**<https://github.com/abdowaili/solar-Tracking-.git>**

**Coding Arduino UNO for Simulation**

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x27, 16, 2);

int LDR;

int LDRPin=A3;

int LDR2;

int LDRPin2=A2;

int LDR3;

int LDRPin3=A1;

#include <Servo.h>

Servo myservo1;

Servo myservo2;

void setup()

{

lcd.init();

lcd.backlight();

myservo1.attach(A0);

myservo2.attach(10);

Serial.begin(9600);

}

void loop(){

lcd.setCursor(0, 0);

lcd.print("sensor=9V");

LDR=analogRead(LDRPin);

Serial.println(LDR);

delay(100);

LDR2=analogRead(LDRPin2);

Serial.println(LDR2);

delay(100);

LDR3=analogRead(LDRPin3);

Serial.println(LDR3);

delay(100);

if(LDR>500){

myservo1.write(180);

myservo2.write(0);

}

if(LDR2>500){

myservo1.write(150);

myservo2.write(60);

}

if(LDR3>500){

myservo1.write(0);

myservo2.write(0);

}

}

**Coding Arduino UNO for Hardware**

#include <SoftwareSerial.h>

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

#include <Servo.h>

SoftwareSerial esp(6, 7);

#define DEBUG true

String mySSID = "Omar Al Waaili";

String myPWD = "Omar1823";

String myAPI = "XWBKJTA6VSKP5X7D";

String myHOST = "api.thingspeak.com";

String myPORT = "80";

String myFIELD = "field1";

LiquidCrystal\_I2C lcd(0x27, 16, 2);

Servo myservo1;

Servo myservo2;

#define ANALOG\_IN\_PIN A0

float R1 = 30000.0;

float R2 = 7500.0;

float ref\_voltage = 5.0;

float in\_voltage = 0.0;

int batteryPin = A4; // Analog pin to monitor battery voltage

int wireLossPin = A5; // Analog pin to monitor wire connection

void setup() {

Serial.begin(115200);

esp.begin(115200);

lcd.init();

lcd.backlight();

myservo1.attach(11);

myservo2.attach(10);

pinMode(2, OUTPUT);

pinMode(3, OUTPUT);

Send\_AT\_Cmd("AT+RST", 1000, DEBUG);

Send\_AT\_Cmd("AT+CWMODE=1", 1000, DEBUG);

Send\_AT\_Cmd("AT+CWJAP=\"" + mySSID + "\",\"" + myPWD + "\"", 1000, DEBUG);

delay(1000);

}

void loop() {

// Read solar panel voltage

float adc\_voltage = analogRead(ANALOG\_IN\_PIN) \* ref\_voltage / 1024.0;

in\_voltage = adc\_voltage \* (R1 + R2) / R2;

// Display voltage on LCD

lcd.setCursor(0, 0);

lcd.print("Voltage: ");

lcd.print(in\_voltage, 2);

lcd.print("V ");

// Check voltage and update LED

if (in\_voltage == 0) {

lcd.setCursor(0, 1);

lcd.print("No Voltage ");

digitalWrite(2, LOW); // Green LED

digitalWrite(3, HIGH); // Red LED indicating problem

} else if (in\_voltage < 2) {

lcd.setCursor(0, 1);

lcd.print("Low Voltage ");

digitalWrite(2, LOW); // Green LED

digitalWrite(3, HIGH); // Red LED indicating problem

} else {

lcd.setCursor(0, 1);

lcd.print("Normal ");

digitalWrite(2, HIGH); // Green LED

digitalWrite(3, LOW); // Red LED off indicating normal operation

}

// Solar tracking based on LDR readings

int LDR1 = analogRead(A3);

int LDR2 = analogRead(A2);

int LDR3 = analogRead(A1);

if (LDR1 > LDR2 && LDR1 > LDR3) {

myservo1.write(180);

myservo2.write(0);

} else if (LDR2 > LDR1 && LDR2 > LDR3) {

myservo1.write(150);

myservo2.write(60);

} else {

myservo1.write(0);

myservo2.write(0);

}

// Send data to Thingspeak

String sendData = "GET /update?api\_key=" + myAPI + "&" + myFIELD + "=" + String(in\_voltage, 2);

Send\_AT\_Cmd("AT+CIPMUX=1", 1000, DEBUG);

Send\_AT\_Cmd("AT+CIPSTART=0,\"TCP\",\"" + myHOST + "\"," + myPORT, 1000, DEBUG);

Send\_AT\_Cmd("AT+CIPSEND=0," + String(sendData.length() + 4), 1000, DEBUG);

esp.find(">");

esp.println(sendData);

Send\_AT\_Cmd("AT+CIPCLOSE=0", 1000, DEBUG);

delay(100); // Adjust delay as necessary, or replace with millis() based timing

}

String Send\_AT\_Cmd(String command, const int timeout, boolean debug) {

Serial.print(command);

Serial.println(" ");

String response = "";

esp.println(command);

long int time = millis();

while ((time + timeout) > millis()) {

while (esp.available()) {

char c = esp.read();

response += c;

}

}

if (debug) {

//Serial.print(response);

}

return response;

}