້ pin6 ເປັນ LOW delay(20); //
ຄວາມລ່າຊ້າຂອງ 20 mS

  }
ອື່ນ // ຖ້າບ ໍ ດັ ້ ງ ນ ັ ້ ນ
  {
lcd.clear(); // ລ້າງເນື້ອໃນຂອງ LCD lcd.setCursor(0, 3); // ຕັ້ງຕົວກະພິບເປັນ
column0 ແລະ row3 lcd.print("MOTION NOT DETECTED"); // ພິມຂໍ້ຄວາມໃສ່
LCD. digitalWrite(BLUE_LED, HIGH); // ເຮັດໃຫ້ pin 7 ເປັນ HIGH
digitalWrite(RED_LED,LOW); // ຕ່ໍາ pin6 ຫາ LOW

ຊັກຊ້າ(20); // ຄວາມລ່າຊ້າຂອງ 20 mS
  }
}
```

6.3 ຫຼີ້ນກັບເຊັນເຊີ Analog

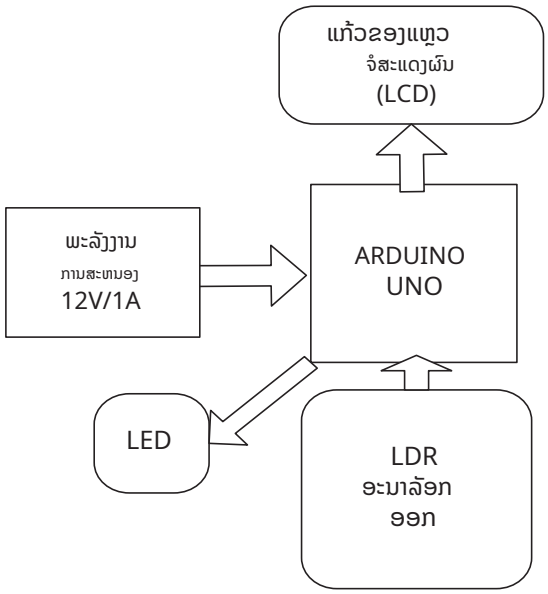
ເພື່ອອ່ານເຊັນເຊີອະນາລັອກກັບ Arduino, ພຽງແຕ່ເຊື່ອມຕໍ່ເຊັນເຊີກັບ pins ອະນາລັອກໃດໆຂອງ ກະດານ. ເພື່ອເຂົ້າໃຈການເຮັດວຽກຂອງເຊັນເຊີອະນາລັອກ, ຕົວຢ່າງຂອງຕົວຕ້ານທານແສງ (LDR) ໄດ້ ຖືກອະທິບາຍຢູ່ທີ່ນີ້. LDR ມີແຄດມຽມຊູນຟິດ (CdS) ເຊລການນໍາຮູບຖ່າຍທີ່ມີການຕອບສະໜອງ ສະເປກຕາ. ຄວາມຕ້ານທານຂອງຈຸລັງຫຼຸດລົງດ້ວຍການເພີ່ມຄວາມເຂັ້ມຂອງແສງສະຫວ່າງ. LDR ສາມາດຖືກນໍາໃຊ້ໃນຫຼາຍຄໍາຮ້ອງສະຫມັກ, ເຊັ່ນ: ການກວດຈັບຄວັນຢາສູບ, ການຄວບຄຸມແສງສະຫວ່າງ ອັດຕະໂນມັດ, ການນັບ batch, ແລະລະບົບປຸກ burglar.ຮູບທີ 6.4ສະແດງແຜນຜັງບລັອກເພື່ອ ໂຕ້ຕອບກັບ LDR ກັບ Arduino. ມັນປະກອບດ້ວຍ Arduino Uno, ການສະຫນອງພະລັງງານ, ຈໍ ສະແດງຜົນຜລຶກຂອງແຫຼວ, ແລະ LDR. ລະບົບໄດ້ຖືກອອກແບບເພື່ອສະແດງຄວາມເຂັ້ມຂອງແສງຂອງ LCD.

