Sistem de irigatie smart

Introducere

Aplicatia dezvoltata se numeste Sistem de irigatie smart. Aceasta are rolul de a actiona o pompa (reprezentata de un micro servo motor) care elibereaza apa cu un anumit debit in functie de informatiile inregistrate de senzorii de umiditate a aerului, a solului, lumina si temperatura. Importanta partilor hardware si software este egala: partea hardware se ocupa cu masurarea celor 4 parametrii si de eliberarea apei in sol, partea software are rolul de a asigura logica din spatele starii pompei, de conversii ale datelor inregistrate si de afisarea datelor intr-un format usor de citit.

Descriere generală

Hardware Design

Lista componente:

- Arduino UNO
- Breadboard Small
- Micro Servo Motor
- Water level Sensor
- DHT11 Sensor
- Photoresistor
- LCD 16 x 2
- Rezistente
- Circuite de diferite culori (pt GND, 5V etc.)
- Potentiometru (pentru afisajul LCD-ului)

Schema electrica a fost realizata cu ajutorul Friltzing

Software Design

Biblioteci folosite:

- DHT.h (folosita pentru senzorul de temperatura si umiditate DHT11)
- Servo.h (folosita pentru micro servo motor)
- LiquidCrystal.h (folosita pentru LCD display-ul 16×2)

Functii folosite din biblioteca DHT.h:

```
  DHT(uint8_t pin, uint8_t type, uint8_t count = 6);
  void begin(uint8_t usec = 55);
  float readHumidity(bool force = false);
  float readTemperature(bool S = false, bool force = false);
```

Functii folosite din biblioteca Servo.h:

```
  uint8_t attach(int pin);
  void write(int value);
```

Functii folosite din biblioteca LiquidCrystal.h:

```
  LiquidCrystal(uint8_t rs, uint8_t enable, uint8_t d0, uint8_t d1, uint8_t
d2, uint8_t d3);
  void begin(uint8_t cols, uint8_t rows, uint8_t charsize = LCD_5x8DOTS);
  void clear();
  void setCursor(uint8_t, uint8_t);
```

Codul sursa din cadrul aplicatiei:

```
#include "DHT.h"
#include <Servo.h>
#include <LiquidCrystal.h>

#define DHTPIN 2     // Digital pin connected to the DHT sensor

#define SERVOPIN 9   // Digital pin connected to the micro servo motor

#define WATERSENSORPIN A4   // Analog pin connected to the water sensor

#define LIGHTPIN A0   // Analog pin connected to the photoresistor

#define POWER_PIN 7  // Digital pin used to power up the water sensor when
the value is scanned

#define DHTTYPE DHT11   // DHT 11

#define water_limit 130

#define light_limit 400

#define humidity_limit 30.0
```

