DESIGN LAB – REPORT

WEATHER STATION

AUTHORS: WOJCIECH HAJDUK, JAN SOBOTA, JAKUB ZAMIRSKI

LEADER: AGNIESZKA DĄBROWSKA – BORUCH

PROJECT DESCRIPTION

The purpose of the project we have created is to measure the environmental changes, using integrated sensors as for example: temperature and moisture sensor DHT11, rainfall sensor FC37 and pressure sensor GY-BMP 280 V3.3. Everyone had special role in realization of the project.

TEAM

- ***Wojtek (programming):***

Wojtek's task was to create code used to colleting data from sensors and logic of the system. Wojtek worked on interactions between every module, assuring their proper integration.

- ***Jan (hardware):***

Jan's role was to implement physical aspect of the project. He connected sensors to the board, providing stable connections and power supply. His task was also to optimize sensors placement on the board.

- ***Jakub (documentation):***

Jakub have dealt with documentation of the project, also creation of technical specification, documentation of connections and checking compatibility of chosen sensors.

REALIZATION OF THE PROJECT

- ***Wojtek***

Wojtek focused on writing control code, that collected data from every sensor, he proceeded it and held decisions depending on the effects to optimize the code. He also worked on warning module, which informs about extreme atmospheric circumstances.

- ***Jan***

Jan focused on physical aspect of the project. He connected sensors to the board and checked if the connections are stable. Ha also was handling optimization of the sensors placement to get best results.

- ***Jakub***

Jakub held documentation of the project. He worked on technical specification of the project, describing every element of the project. He also worked on choosing components.

Biggest problem was integration of every component of the weather station. Wojtek had to deal with efficient software that communicates with hardware made by Jan. Jakub had to check if all components are compatible and fulfills project declarations.

Cooperation and appointed roles made the work efficient. Team was able to make weather station project running. Everyone had key role, that allowed to make swift integration between hardware and software and making complex documentation.

In the future project can be developed with extra functions as remote access to data, trends analysis or integration with cloud platforms. And so on its able to optimize sensors placement to get more accurate measurements.

Project ended well, the team worked efficiently, using everyone skills and specializations to make complex weather station based on IoT.

CODE

The code is a script for reading data from sensors (temperature, humidity, atmospheric pressure and rain sensor) and presenting the results on the LCD display and LEDs.

```
I2C1.setup({ scl: B8, sda: B9 });

var analogValueFS37;
var dht = require("DHT11").connect(C11);
var bme = require("BME280").connect(I2C1);
var lcd = require("HD44780").connect(A0, A1, C0, C1, C2, C3);

var fs37LRain = 0.7;
var fs37MRain = 0.4;
var fs37HRain = 0.2;

var data;
var temp_act;
var press_act;

var LED1 = new Pin(A2);
var LED2 = new Pin(A3);
var LED3 = new Pin(A4);

var on = false;

LED1.write(on);
LED2.write(on);
LED3.write(on);
```

This part of the code is responsible for the initialization of variables and sensors, but we also create the necessary variables that we will need in the next steps.

```
function readFS37(callback) {
  analogValueFS37 = analogRead(C5);
  callback(analogValueFS37);
}

function readDHT11(callback) {
  dht.read(function (dhtData) {
    callback(dhtData);
  });
}

function readBME280(callback) {
  data = bme.readRawData();
  var temp_cal = bme.calibration_T(bme.temp_raw);
  var press_cal = bme.calibration_P(bme.pres_raw);
  temp_act = temp_cal / 100.0;
  press_act = press_cal / 100.0;
  callback({ temp: temp_act, pr: press_act });
}
```

Here we have three functions that are responsible for reading data from the sensors - these are simple functions. The first function performs an analog value reading. The second function uses a ready-made library created by the board manufacturer. The third function uses I2C.