6.3.1 ແຜນວາດວົງຈອນ

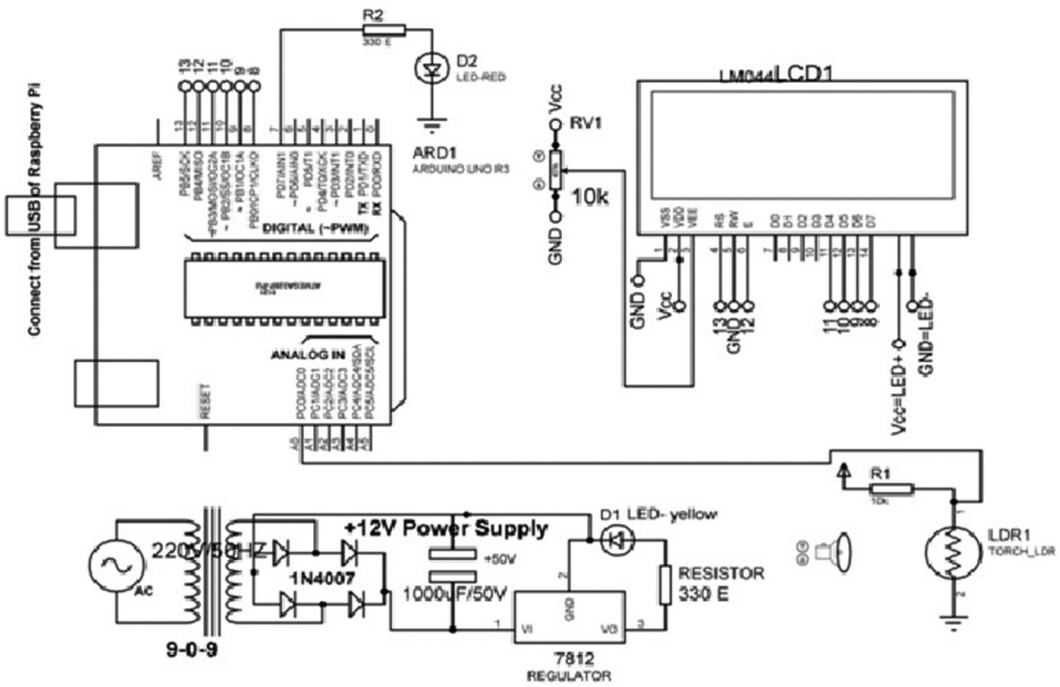
ເຊື່ອມຕໍ່ອົງປະກອບຕາມທີ່ສະແດງຢູ່ໃນຮູບທີ 6.5ເພື່ອກວດສອບການເຮັດວຽກຂອງເຊັນເຊີ LDR ເປັນ ເຊັນເຊີອະນາລັອກແບບງ່າຍດາຍ. LDR ນີ້ມີສາມຈຸດ: Ground, Vcc, OUT. ອັບໂຫລດໂຄງການທີ ອະທິບາຍໄວ້ໃນພາກທີ 6.3.2ແລະ ກວດ ສອບ ການ ເຮັດ ວຽກ .

ການເຊື່ອມຕໍ່ເຊັນເຊີແສງ

- •ເຊື່ອມຕໍ່ Arduino GND ກັບ LDR module GND.
- •ເຊື່ອມຕໍ່ Arduino +5 V ກັບ LDR Module +.
- • ເຊື່ອມຕໍ່ Arduino A0 pin ກັບ OUT pin ຂອງເຊັນເຊີ.



ຮູບ 6.4
ບລັອກແຜນວາດເພື່ອໂຕ້ຕອບກັບ LDR ກັບ Arduino.



ຮູບ 6.5
ແຜນວາດວົງຈອນສໍາລັບການຕິດຕໍເຊັນເຊີ LDR ກັບ Arduino.

ການເຊື່ອມຕໍ LCD

- ເຊື່ອມຕໍ Arduino digital pin 13 ກັບ RS pin(4) ຂອງ LCD.
- ເຊື່ອມຕໍ Arduino digital pin GND ກັບ RW pin(5) ຂອງ LCD.
- ເຊື່ອມຕໍ Arduino digital pin 12 ກັບ E pin(6) ຂອງ LCD.
- ເຊື່ອມຕໍ Arduino digital pin 11 ຫາ D4 pin(11) ຂອງ LCD.
- ເຊື່ອມຕໍ Arduino digital pin 10 ຫາ D5 pin(12) ຂອງ LCD.
- ເຊື່ອມຕໍ Arduino digital pin 9 ຫາ D6 pin(13) ຂອງ LCD.
- ເຊື່ອມຕໍ Arduino digital pin 8 ຫາ D7 pin(14) ຂອງ LCD.

