Solar Panel Tracker - Radulescu Stefan

Introducere

Solar Panel Tracker este o solutie care permite orientarea panoului fotovoltaic în funcție de poziția soarelui. Acest sistem de urmărire a soarelui maximizează producția de energie electrică a panoului prin expunerea constantă a suprafeței sale la lumina directă a soarelui.

Descriere generală

Sistemul se foloseste de 2 servo motoare pentru a modifica pozitia panoului: unul pentru axa orizontala, iar celalalt pentru verticala.
Acesta are la dispozitie doua moduri de functionare:

1. Automat:

- Sistemul cauta in permanenta cea mai buna pozitie pentru maximizarea obtinerii de energie solara.
- Se incearca gasirea unghiului la care lumina solara cade perpendicular pe panoul fotovoltaic.

2. Manual

- Pozitia panoului poate fi modificata prin intermediul unei telecomenzi care transmite semnale IR, semnale care se adreseaza motoarelor. Pozitia lor este modificata in functie de acestea.

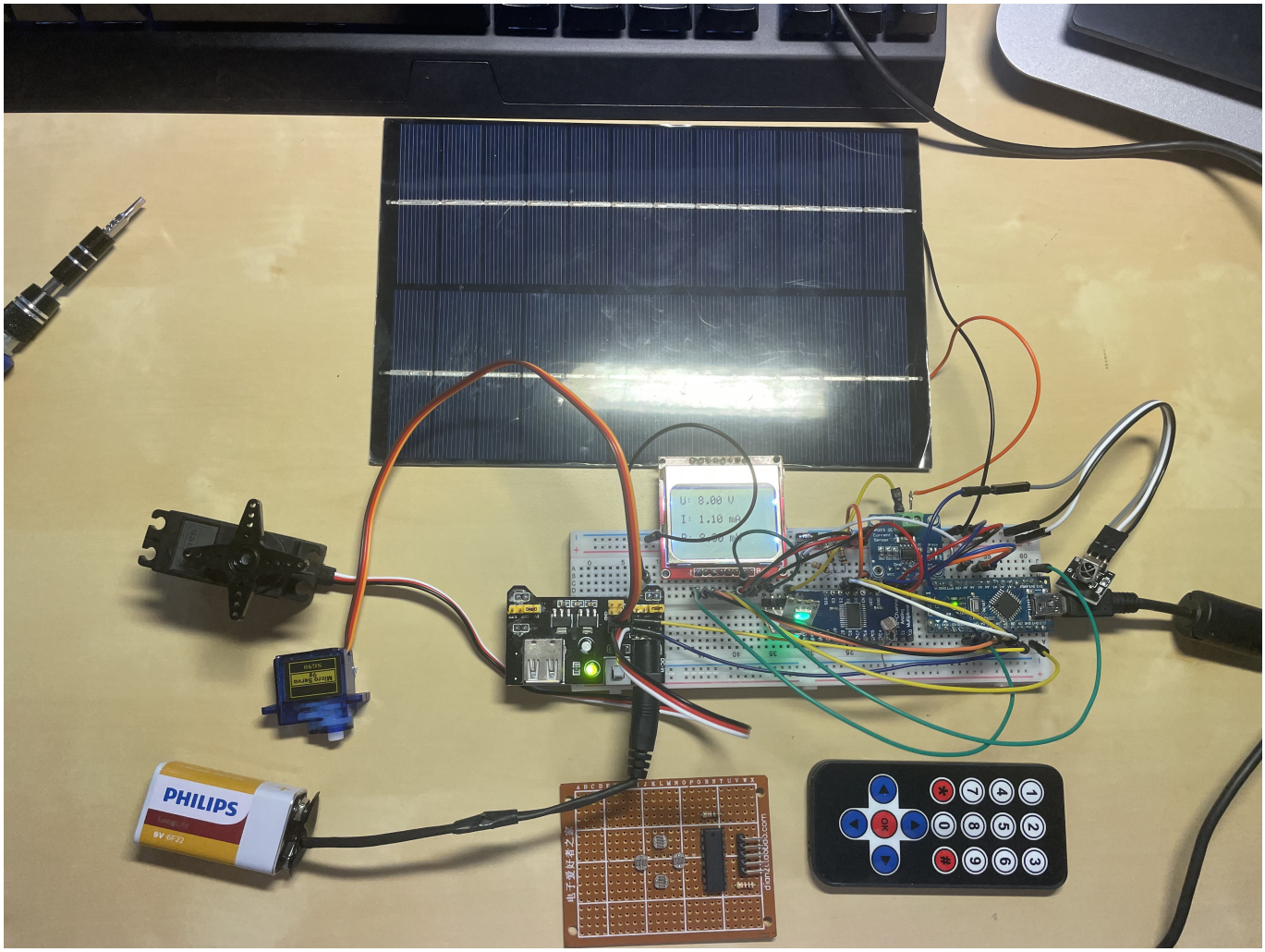
Hardware Design

Lista piese:

1. Arduino Nano
2. Senzor lumina format din 4 fotorezistori cu multiplexor analogic
3. 2 motoare servo S3003
4. Senzor curent si tensiune INA219
5. Display Nokia 5110
6. Senzor IR pentru control prin telecomanda
7. LED-uri aditionale

8. Panou solar 12V 0.4A(max)
9. Sursa de curent separata pentru servomotoare

Senzorul de lumina este format din 4 fotorezistori asezati in forma de plus, delimitati de un mini perete, pentru a se putea forma umbre. Astfel, in functie de tensiunea de pe fiecare fotorezistor, ne putem da seama de pozitia soarelui.



Software Design

1. Senzor curent si tensiune INA219 - I2C
2. Display 5110 - SPI (initial hardware SPI, dar cand am lipit totul pe PCB am folosit bliblioteca Adafruit, deoarece pinii erau prea distantati si imi era greu sa ii conectez)
3. Am folosit intreruperi externe pentru senzorul de lumina inflarosie.
4. PWM pentru servomotoare.
5. ADC pentru a citi fotorezistorii

Codul sursa:

```
#include <SPI.h>
#include <Adafruit_GFX.h>
```

```
#include <Adafruit_PCD8544.h>
#include <Wire.h>
#include <Adafruit_INA219.h>
#include <Arduino.h>
#include <IRremote.h>
#include <Servo.h>

#define S0_PIN 8
#define S1_PIN 9
#define PHRES_PIN A7
#define IR_PIN 2
#define SCLK 3
#define DIN 4
#define DC 5
#define CS 6
#define RST 7
#define SERVO1 11
#define SERVO2 10


unsigned int pos = 85;
int movement = 0;
Servo servoX;
Servo servoY;
unsigned long time, lastTime;
bool ok = 0;
bool MODE_AUTO = 1;


float voltage;
float current;
int left_value = 0;
int right_value = 0;
int up_value = 0;
int down_value = 0;
Adafruit_INA219 ina219;




Adafruit_PCD8544 display = Adafruit_PCD8544(SCLK, DIN, DC, CS, RST);

void setup() {
  Serial.begin(9600);
  while (!Serial)
    delay(1);
  if (! ina219.begin()) {
    Serial.println("Failed to find INA219 chip");
    while (1) { delay(10); }
  }
  display.begin();
```

```
  display.setContrast(25);
  pinMode(S0_PIN, OUTPUT);
  pinMode(S1_PIN, OUTPUT);
  pinMode(LED_BUILTIN, OUTPUT);

  IrReceiver.begin(IR_PIN, DISABLE_LED_FEEDBACK);
  servoX.attach(SERVO1);
  servoY.attach(SERVO2);
  servoY.write(pos);
  attachInterrupt(0, ISR_IR, RISING);
}

void loop() {
  voltage = ina219.getBusVoltage_V();
  current = ina219.getCurrent_mA();
  printPowerInfo(voltage, current);
  if (MODE_AUTO)
    automaticMode();
  else
    manualMode();


  delay(1000);
}

void automaticMode() {
  digitalWrite(LED_BUILTIN, HIGH);
  Serial.println("AUTO MODE");
  readPhotoresistors();
  printPhotoresistors();
  if (up_value - down_value > 150) {
    Serial.println("MOVE DOWN");
    moveDown();
  }
  else if (down_value - up_value > 150) {
    Serial.println("MOVE UP");
    moveUp();
  }
  else if (left_value - right_value > 150) {
    Serial.println("MOVE LEFT");
    moveLeft();
    delay(250);
    servoStop();
  }
  else if (right_value - left_value > 150) {
    Serial.println("MOVE RIGHT");
    moveRight();
    delay(250);
    servoStop();
  }
  else Serial.println("OK!");
```

```
}

void ISR_IR() {
  MODE_AUTO = 0;
}

void manualMode() {
  digitalWrite(LED_BUILTIN, LOW);
  detachInterrupt(0);
  Serial.println("MANUAL MODE");
  while (1) {
    voltage = ina219.getBusVoltage_V();
    current = ina219.getCurrent_mA();
    printPowerInfo(voltage, current);
    if (ok && micros() - lastTime > 150000) {
      ok = 0;
      servoStop();