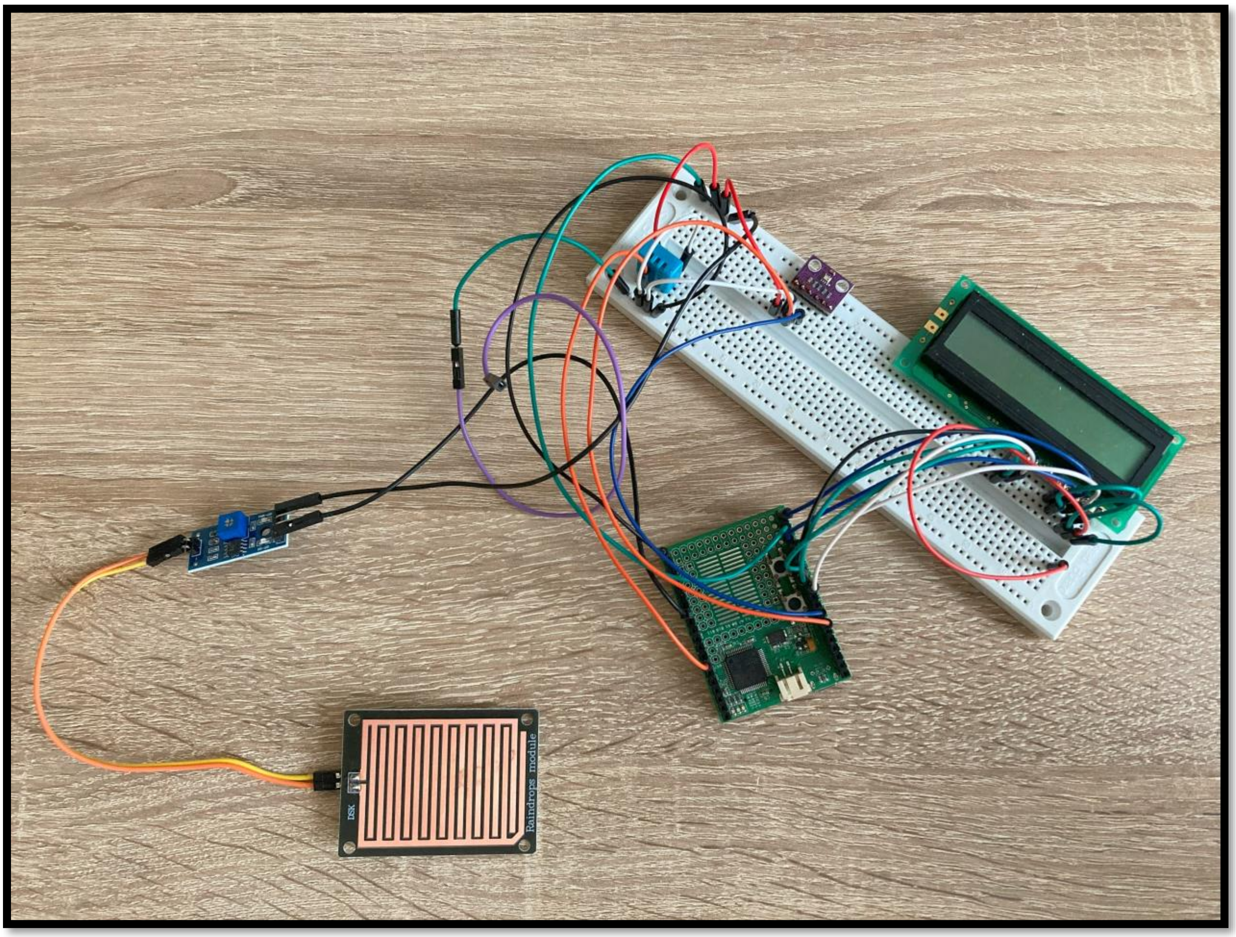
```
function showReadingsOnLCD() {
  readBME280(function (bmeData) {
    lcd.clear();
    if (bmeData && bmeData.temp > 30.0) {
      lcd.print(Math.floor(bmeData.temp) + " *Hot*");
    } else if (bmeData && bmeData.temp < 0.0) {
      lcd.print(Math.floor(bmeData.temp) + " *Cold*");
    } else {
      lcd.setCursor(0, 0);
      lcd.print(Math.floor(bmeData.temp) + "C");
    }

    readDHT11(function (dhtData) {
      lcd.setCursor(4, 0);
      if (dhtData && dhtData.rh > 70.0) {
        lcd.print(dhtData.rh + "%");
      } else {
        lcd.print(dhtData.rh + "%");
      }

      readBME280(function (bmeData) {
        lcd.setCursor(0, 1);
        if (bmeData && bmeData.pr < 1013.25) {
          lcd.print(Math.floor(bmeData.pr) + "hPa");
        } else if (bmeData && bmeData.pr > 1013.25) {
          lcd.print(Math.floor(bmeData.pr) + "hPa");
        } else {
          lcd.print(Math.floor(bmeData.pr) + "hPa");
        }

        readFS37(function (analogValueFS37) {
          if (analogValueFS37 > fs37LRain) {
            LED1.write(0);
            LED2.write(1);
            LED3.write(0);
          } else if (analogValueFS37 > fs37MRain) {
            LED1.write(0);
            LED2.write(0);
            LED3.write(1);
          } else if (analogValueFS37 > fs37HRain) {
            LED1.write(1);
            LED1.write(0);
            LED2.write(0);
          }
        });
```