6.3.2 ແຜນຜັງ

```
# ປະກອບມີ <LiquidCrystal.h> // ປະກອບມີຫ້ອງສະຫມຸດ LCD
LiquidCrystalcd(13, 12, 11, 10, 9, 8); // ຕິດ LCD PIN RS,E,D4,D5,D6,D7
   ກັບ pins ທີ່ໃຫ້
intLDR_sensor_Pin = A0; // ເລືອກ pin input ສໍາລັບ potentiometer
intLDR_sensor_ADC_Value = 0; // ຕົວແປເພື່ອເກັບຮັກສາມູນຄ່າທີ່ຈະມາເຖິງ
   ຈາກເຊັນເຊີ
int RED_LED=7; // ມອບໝາຍ PIN 7 ໃຫ້ RED_LED
```

```
ການຕັ້ງຄ່າ void()
{
lcd.begin(20, 4); // ເລີ່ມຕົ້ນ 20*4 LCD
pinMode(RED_LED,OUTPUT); // ໃຊ້ RED_LED ເປັນຜົນຜະລິດ
lcd.setCursor(0, 0); // ຕັ້ງ ຕົວ ກະ ພິບ ຂອງ LCD ຢູ ທີ column0 ແລະ
Row0 lcd.print("LDR based light"); // ພິມ string ໃນ LCD
lcd.setCursor(0, 1); // ຕັ້ງຕົວກະພິບເທິງ LCD
lcd.print("ການຕິດຕາມຄວາມເຂັ້ມງວດ"); // ພິມ string ໃນ LCD
lcd.setCursor(0, 2); // ຕັ້ງຕົວກະພິບເທິງ LCD
lcd.print("ລະບົບທີ LPU"); // ພິມສະຕຣິງໃນ LCD
delay(1000); // ຄວາມລ່າຊ້າຂອງ 1000 mS
lcd.clear(); // ລຶບ ເນື້ອ ຫາ ຂອງ LCD }



void loop()
{
LDR_sensor_ADC_Value = analogRead(LDR_sensor_Pin); // ອ່ານ
   ຄ່າຈາກເຊັນເຊີ
lcd.setCursor(0,2); // ຕັ້ງຕົວກະພິບເທິງ LCD lcd.print("ADC
LEVEL+LDR:"); // ພິມ string ໃນ LCD
lcd.setCursor(17,2); // ຕັ້ງຕົວກະພິບເທິງ LCD
lcd.print(LDR_sensor_ADC_Value); // // ຄ່າ ພິມ ໃນ LCD ຖ້າ ຫາກ ວ່າ
(LDR_sensor_ADC_Value>=100 )
  {
digitalWrite(RED_LED,HIGH); // ເຮັດໃຫ້ pin7 ເປັນຄວາມລ່າຊ້າສູງ
(20); // ຄວາມລ່າຊ້າຂອງ 20 mS
  }
ອື່ນ
  {
digitalWrite(RED_LED,LOW); // ເຮັດໃຫ້ pin7 ເປັນຄວາມລ່າຊ້າສູງ
(20); // ຄວາມລ່າຊ້າຂອງ 20 mS
  }
}
```

6.4 ຫຼີ້ນກັບ Actuators

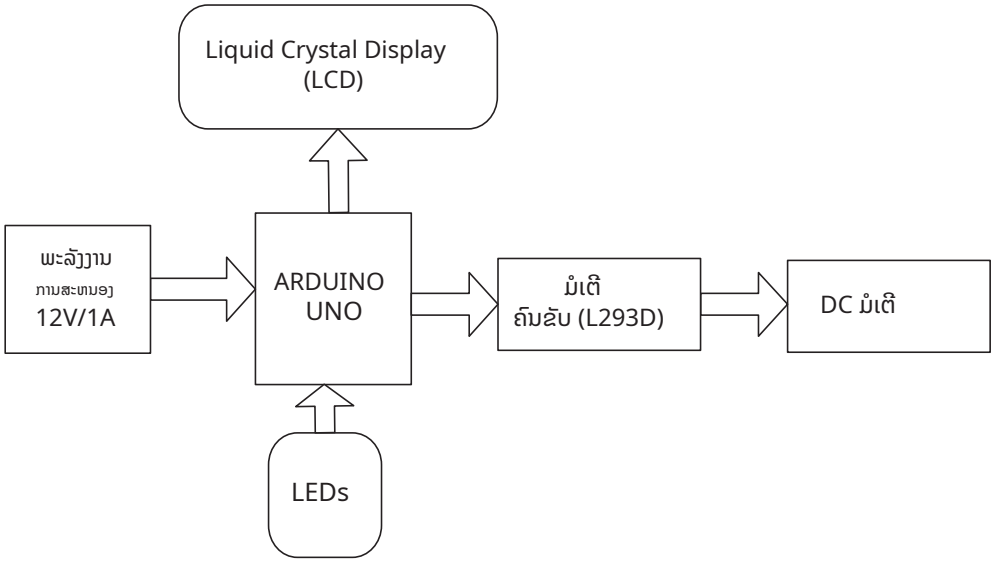
ຕົວກະຕຸ້ນແມ່ນສ່ວນປະກອບຂອງເຄື່ອງຈັກທີ່ໃຊ້ໃນການເຄື່ອນຍ້າຍແລະຄວບຄຸມກົນໄກຫຼືລະບົບ. ມໍເຕີ DC, ມໍເຕີ stepper, ແລະມໍເຕີ servo ແມ່ນຕົວກະຕຸ້ນທີ່ໃຊ້ທົ່ວໄປໃນລະບົບຕ່າງໆ.

