Design with microprocessors - Final Project
- Weather station with solar tracker -

Horvath Andrea - Anett

08/01/2021

1 Project specifications

The following project represents a *Weather station with solar tracker* incorporated. The main purpose of this project was to design and implement a device which can display the weather conditions. The weather parameters processed by the device are the following:

- air temperature
- humidity
- precipitation
- position of sun

The data is collected from sensors and is displayed on an LCD. For the solar tracker a panel is used and it rotates on 2 axes in the direction of the sun.

2 Components

In order to implement the project, the following components were used:

- Arduino Uno - R3 with USB cable x1
- DHT11 Temperature and Humidity Sensor Module x1
- Rain drop Sensor Module x1
- LCD 1602 with i2C adapter x1
- Servo Motor SG-90 x2
- Photoresistor/LDR x4
- 4.6 kΩ resistor x4
- 830-Point Breadboard x1
- wires

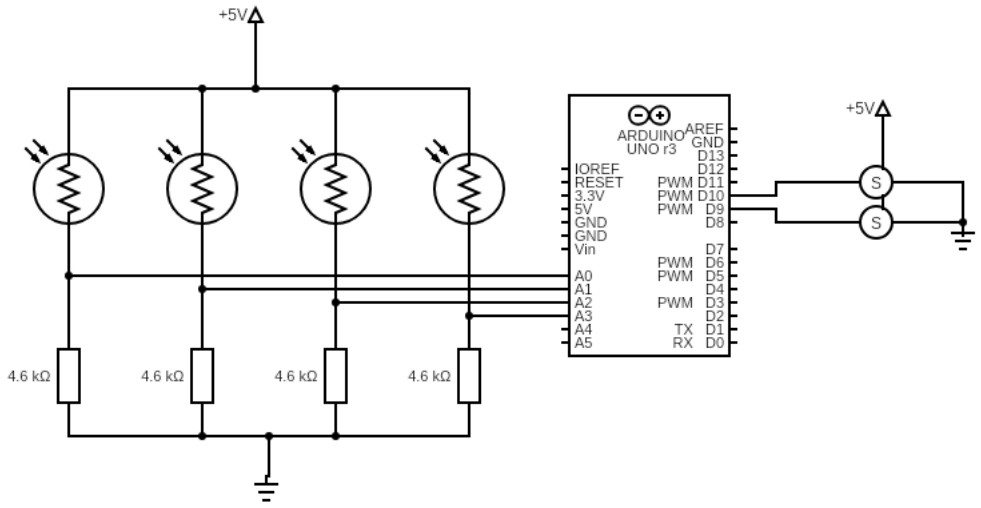
3 Design and implementation

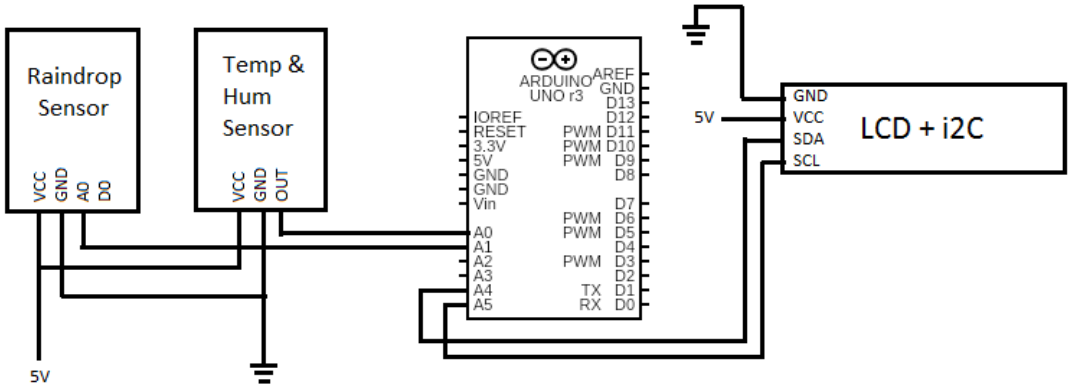
3.1 Design

Because the development board which was available does not have enough analog ports for our implementation, the design was split in two. The problem can be solved using a different board (i.e. Arduino Mega) or using two boards of this kind. Hence, the project is split in two parts:

- the temperature, humidity and precipitation recording part
- the solar tracker part

The circuit diagrams of the two components are presented below.





3.2 Implementation

The code uploaded on the Arduino board is also split in two parts corresponding to the two schematics.

