TECHNICAL UNIVERSITY OF MOLDOVA
FACULTY OF COMPUTERS, INFORMATICS AND MICROELECTRONICS
DEPARTMENT OF SOFTWARE ENGINEERING AND AUTOMATICS

Report of laboratory work №3

Theme: Sensors

Fulfilled: st.gr. FAF-202 **Pogorevici Daniel**

Controlled: univ. lecturer **Moraru Dumitru**

Chișinău 2023

1. The Task of the Laboratory Work

Create an MCU-based application that will receive a signal from at least 3 signal sources (analog and digital sensors), condition the signals, and display the physical parameter (temperature, pressure, etc.) at a terminal Serial.

2. Implementation

For the following laboratory work I used 5 different sensors to display their physical parameters in the serial terminal. In order to do this I used several libraries such as:

1. DHT: A library for reading temperature and humidity data from DHT11/22 sensors.
2. Ultrasonic: A library for reading distance data from ultrasonic sensors.
3. Adafruit_BMP085: A library for reading pressure and temperature data from BMP180 sensors.

In my script I set up pins for various sensors and initialize them in the **setup()** function. It then enters the **loop()** function, which repeatedly reads sensor data, prints it to the Serial Monitor, and delays for 3 seconds before repeating.

```
DHT dht(DHT_PIN, DHT11);
Ultrasonic ultrasonic(ULTRASONIC_ECHO_PIN, ULTRASONIC_TRIG_PIN);
Adafruit_BMP085 bmp;

void setup() {
  pinMode(TMP36_PIN, INPUT);
  pinMode(ULTRASONIC_TRIG_PIN, OUTPUT);
  pinMode(ULTRASONIC_ECHO_PIN, INPUT);
  dht.begin();
  Wire.begin();
  Serial.begin(9600);
  if (!bmp.begin(BMP180_ADDRESS)) {
    Serial.println("Could not find a valid BMP180 sensor");
    while (1) {}
  }
}
```

Fig. 1.1 – setup()

Particularly, I dealt with the following sensors:

1. TMP36 temperature sensor

This sensor is read using the **analogRead()** function, which reads an analog voltage value from the sensor and converts it to a temperature value using the TMP36's calibration. The temperature is then printed to the Serial Monitor.

```
// Read TMP36 temperature sensor
float tmp36Raw = analogRead(TMP36_PIN);
float tmp36Voltage = tmp36Raw * 5.0 / 1023;
float tmp36Temperature = (tmp36Voltage - 0.5) * 100;
```

Fig. 1.2 – TMP36

2. LDR light sensor

This sensor is also read using the **analogRead()** function, which reads an analog voltage value from the sensor. The raw value is printed to the Serial Monitor.

```
// Read LDR light sensor
int ldrRaw = analogRead(LDR_PIN);
```

Fig. 1.3 - LDR

3. DHT11 temperature and humidity sensor

This sensor is read using the **readTemperature()** and **readHumidity()** functions provided by the DHT library. The temperature and humidity values are then printed to the Serial Monitor.

```
// Read DHT11 temperature and humidity sensor
float dhtHumidity = dht.readHumidity();
float dhtTemperature = dht.readTemperature();
```

Fig. 1.4 – DHT11

4. Ultrasonic sensor

This sensor is read by sending a 10 microsecond pulse to the sensor using the **digitalWrite()** function, and measuring the time it takes for the pulse to bounce back using the **pulseIn()** function. The distance to the object is then calculated based on the time and the speed of sound and printed to the Serial Monitor.

```
// Read Ultrasonic sensor
digitalWrite(ULTRASONIC_TRIG_PIN, LOW);
delayMicroseconds(2);
// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(ULTRASONIC_TRIG_PIN, HIGH);
delayMicroseconds(10);
digitalWrite(ULTRASONIC_TRIG_PIN, LOW);
// Reads the echoPin, returns the sound wave travel time in microseconds
float duration = pulseIn(ULTRASONIC_ECHO_PIN, HIGH);
// Calculating the distance
float distance= duration*0.034/2;
```

Fig. 1.5 – Ultrasonic

5. BMP180 pressure sensor

This sensor is read using the **readPressure()** function provided by the Adafruit_BMP085 library. The pressure value is then printed to the Serial Monitor.

```
// Read BMP180 pressure sensor
float pressure = bmp.readPressure() / 100.0F;
```