6.4.1 DC Motor

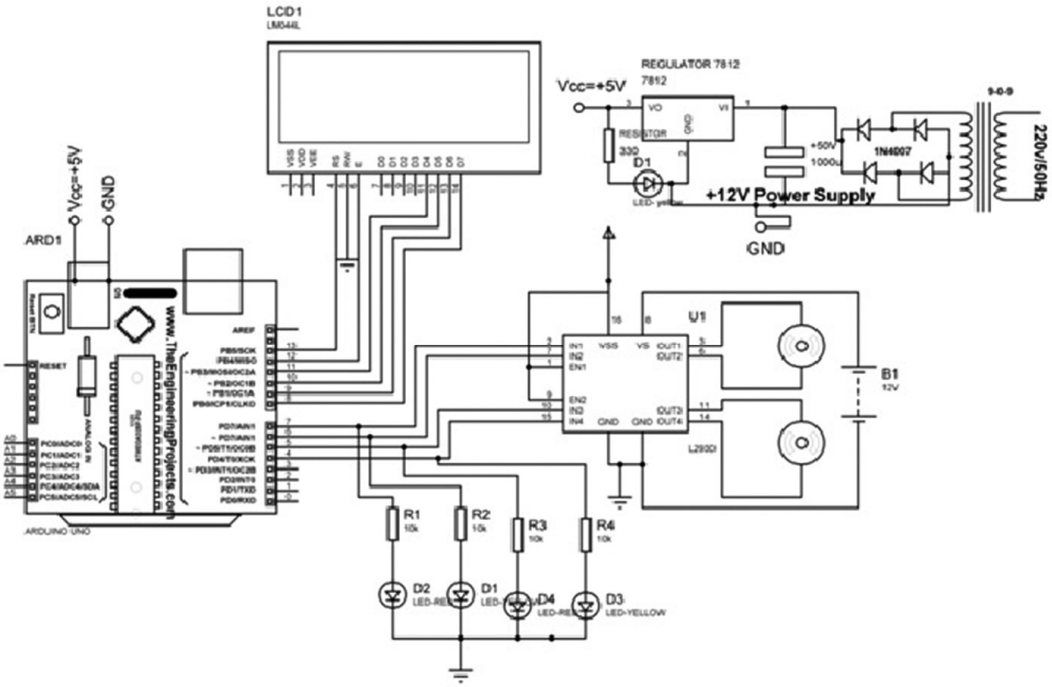
ມໍເຕີ DC-geared ທີ່ມີ 100 rpm 12 V ໂດຍທົ່ວໄປແມ່ນໃຊ້ສໍາລັບການນໍາໃຊ້ຫຸ່ນຍົນ. ພວກເຂົາເຈົ້າແມ່ນງ່າຍຫຼາຍທີ່ຈະນໍາໃຊ້. ພວກເຂົາເຈົ້າມີແກ່ນຫມາກແລະກະທູ້ຢູ່ໃນ shafts ເພື່ອເຊື່ອມຕໍ່ໄດ້ຢ່າງງ່າຍດາຍແລະ shafts threaded ພາຍໃນສໍາລັບການເຊື່ອມຕໍ່ໄດ້ງ່າຍກັບລໍ້.ຮູບທີ 6.6ສະແດງແຜນຜັງບລັອກເພື່ອໂຕ້ຕອບກັບມໍເຕີ DC ກັບ Arduino. ມັນປະກອບດ້ວຍ Arduino Uno, ການສະຫນອງພະລັງງານ, ຈໍສະແດງຜົນຜລຶກຂອງແຫຼວ, ໄດເວີມໍເຕີ (L293D), ແລະສອງມໍເຕີ DC.

6.4.1.1 ແຜນວາດວົງຈອນ

ເຊື່ອມຕໍ່ອົງປະກອບຕາມທີ່ສະແດງຢູ່ໃນຮູບທີ 6.7ເພື່ອກວດກາເບິ່ງການເຮັດວຽກຂອງມໍເຕີ DC. ອັບໂຫລດໂຄງການທີ່ອະທິບາຍໄວ້ໃນພາກທີ 6.4.1.2ແລະ ກວດ ສອບ ການ ເຮັດ ວຽກ .



ຮູບທີ 6.6
ຕັນແຜນວາດຂອງມໍເຕີ DC interfacing ກັບ Arduino.



ຮູບ 6.7
ແຜນວາດວົງຈອນຂອງ DC motor interfacing ກັບ Arduino.

ການເຊື່ອມຕໍ່ມໍເຕີ L293D ແລະ DC

- ເຊື່ອມຕໍ່ L293D pin 3 ກັບ +ve pin ຂອງ DC motor1.
- ເຊື່ອມຕໍ່ L293D pin 6 ກັບ −ve pin ຂອງ DC motor1.
