bool motorState = false;

#include <SPI.h>

#include <Wire.h>

#include <Adafruit\_GFX.h>

#include <Adafruit\_SSD1306.h>

#define SCREEN\_WIDTH 128 // OLED display width, in pixels

#define SCREEN\_HEIGHT 32 // OLED display height, in pixels

#define cellPin A0

const int analogPin = A1; // Pin connected to S

const float referenceVoltage = 5.0; // Arduino's operating voltage

const float voltageDividerRatio = 0.1; // Adjust according to your module's spec (example: 1:10 divider)

#define OLED\_RESET -1 // Reset pin # (or -1 if sharing Arduino reset pin)

#define SCREEN\_ADDRESS 0x3C ///< See datasheet for Address; 0x3D for 128x64, 0x3C for 128x32

Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, OLED\_RESET);

#define LOGO\_HEIGHT 16 // Removed semicolon

#define LOGO\_WIDTH 16 // Removed semicolon

#include <IRremote.h>

#define STEPPER\_PIN\_1 10

#define STEPPER\_PIN\_2 11

#define STEPPER\_PIN\_3 12

#define STEPPER\_PIN\_4 13

#define BLUE\_LED 5

#define RED\_LED 6

int step\_number = 0;

int receiver = 9;

IRrecv IR(receiver);

void setupDisplay() {

display.begin(SSD1306\_SWITCHCAPVCC, SCREEN\_ADDRESS); // Removed extra closing parenthesis

display.clearDisplay();

display.setTextSize(3,5);

display.setTextColor(WHITE);

display.setCursor(0, 0);

}

void setup() {

pinMode(BLUE\_LED, OUTPUT);

pinMode(STEPPER\_PIN\_1, OUTPUT);

pinMode(STEPPER\_PIN\_2, OUTPUT);

pinMode(STEPPER\_PIN\_3, OUTPUT);

pinMode(STEPPER\_PIN\_4, OUTPUT);

Serial.begin(9600);

IR.enableIRIn();

setupDisplay();

digitalWrite(RED\_LED, HIGH);

}

void loop() {

if(IR.decode()){

if(IR.decodedIRData.decodedRawData == 0XEA15FF00 && motorState == false) { //+ == open

for(int i = 0; i < 2048; i++){

OneStep(false);

delay(2);

}

motorState = true;

digitalWrite(BLUE\_LED, HIGH);

digitalWrite(RED\_LED, LOW);

}

}

if(IR.decode()){

if(IR.decodedIRData.decodedRawData == 0XF807FF00 && motorState == true) { //- == close

for(int i = 0; i < 2048; i++){

OneStep(true);

delay(2);

}

motorState = false;

digitalWrite(BLUE\_LED, LOW);

digitalWrite(RED\_LED, HIGH);

}

}

IR.resume();

delay(1000);

int sensorValue = analogRead(analogPin); // Read the analog pin

float measuredVoltage = sensorValue \* (referenceVoltage / 1023.0); // Convert to voltage

float actualVoltage = (measuredVoltage / voltageDividerRatio - 13.0) / 10; // Scale back to the input voltage

Serial.print("Measured Voltage: ");

Serial.print(actualVoltage);

Serial.println(" V");

display.println(actualVoltage);

display.display();

delay(1500);

setupDisplay();

}

void OneStep(bool dir) {

if(dir) {

switch(step\_number) {

case 0:

digitalWrite(STEPPER\_PIN\_1, HIGH);

digitalWrite(STEPPER\_PIN\_2, LOW);

digitalWrite(STEPPER\_PIN\_3, LOW);

digitalWrite(STEPPER\_PIN\_4, LOW);

break;

case 1:

digitalWrite(STEPPER\_PIN\_1, LOW);

digitalWrite(STEPPER\_PIN\_2, HIGH);

digitalWrite(STEPPER\_PIN\_3, LOW);

digitalWrite(STEPPER\_PIN\_4, LOW);

break;

case 2:

digitalWrite(STEPPER\_PIN\_1, LOW);

digitalWrite(STEPPER\_PIN\_2, LOW);

digitalWrite(STEPPER\_PIN\_3, HIGH);

digitalWrite(STEPPER\_PIN\_4, LOW);

break;

case 3:

digitalWrite(STEPPER\_PIN\_1, LOW);

digitalWrite(STEPPER\_PIN\_2, LOW);

digitalWrite(STEPPER\_PIN\_3, LOW);

digitalWrite(STEPPER\_PIN\_4, HIGH);

break;

}

} else {

switch(step\_number) {

case 0:

digitalWrite(STEPPER\_PIN\_1, LOW);

digitalWrite(STEPPER\_PIN\_2, LOW);

digitalWrite(STEPPER\_PIN\_3, LOW);

digitalWrite(STEPPER\_PIN\_4, HIGH);

break;

case 1:

digitalWrite(STEPPER\_PIN\_1, LOW);

digitalWrite(STEPPER\_PIN\_2, LOW);

digitalWrite(STEPPER\_PIN\_3, HIGH);

digitalWrite(STEPPER\_PIN\_4, LOW);

break;

case 2:

digitalWrite(STEPPER\_PIN\_1, LOW);

digitalWrite(STEPPER\_PIN\_2, HIGH);

digitalWrite(STEPPER\_PIN\_3, LOW);

digitalWrite(STEPPER\_PIN\_4, LOW);

break;

case 3:

digitalWrite(STEPPER\_PIN\_1, HIGH);

digitalWrite(STEPPER\_PIN\_2, LOW);

digitalWrite(STEPPER\_PIN\_3, LOW);

digitalWrite(STEPPER\_PIN\_4, LOW);

break;

}

}

step\_number++;

if(step\_number > 3) {

step\_number = 0;

}

}