Fig. 1.6 – BMP180

Latest, I have the output part where I print in the terminal all the sensor data.

```
// Print sensor data to Serial Monitor
Serial.print("TMP36 Temperature: ");
Serial.print(tmp36Temperature);
Serial.println(" C");
Serial.print("LDR Value: ");
Serial.println(ldrRaw);
Serial.print("DHT11 Humidity: ");
Serial.print(dhtHumidity);
Serial.println(" %");
Serial.print("DHT11 Temperature: ");
Serial.print(dhtTemperature);
Serial.println(" C");
Serial.print("Ultrasonic Distance: ");
Serial.print(distance);
Serial.println(" cm  ");
Serial.print("BMP180 Pressure: ");
Serial.print(pressure);
Serial.println(" hPa");
Serial.println("\n");
```

Fig. 1.7 – Print functionality

3. Electrical Scheme

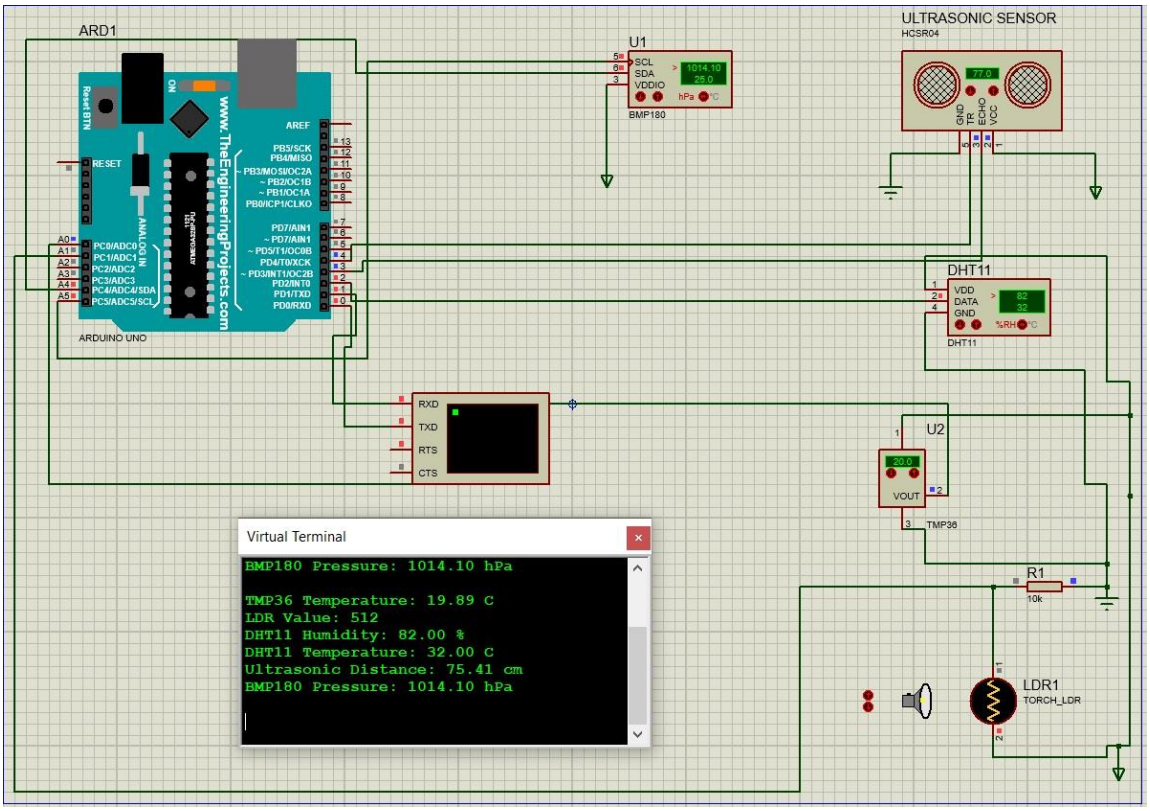

Fig. 2.1 – Electrical scheme

4. Conclusions

In the following laboratory work, I made a program which is capable of reading temperature, humidity, light, distance, and pressure data, making it suitable for a wide range of applications, from environmental monitoring to robotics. The program also demonstrates how to use various Arduino functions and libraries to interact with sensors, making it a useful resource for beginners and experienced Arduino users alike. I also made usage of several libraries to interface with the sensors and sets up pins for each sensor. Then repeatedly reading data from each sensor and printing the results to the Serial Monitor, with a delay of 3 seconds between each iteration.

In conclusion, I got experience with different types of sensors, growing my knowledges in this domain.