This part of the code is responsible for displaying all values on the LCD. The bmeData function is responsible for ensuring that the variables read from the sensors are numbers without commas. We also have an LED control that works in the following way: when rainfall is low, the GREEN LED lights up. During average rainfall, the BLUE LED lights up. In case of very heavy rainfall, the RED LED lights up. These values are initialized at the very beginning of the code during initialization.

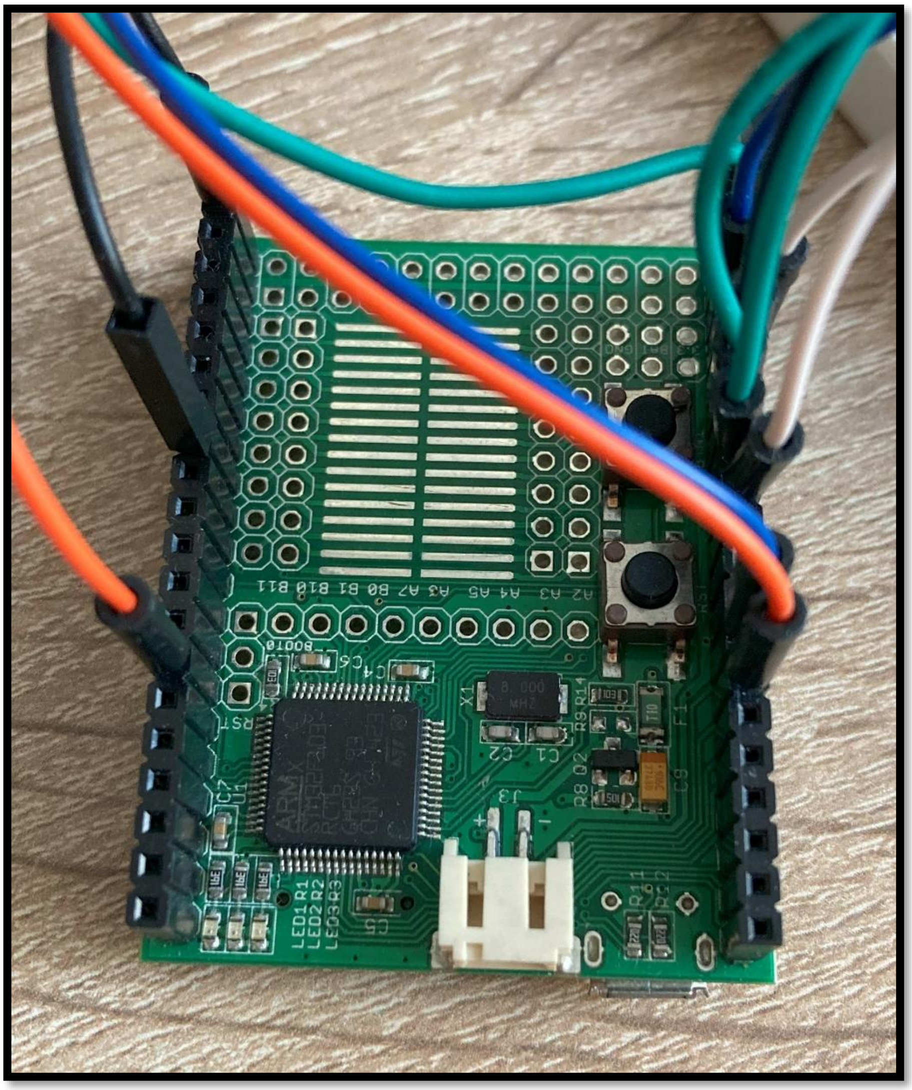
```
setInterval(showReadingsOnLCD, 1000);
```