- ເຊື່ອມຕໍ່ L293D pin 11 ກັບ +ve pin ຂອງ DC motor2.
- ເຊື່ອມຕໍ່ L293D pin 14 ກັບ +ve pin ຂອງ DC motor2.

ການເຊື່ອມຕໍ່ L293D

- ເຊື່ອມຕໍ່ Arduino GND ກັບ pins 4, 5, 12, 13 ຂອງ L293D.
- ເຊື່ອມຕໍ່ Arduino +5 V ກັບ pins 1, 9, 16 ຂອງ L293D.
- ເຊື່ອມຕໍ່ Arduino pin 7 ກັບ pin 2 ຂອງ L293D.
- ເຊື່ອມຕໍ່ Arduino pin 6 ກັບ pin 7 ຂອງ L293D.
- ເຊື່ອມຕໍ່ Arduino pin 5 ກັບ pin 10 ຂອງ L293D.
- ເຊື່ອມຕໍ່ Arduino pin 4 ກັບ pin 15 ຂອງ L293D.
- ເຊື່ອມຕໍ່ L293D pin 8 ກັບ +ve ຂອງຫມໍ້ໄຟ 12V

ການເຊື່ອມຕໍ່ LED

- ເຊື່ອມຕໍ່ Arduino pin 7 ກັບ anode ຂອງ LED1.
- ເຊື່ອມຕໍ່ Arduino pin 6 ກັບ anode ຂອງ LED2.
- ເຊື່ອມຕໍ່ Arduino pin 5 ກັບ anode ຂອງ LED3.

• ເຊື່ອມຕໍ Arduino pin 4 ກັບ anode ຂອງ LED4.

•ເຊື່ອມຕໍ cathode ຂອງ LEDs ທັງຫມົດກັບດິນ.

ການເຊື່ອມຕໍ LCD

• ເຊື່ອມຕໍ Arduino digital pin 13 ກັບ RS pin(4) ຂອງ LCD.

• ເຊື່ອມຕໍ Arduino digital pin GND ກັບ RW pin(5) ຂອງ LCD.

• ເຊື່ອມຕໍ Arduino digital pin 12 ກັບ E pin(6) ຂອງ LCD.

• ເຊື່ອມຕໍ Arduino digital pin 11 ຫາ D4 pin(11) ຂອງ LCD.

• ເຊື່ອມຕໍ Arduino digital pin 10 ຫາ D5 pin(12) ຂອງ LCD.

• ເຊື່ອມຕໍ Arduino digital pin 9 ຫາ D6 pin(13) ຂອງ LCD.

• ເຊື່ອມຕໍ Arduino digital pin 8 ຫາ D7 pin(14) ຂອງ LCD.

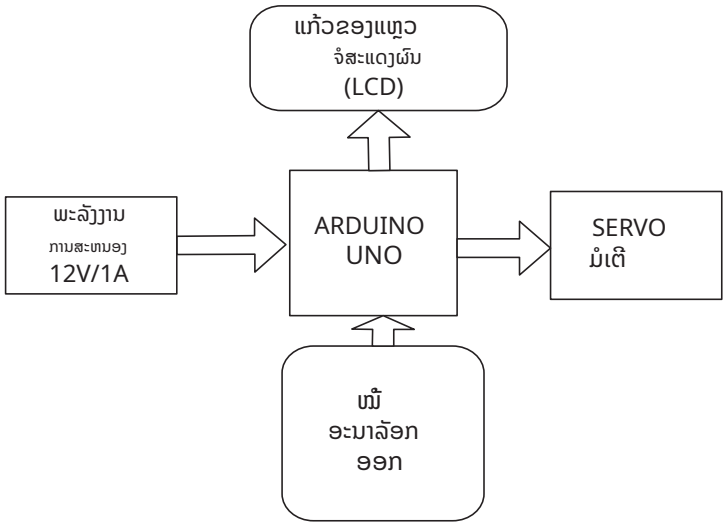
6.4.1.2 ແຜນຜັງ

```
# ປະກອບມີ <LiquidCrystal.h> // ປະກອບມີຫ້ອງສະຫມຸດ LCD
LiquidCrystalcd(13, 12, 11, 10,9, 8); // ຕິດ LCD pin RS, E, D4, D5, D6,
   D7 ກັບ pins ທີ່ໃຫ້
int MPIN1= 7; // ກໍານົດ PIN 7 ເປັນ MPIN1
int MPIN2= 6; // ກໍານົດ PIN 6 ເປັນ MPIN2
int MPIN3= 5; // ກໍານົດ PIN 5 ເປັນ MPIN3
int MPIN4= 4; // ກໍານົດ pin 4 ເປັນ MPIN4

ການຕັ້ງຄ່າ void()
{
pinMode(MPIN1, OUTPUT); // ເຮັດໃຫ້ MPIN1 ເປັນ output
pinMode(MPIN2, OUTPUT); // ເຮັດໃຫ້ MPIN2 ເປັນ output
pinMode(MPIN3, OUTPUT); // ເຮັດໃຫ້ MPIN3 ເປັນ output
pinMode(MPIN4, OUTPUT); // ເຮັດໃຫ້ MPIN4 ເປັນຜົນຜະລິດ
lcd.begin(20,4); // ເລີ່ມຕົ້ນ LCD
lcd.setCursor(0, 0); // ຕັ້ງຕົວກະພິບເທິງ LCD lcd.print("DC
Motor direction"); // ພິມ string ໃນ LCD lcd.setCursor(0,
1); // ຕັ້ງຕົວກະພິບເທິງ LCD
lcd.print("ລະບົບຄວບຄຸມ..."); // ພິມສະຕຣິງໃນ LCD
delay(1000); // ຄວາມລ່າຊ້າຂອງ 1000 mS
lcd.clear(); // ລຶບ ເນື້ອ ຫາ ຂອງ LCD }

void loop() // innite loop {

digitalWrite(MPIN1, HIGH); // ເຮັດໃຫ້ MPIN1 ເປັນ HIGH