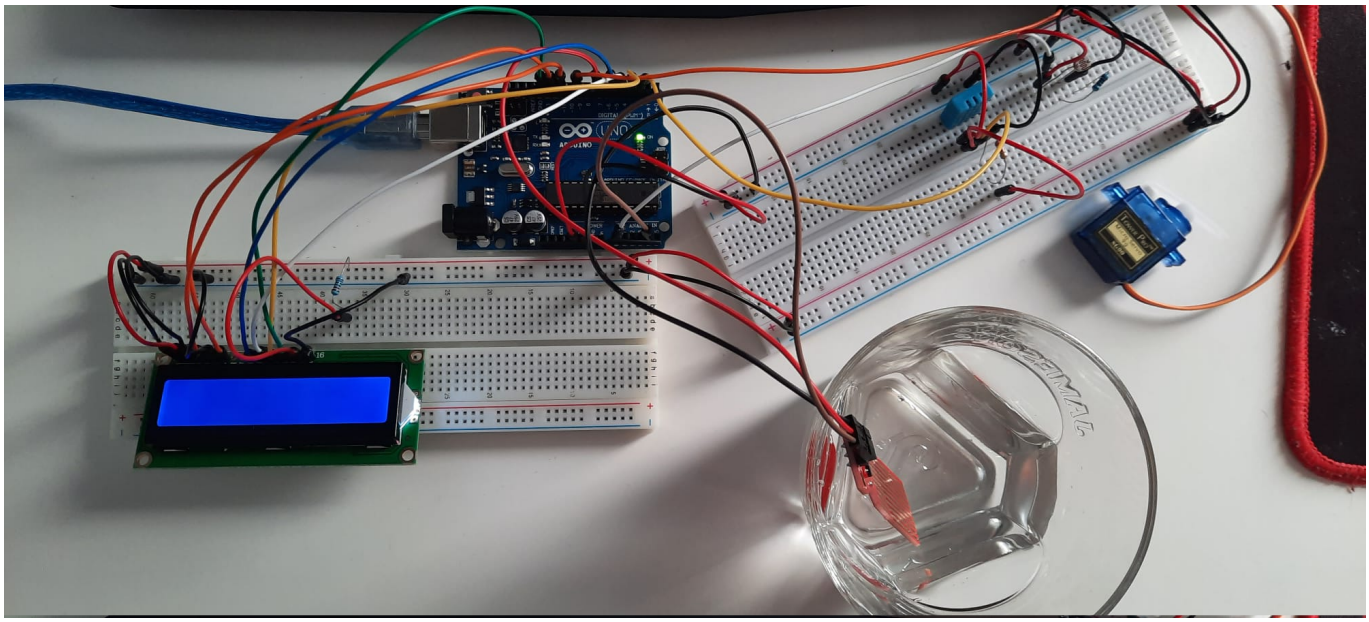
```
#define temp_limit 24.0

// DHT11 sensor init

DHT dht(DHTPIN, DHTTYPE);

// mircroservo motor

Servo servo;

LiquidCrystal lcd(12, 11, 5, 4, 3, 13);

// pump on or off

bool on = false;

// micro servo motor delay

int movement_delay = 300;

void setup() {
  Serial.begin(9600);

  lcd.begin(16, 2);

  servo.attach(SERVOPIN);

  dht.begin();

  pinMode(POWER_PIN, OUTPUT);   // configure D7 pin as an OUTPUT
  digitalWrite(POWER_PIN, LOW); // turn the sensor OFF
}

void loop() {
  // Wait a few seconds between measurements.
  delay(2000);

  // counter for the weather conditions
  // if the count is bigger than 2 than the pump is activated and water is
realesed
  int conditions_count = 0;

  // Reading temperature or humidity takes about 250 milliseconds!
  float h = dht.readHumidity();
  // Read temperature as Celsius (the default)
  float t = dht.readTemperature();

  // light photoresistor read value
  int light_value = analogRead(LIGHTPIN);

  // Check if any reads failed and exit early (to try again).
```

```
  if (isnan(h) || isnan(t)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }

  Serial.print(F(" Humidity: "));
  Serial.print(h);
  Serial.print(F("%  Temperature: "));
  Serial.print(t);
  Serial.print(F("C "));

  Serial.println("Light Value :");
  Serial.print(light_value);

  // water sensor

  int water_value = 0;

  digitalWrite(POWER_PIN, HIGH);  // turn the sensor ON
  delay(10);                      // wait 10 milliseconds

  water_value = analogRead(WATERSENSORPIN); // read the analog value from
sensor

  Serial.print("Water value: ");
  Serial.println(water_value);

  digitalWrite(POWER_PIN, LOW);   // turn the sensor OFF

  delay(1000);

  if (light_value > light_limit){
    conditions_count++;
  }

  if (h < humidity_limit){
    conditions_count++;
  }

  if (t > temp_limit){
    conditions_count++;
  }

  if (water_value < water_limit){
    conditions_count++;
  }

  Serial.print(conditions_count);

  if (conditions_count > 2){
    on = true;
```