3.2.1 Temperature, humidity and precipitation recording part

```
1 #include <LiquidCrystal_I2C.h>
2 #include <Wire.h>
```

```
3
4  #include "dht.h"
5  #define dht_apin A0 // Analog Pin sensor is connected to
6
7  dht DHT;
8  int hum;
9  int temp;
10 int rainSensVal;
11
12 LiquidCrystal_I2C lcd(0x27, 16, 2);
13 void setup()
14 {
15   Serial.begin(9600);
16   delay(500);
17   lcd.begin(16,2);
18   lcd.clear();
19   lcd.backlight();
20   lcd.setCursor(0,0);
21   lcd.print("Weather␣station");
22   delay(5000);
23
24 }
25
26 void loop()
27 {
28   DHT.read11(dht_apin);
29   rainSensVal = analogRead(A1);
30
31   ////////////////////TEMPERATURE & HUMIDITY//////////////////////
32
33   lcd.clear();
34   lcd.setCursor(0,0);
35   lcd.print("Humidity:␣");
36
37   hum = DHT.humidity;
38   lcd.print(hum);
39
40   Serial.println("Humidity:␣");
41   Serial.print(DHT.humidity);
42
43   lcd.setCursor(0,1);
44   lcd.print("Temperature:␣");
45
46   temp = DHT.temperature;
47   lcd.print(temp);
48
49   Serial.println("Temperature:␣");
50   Serial.print(temp);
51
52   delay(5000);
53
54   ////////////////////RAIN DETECTION//////////////////////////
55
56   Serial.println("Rain␣sensor:␣");
```

```
  Serial.print(rainSensVal);

  if(rainSensVal > 800){
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("    NO RAIN ");
  }

  if(rainSensVal <= 800 && rainSensVal > 600){
    lcd.setCursor(0,0);
    lcd.print("IT IS RAINING   ");
    lcd.setCursor(0,1);
    lcd.print("AMOUNT: LOW ");
  }

  if(rainSensVal <= 600 && rainSensVal > 460){
    lcd.setCursor(0,0);
    lcd.print("IT IS RAINING   ");
    lcd.setCursor(0,1);
    lcd.print("AMOUNT: MEDIUM ");
  }

  if(rainSensVal < 460){
    lcd.setCursor(0,0);
    lcd.print("IT IS RAINING   ");
    lcd.setCursor(0,1);
    lcd.print("AMOUNT: HIGH    ");
  }

  delay(5000);

}
```

3.2.2 Solar tracker part

```
#include <Servo.h>

Servo servoHorizontal;
int servoH = 0;
int servoHLimitHigh = 160;
int servoHLimitLow = 20;

Servo servoVertical;
int servoV = 0;
int servoVLimitHigh = 160;
int servoVLimitLow = 20;

int ldrtopl = 2; //top left LDR
int ldrtopr = 1; //top right LDR
int ldrbotl = 3; // bottom left LDR
int ldrbotr = 0; // bottom right LDR

 void setup ()
```