```

```
digitalWrite(MPIN2, ต่ำ); // เธ•เธฑเนเธ MPIN2 เน€เธเนเธ low
digitalWrite(MPIN3, HIGH); // เธ•เธฑเนเธ MPIN3 เน€เธเนเธ HIGH
digitalWrite(MPIN4, LOW); // เธ•เธฑเนเธ MPIN4 เน€เธเนเธ LOW
lcd.setCursor(0, 2); // เธ•เธฑเนเธเธ•เธณเนเธซเธเนเธเธ LCD
lcd.print("เนเธกเธ"); // เธเธดเธกเธเนเธเธฃเธดเธเนเธ LCD delay(2000); //
เธเธฑเธเธเนเธฒ 2 เธงเธดเธเธฒเธ—เธต
lcd.clear(); // เธฅเธเธฅเนเธฒเธเน€เธเธทเนเธญเนเธเธเธญเธ LCD digitalWrite(MPIN1,
LOW); // เธ•เธฑเนเธเนเธซเน MPIN1 เน€เธเนเธ low digitalWrite(MPIN2,
HIGH); // เธ•เธฑเนเธเนเธซเน MPIN2 เน€เธเนเธ HIGH digitalWrite(MPIN3,
LOW); // เธ•เธฑเนเธเนเธซเน MPIN3 เน€เธเนเธ low digitalWrite(MPIN4,
HIGH); // เธ•เธฑเนเธเนเธซเน MPIN4 เน€เธเนเธ HIGH lcd.setCursor(0, 2); //
เธ•เธฑเนเธเธ•เธณเนเธซเธเนเธเธ LCD
lcd.print("เนเธกเธเธ•เนเธเธเธฒเธ"); // เธเธดเธกเธเนเธเธฃเธดเธเนเธ LCD delay(2000); //
เธเธฑเธเธเนเธฒ 2 เธงเธดเธเธฒเธ—เธต
lcd.clear(); // เธฅเธเธฅเนเธฒเธเน€เธเธทเนเธญเนเธเธเธญเธ LCD digitalWrite(MPIN1,
LOW); // เธ•เธฑเนเธเนเธซเน MPIN1 เน€เธเนเธ low digitalWrite(MPIN2,
LOW); // เธ•เธฑเนเธเนเธซเน MPIN2 เน€เธเนเธ low digitalWrite(MPIN3,
HIGH); // เธ•เธฑเนเธเนเธซเน MPIN3 เน€เธเนเธ HIGH digitalWrite(MPIN4,
LOW); // เธ•เธฑเนเธเนเธซเน MPIN4 เน€เธเนเธ LOW lcd.setCursor(0, 2); // เธ•เธฑเนเธ
เธ•เธณเนเธซเธเนเธเธ LCD
lcd.print("เธเนเธฒเธข"); // เธเธดเธกเธเนเธเธฃเธดเธเนเธ LCD
delay(2000); // เธเธฑเธเธเนเธฒ 2 เธงเธดเธเธฒเธ—เธต
lcd.clear(); // เธฅเธเธฅเนเธฒเธเน€เธเธทเนเธญเนเธเธเธญเธ LCD digitalWrite(MPIN1,
HIGH); // เธ•เธฑเนเธเนเธซเน MPIN1 เน€เธเนเธ HIGH digitalWrite(MPIN2,
LOW); // เธ•เธฑเนเธเนเธซเน MPIN2 เน€เธเนเธ low digitalWrite(MPIN3,
LOW); // เธ•เธฑเนเธเนเธซเน MPIN3 เน€เธเนเธ low digitalWrite(MPIN4,
LOW); // เธ•เธฑเนเธเนเธซเน MPIN4 เน€เธเนเธ LOW lcd.setCursor(0, 2); // เธ•เธฑเนเธ
เธ•เธณเนเธซเธเนเธเธ LCD
lcd.print("เธเธงเธฒ"); // เธเธดเธกเธเนเธเธฃเธดเธเนเธ LCD
delay(2000); // เธเธฑเธเธเนเธฒ 2 เธงเธดเธเธฒเธ—เธต
lcd.clear(); // เธฅเนเธฒเธเน€เธเธทเนเธญเนเธเธเธญเธ LCD
  }
```

6.4.2 ม็อเตเซิโว

ม็อเต servo แม่นต์วกะตุ้น rotary ที่ใช้สำลับกานถวบถุมที่ชัดเจนของตำแหน่งมุม. มัน ปะกอบด้วยม็อเตบวกกับเซ็นเซิสำลับ



ຮູບ 6.8
ບລັອກແຜນວາດໃນການໂຕ້ຕອບ servo motor ກັບ Arduino.