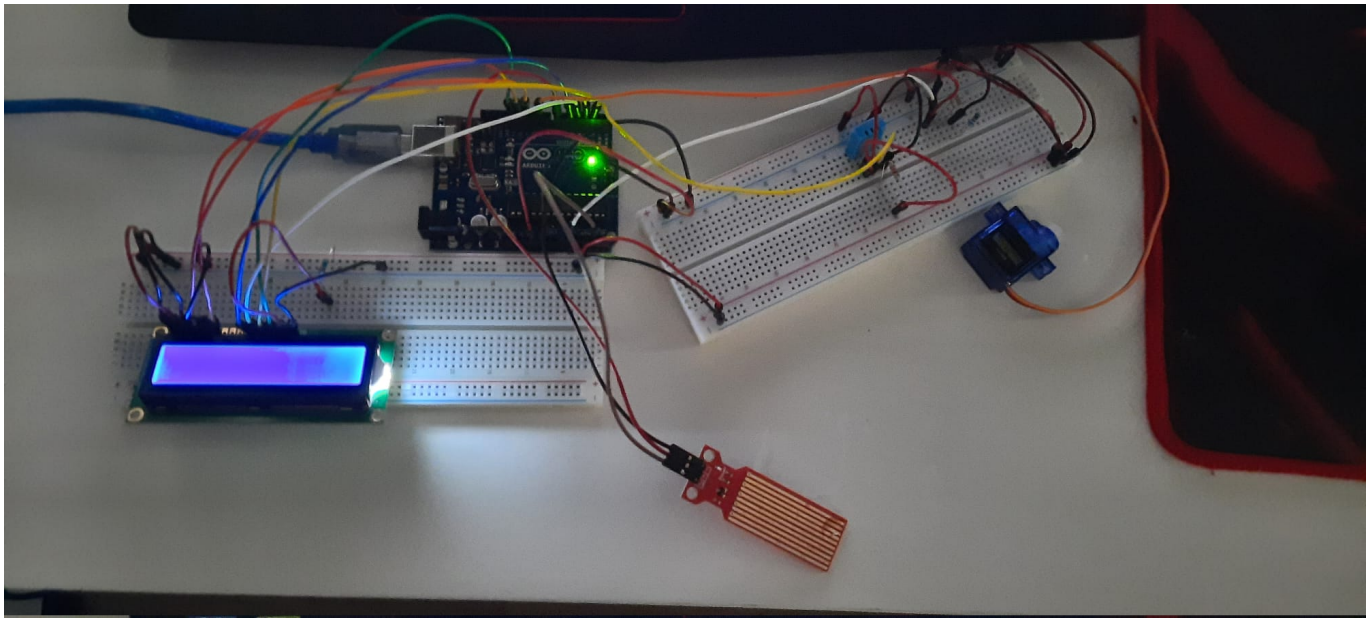
```
  } else {
    on = false;
  }

  // servo motor
  if(on){
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Pouring...");
    servo.write(0);
    delay(movement_delay);
    servo.write(90);
    delay(movement_delay);
    servo.write(180);
    delay(movement_delay);
    servo.write(90);
    delay(movement_delay);
    servo.write(180);
    delay(movement_delay);
  } else {
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Idle");
  }
}
```