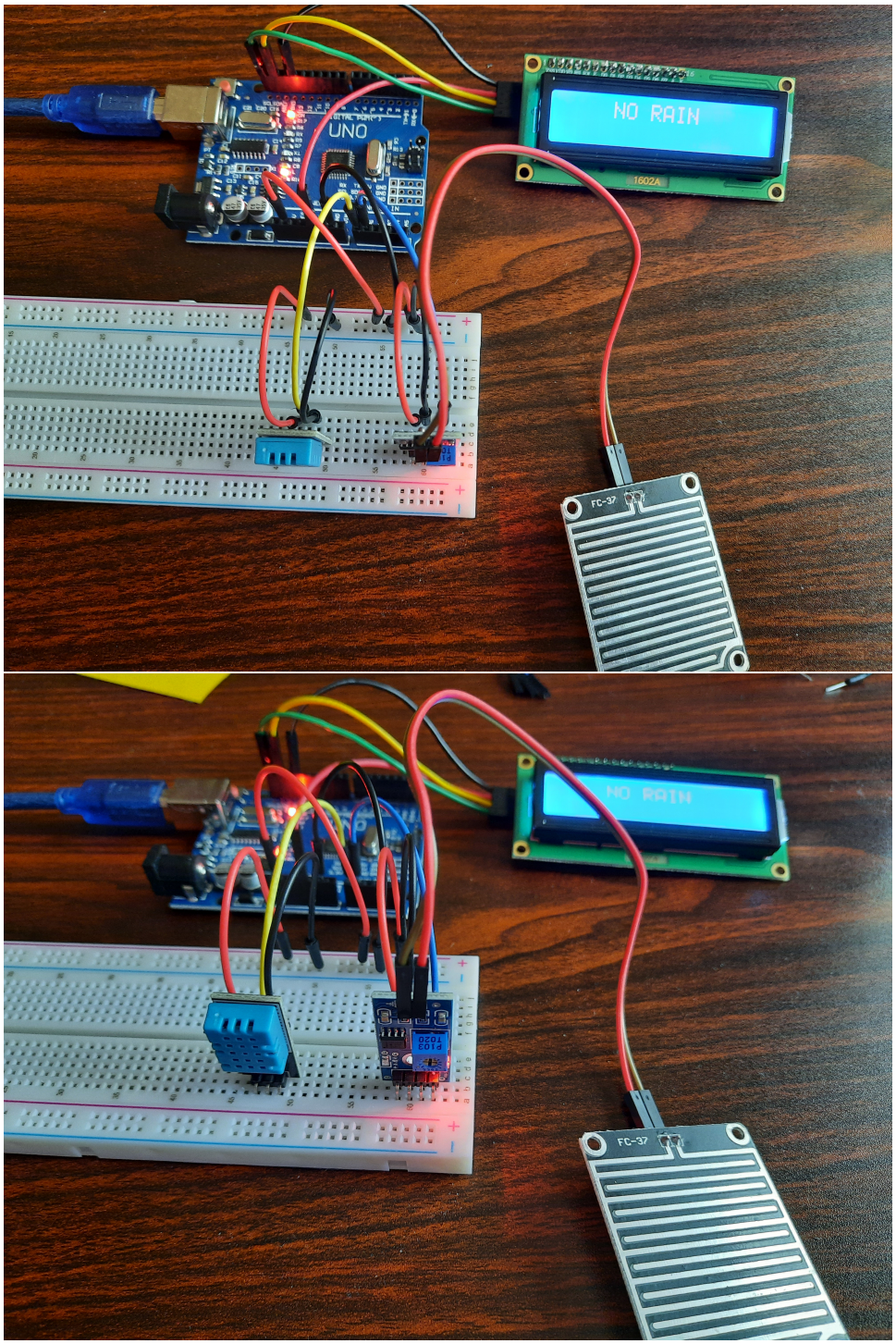
```
 {
  servoHorizontal.attach(11);
  servoHorizontal.write(0);
  servoVertical.attach(9);
  servoVertical.write(0);
  delay(500);
 }

void loop()
{
  servoH = servoHorizontal.read();
  servoV = servoVertical.read();

  //capturing analog values of each LDR
  int topl = analogRead(ldrtopl);
  int topr = analogRead(ldrtopr);
  int botl = analogRead(ldrbotl);
  int botr = analogRead(ldrbotr);

  // calculating average
  int avgtop = (topl + topr) / 2; //average of top LDRs
  int avgbot = (botl + botr) / 2; //average of bottom LDRs
  int avgleft = (topl + botl) / 2; //average of left LDRs
  int avgright = (topr + botr) / 2; //average of right LDRs

  if (avgtop < avgbot)
  {
    servoVertical.write(servoV +1);
    if (servoV > servoVLimitHigh)
     {
      servoV = servoVLimitHigh;
     }
    delay(10);
  }
  else if (avgbot < avgtop)
  {
    servoVertical.write(servoV -1);
    if (servoV < servoVLimitLow)
  {
    servoV = servoVLimitLow;
  }
    delay(10);
  }
  else
  {
    servoVertical.write(servoV);
  }

  if (avgleft > avgright)
  {
    servoHorizontal.write(servoH +1);
    if (servoH > servoHLimitHigh)
    {
    servoH = servoHLimitHigh;
```

```
    }
    delay(10);
  }
  else if (avgright > avgleft)
  {
    servoHorizontal.write(servoH -1);
    if (servoH < servoHLimitLow)
     {
     servoH = servoHLimitLow;
     }
    delay(10);
  }
  else
  {
    servoHorizontal.write(servoH);
  }
  delay(50);
}
```

4 Testing

In this section some pictures of the physical implementation of the project will be presented. Also some demonstrative videos presenting the functionality of the project will be handed in together with the documentation.

4.1 Temperature, humidity and precipitation recording part



4.1.1 Solar tracker part