ຄໍາ ຄຶດ ຄໍາ ເຫັນ ຕໍາ ແຫນ່ງ . ມັນຍັງຕ້ອງການ servo drive. ໄດຣຟ໌ໃຊ້ເຊັນເຊີຕິຊົ່ມເພື່ອຄວບຄຸມ ຕໍາແໜ່ງ rotary ຂອງມໍເຕີຢ່າງແນ່ນອນ. ອັນນີ້ເອີ້ນວ່າການດໍາເນີນງານແບບວົງປິດ. ມໍເຕີ servo ມາດຕະຖານແຮງບິດສູງທີ່ມີເກຍໂລຫະແລະ 360°ການຫມຸນສາມາດສະຫນອງ 11 kg / cm ຢູ່ທີ 4.8 V, 13.5 kg / cm ທີ 6 V, ແລະ 16 kg / cm ທີ 7.2 V.ຮູບ 6.8ສະແດງແຜນຜັງບລັອກເພື່ອ ໂຕ້ຕອບກັບ servo motor ກັບ Arduino. ມັນປະກອບດ້ວຍ Arduino Uno, ການສະຫນອງ ພະລັງງານ, ຈໍສະແດງຜົນຜລຶກຂອງແຫຼວ, potentiometer (POT), ແລະ servo motor. ລະບົບ ໄດ້ຖືກອອກແບບເພື່ອຄວບຄຸມມຸມຂອງມໍເຕີ servo ກັບ potentiometer.

6.4.2.1 ແຜນວາດວົງຈອນ

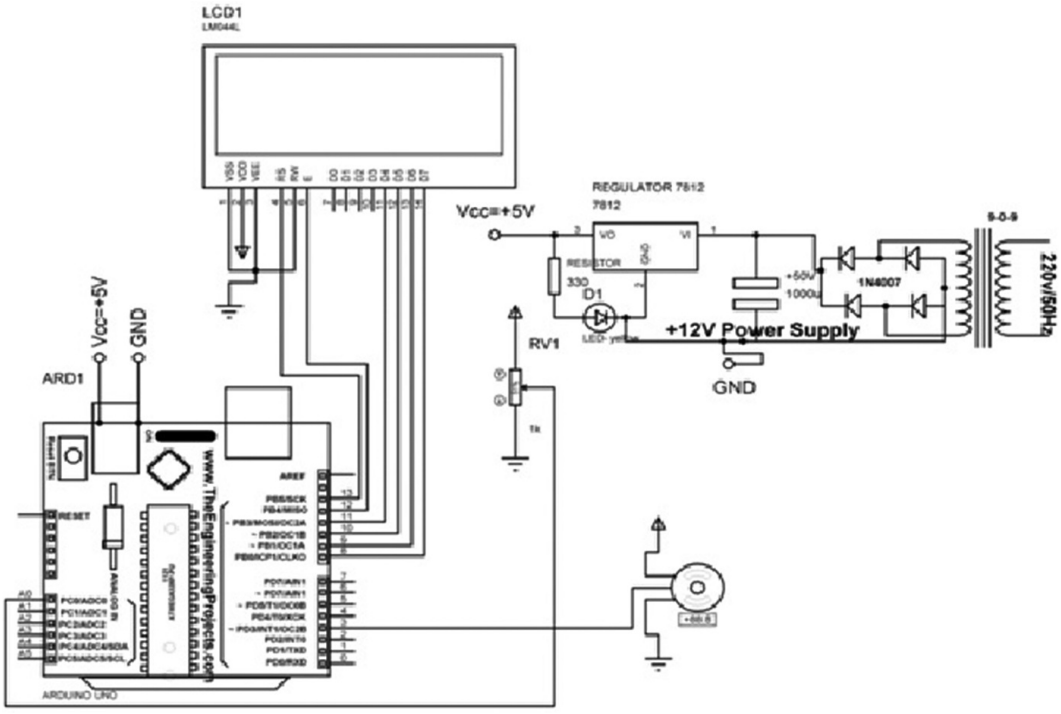
ເຊື່ອມຕໍ່ອົງປະກອບຕາມທີ່ສະແດງຢູ່ໃນຮູບທີ 6.9ເພື່ອກວດກາເບິ່ງການເຮັດວຽກຂອງມໍເຕີ servo. ອັບ ໂຫລດໂຄງການທີ່ອະທິບາຍໄວ້ໃນພາກທີ 6.4.2.2ແລະ ກວດ ສອບ ການ ເຮັດ ວຽກ .