Rezultate Obţinute

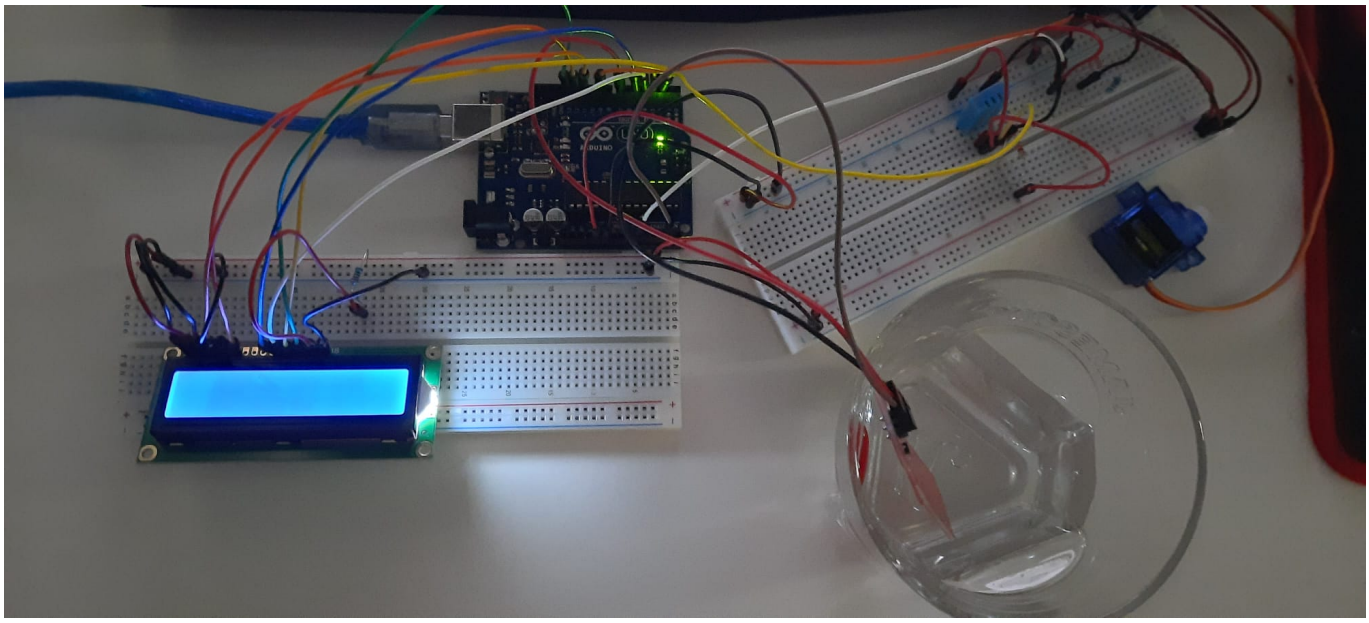
Cazul 1: Lumina, umiditatea si temperatura activeaza pompa



Cazul 2: Apa, umiditatea si temperatura activeaza pompa



Cazul 3: Pompa inactiva



Au fost testate, de asmenea si cazurile in care temperatura este mai mare sau umiditatea este mai mica decat valoarea definita, o simulare live a fost dificil de obtinut.

Concluzii

Always solder your LCD display, a fost destul de straight-forward proiectul, nu am avut probleme cu componentele, bibliotecile etc. A fost destul de interesanta realizarea unui proiect functional cu un arduino.

Download

O arhivă (sau mai multe dacă este cazul) cu fişierele obţinute în urma realizării proiectului: surse, scheme, etc. Un fişier README, un ChangeLog, un script de compilare şi copiere automată pe uC crează întotdeauna o impresie bună .

Fişierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fişierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume_student** (dacă este cazul). **Exemplu:** Dumitru Alin, 331CC → **:pm:prj2009:cc:dumitru_alin**.

Jurnal

27/05/2022, 18:56: final

Bibliografie/Resurse

Software:

- [1]: https://github.com/adafruit/DHT-sensor-library
- [2]: https://github.com/arduino-libraries/Servo
- [3]: https://github.com/arduino-libraries/LiquidCrystal

Hardware:

- [1]: https://create.arduino.cc/projecthub/pibots555/how-to-connect-dht11-sensor-with-arduino-uno-f4d239
- [2]: https://arduinogetstarted.com/tutorials/arduino-water-sensor
- [3]: https://create.arduino.cc/projecthub/MisterBotBreak/how-to-use-a-photoresistor-46c5eb

Export to PDF

From:
http://ocw.cs.pub.ro/courses/ - **CS Open CourseWare**

Permanent link:
http://ocw.cs.pub.ro/courses/pm/prj2022/arosca/2312

Last update: **2022/06/02 00:33**