    }
    if (MODE_AUTO)
      break;
    if (IrReceiver.decode()) {
      IrReceiver.resume();
      time = micros();
      if (time - lastTime > 150000) {
        if (IrReceiver.decodedIRData.command == 0x8) {
            Serial.println("LEFT");
            moveLeft();
        } else if (IrReceiver.decodedIRData.command == 0x5A) {
            Serial.println("RIGHT");
            moveRight();
        } else if (IrReceiver.decodedIRData.command == 0x18) {
            Serial.println("UP");
            moveUp();
        } else if (IrReceiver.decodedIRData.command == 0x52) {
            Serial.println("DOWN");
            moveDown();
        } else if (IrReceiver.decodedIRData.command == 0x45) {
            Serial.println("AUTO");
            MODE_AUTO = 1;
        }
        ok = 1;
      }
      lastTime = time;
    }
  }
}

void moveRight() {
  servoX.write(100);
}
```

```
void moveLeft() {
  servoX.write(80);
}

void servoStop() {
  Serial.println("STOP");
  servoX.write(95);
  movement = 0;
}

void updatePos() {
  pos += (movement * 5);
  servoY.write(pos);
}

void moveUp() {
  if (pos < 115) {
    pos += 5;
    servoY.write(pos);
  }
  Serial.println(pos);
}

void moveDown() {
  if (pos > 20) {
    pos -= 5;
    servoY.write(pos);
  }
  Serial.println(pos);
}

void printPowerInfo(float voltage, float current) {
  display.clearDisplay();
  display.print("U: ");
  display.print(voltage);
  display.println(" V");
  display.println();
  display.print("I: ");
  display.print(current);
  display.println(" mA");
  display.println();
  display.print("P: ");
  display.print(voltage * current);
  display.println(" mW");
  display.display();
}

void printPhotoresistors() {
  Serial.print("\t");
  Serial.println(up_value);
```

```
  Serial.print(left_value);
  Serial.print("\t\t");
  Serial.println(right_value);
  Serial.print("\t");
  Serial.println(down_value);
  Serial.println();
}

void readPhotoresistors() {
  left_value = readLeft();
  right_value = readRight();
  up_value = readUp();
  down_value = readDown();
}

int readLeft() {
  digitalWrite(S0_PIN, LOW);
  digitalWrite(S1_PIN, LOW);
  delay(10);
  return analogRead(PHRES_PIN);
}

int readDown() {
  digitalWrite(S0_PIN, LOW);
  digitalWrite(S1_PIN, HIGH);
  delay(10);
  return analogRead(PHRES_PIN);
}

int readUp() {
  digitalWrite(S0_PIN, HIGH);
  digitalWrite(S1_PIN, LOW);
  delay(10);
  return analogRead(PHRES_PIN);
}

int readRight() {
  digitalWrite(S0_PIN, HIGH);
  digitalWrite(S1_PIN, HIGH);
  delay(10);
  return analogRead(PHRES_PIN);
}
```

Rezultate Obţinute

Pot aprinde un bec de 1w cu panoul fotovoltaic :D
Solar Tracker

Concluzii

Pentru a putea trage concluzii despre un sistem de acest fel, este nevoie de testare pe perioada indelungata de timp (6 luni - 1 an).

Download

solar_tracker.zip

Jurnal

- 14-15.05.2023 - Testarea pieselor comandate
- 16.05.2023 - Design-ul senzorului de directie a luminii
- 17-20.05 - Depanarea si rezolvarea problemelor, impreuna cu comandarea pieselor aditionale
- 26.05.2023 - Printarea componentelor 3D
- 27-28.05.2023 - Asamblarea tuturor componentelor
- 29.05.2023 - Finalizarea software-ului

Bibliografie/Resurse

- Arduino Examples
- Adafruit Library Sensor Examples (INA219, PCD8544)
- Laboratoarele de intreruperi, PWM, ADC, SPI, I2C.

Export to PDF