ການເຊື່ອມຕໍ່ Servo

- •ເຊື່ອມຕໍ່ Arduino GND ກັບ GND pin ຂອງ servo motor.
- • ເຊື່ອມຕໍ່ Arduino +5 V ກັບ “+” terminal ຂອງ servo motor.
- • ເຊື່ອມຕໍ່ Arduino pin(3) ກັບ PWM pin ຂອງ servo motor.

ການເຊື່ອມຕໍ່ POT

- • ເຊື່ອມຕໍ່ Arduino GND ກັບ GND pin ຂອງ POT.
- • ເຊື່ອມຕໍ່ Arduino +5 V ກັບ “+” terminal ຂອງ POT.
- • ເຊື່ອມຕໍ່ Arduino A0 pin ກັບຂໍ້ມູນອອກ PIN ຂອງ POT.



ຮູບ 6.9
ແຜນວາດວົງຈອນໃນການໂຕ້ຕອບ servo motor ກັບ Arduino.

ການເຊື່ອມຕໍ LCD

- ເຊື່ອມຕໍ Arduino digital pin (13) ກັບ RS pin(4) ຂອງ LCD.
- ເຊື່ອມຕໍ Arduino digital pin (GND) ກັບ RW pin(5) ຂອງ LCD.
- ເຊື່ອມຕໍ Arduino digital pin (12) ກັບ E pin(6) ຂອງ LCD.
- ເຊື່ອມຕໍ Arduino digital pin (11) ກັບ D4 pin(11) ຂອງ LCD.
- ເຊື່ອມຕໍ Arduino digital pin (10) ກັບ D5 pin(12) ຂອງ LCD.
- ເຊື່ອມຕໍ Arduino digital pin (9) ກັບ D6 pin(13) ຂອງ LCD.
- ເຊື່ອມຕໍ Arduino digital pin (8) ກັບ D7 pin(14) ຂອງ LCD.

6.4.2.2 ແຜນຜັງ

```
# ປะกอบมิ <LiquidCrystal.h> // ปะกอบมิห้องสะหมุด LCD
LiquidCrystalcd(13, 12, 11, 10, 9, 8); // ติด LCD PIN RS,E,D4,D5,D6,D7
    กับ pins ที่ให้
เຊີโว myservo; // สร้างวัดถุ servo เพื่อถวบถุม servo
int POT_PIN = A0; // pin analog ใช้เพื่อเชื่อมตๆ potentiometer
int POT_PIN_ADC_LEVEL; // ต๋วแปเพื่ออ่านถ่าจาก
    เซ็มอะมาล้อก
```

```
ภามตั้งถ่า void()
{
myservo.attach(3); // ติด servo ใม pin 9 ภับ servo object
lcd.begin(20,4); // เลิมต้น LCD
lcd.setCursor(0, 0); // ตั้งต์วภะพิบใส่ LCD.print("Servo
ANALOG write"); // พิมสะตริงเทิง LCD lcd.setCursor(0, 1);//
ตั้งต์วภะพิบเทิง LCD
lcd.print("ละบ์บทิ LPU....");// พิมสายยู LCD }

void loop()
{
POT_PIN_ADC_LEVEL = analogRead(POT_PIN); // อ่านถ่า POT
   ใม ธูบ แบบ ของ ละ ดับ
POT_PIN_ADC_LEVEL = แผมทิ(POT_PIN_ADC_LEVEL, 0, 1023, 0, 179);
   // แผมทิถ่า // ละหว่าง 0 หา 180 อ์งสาสำลับ servo
myservo.write(POT_PIN_ADC_LEVEL); // ภำม์ดตำแหน่ง servo
   อิง ตาม มูน ถ่า ขะ หมาด
lcd.setCursor(0, 2); // ตั้งต์วภะพิบเทิง LCD
lcd.print("ANGLE:"); // พิม string เทิง LCD
lcd.print(POT_PIN_ADC_LEVEL); // ถ่า พิม ใม LCD
delay(15); // ภามชักช้าของ 15 mSec
}
```