Activating the display function

THE WHOLE SYSTEM



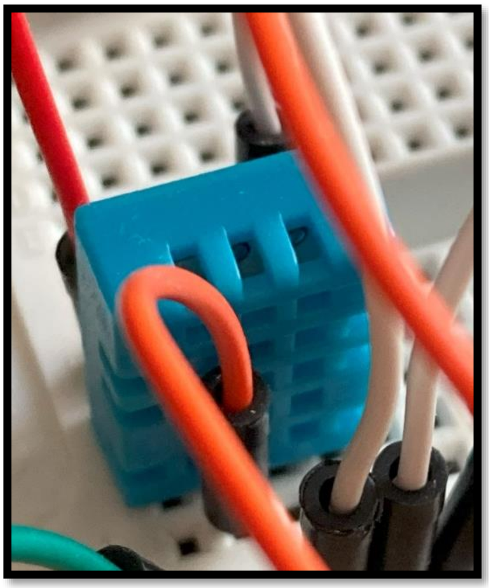
MOTHERBOARD



For this project we use Original Espruino Board. It is a tiny USB-enabled microcontroller with SD card that can be programmed in JavaScript. It just need to be plug it into computer and started to work with the Web IDE - no software installation needed.

SENSORS

- **DHT11**

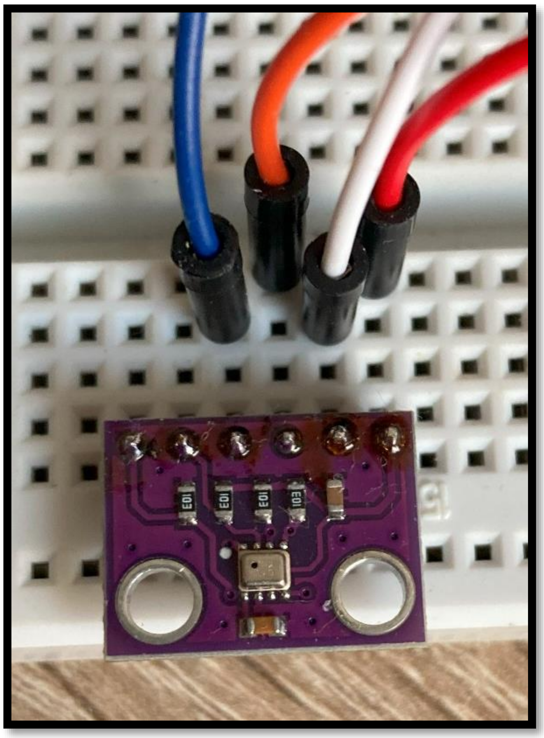


The DHT11 sensor uses a thermosensor to measure temperature and a resistive humidity sensor. These readings are converted into a digital signal that is available for reading by the microcontroller. The sensor is equipped with a built-in integrated circuit that performs measurement and analog-to-digital conversion.

Temperature: Typically, the temperature measurement range of the DHT11 sensor is from 0°C to 50°C.

Humidity: The air humidity measurement range is usually from 20% to 80%.

- **BME280**



Action: The BME280 sensor measures three key atmospheric parameters: temperature, atmospheric pressure and air humidity. It works on the principle of changing the resistance of materials depending on environmental conditions. The measured data is processed by the built-in integrated circuit, which provides digital data ready for reading.

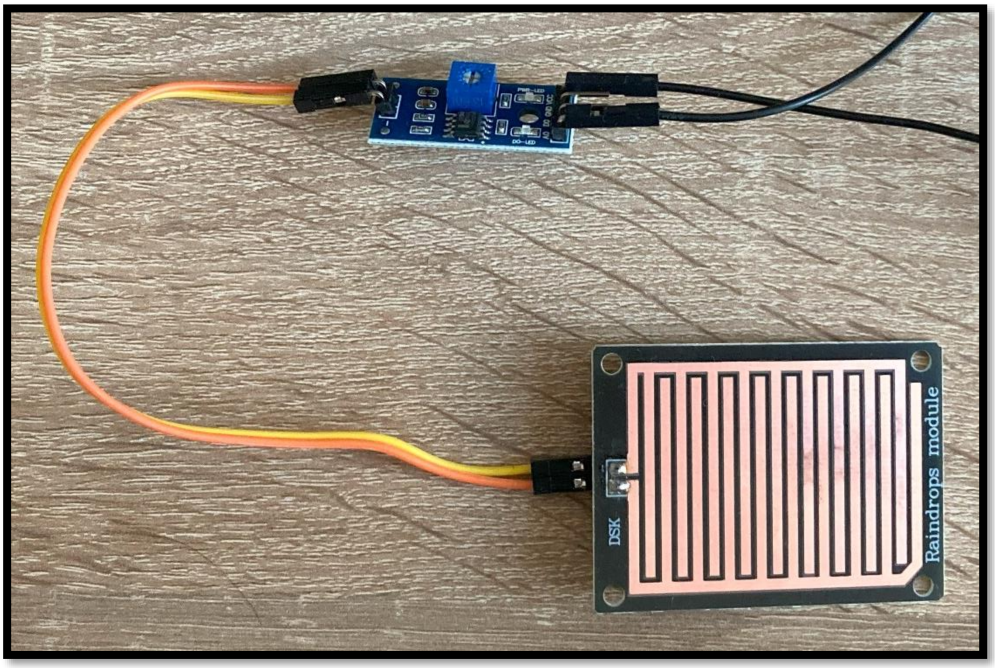
Temperature: The measurement range is typically -40°C to +85°C.

Atmospheric pressure: The measurement range is usually from 300 hPa to 1100 hPa.

Air humidity: The measurement range is usually from 0% to 100%.

BME280 communicates with the microcontroller via interfaces such as I2C (Inter-Integrated Circuit) or SPI (Serial Peripheral Interface), which facilitates its integration with various microcontroller platforms. In our case, we communicate using I2C.

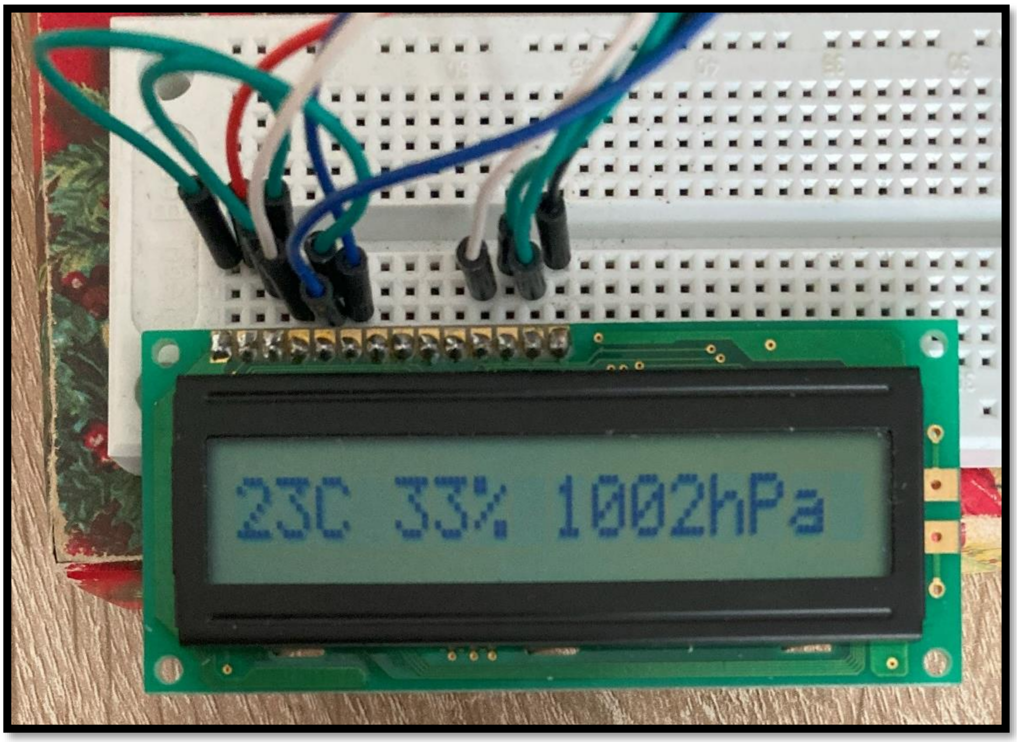
- **Raindrops module**



The Raindrop sensor module operates on a simple and intuitive principle. It features a set of exposed copper paths that function as variable resistors. The resistance of these paths changes proportionally to the quantity of water present on the sensor's surface. Normally, these paths are not connected, but when water is present, they become bridged, allowing for the detection of rainfall.

We also use the fs37 circuit which reads values from Raindrop and sends an analog value in the range of 0 to 1V. In our project, we assumed that if this value is more than 0.7V, it is low rain. Between 0.7V and 0.4V medium rain. And in the range of 0.4V to 0.2V there is high rain.

- **LCD**



We used the simplest single-segment display that displays temperature, humidity and atmospheric pressure

Future expansion

We would definitely like to expand the system so that it is not based on such a breadboard, but rather make our own board. It would also be useful to make a housing for our system. We also replaced the LCD with an OLED, but a single-segment display can display little data. It would also be possible to expand the system with its own power supply, for example a system with batteries rather than a power bank.