LAB 1 REPORT

By

Qais Qamhia

(NB-427)

27/03/2020, Bremen

Jacobs University Bremen

Introduction to Robotics and Intelligent Systems Lab

Dr. Fangning Hu, instructor

Introduction:

Arduino is the brain of the intelligent system, the main concept of this lab and the following ones, is connecting this brain to a specific circuit we create on the breadboard, and programming the brain (The Arduino Uno piece) to receive, or send information (or both) to and from the circuit on the breadboard.

This lab was an introduction to the main parts of this circuit (Resistors, LED's, Potentiometer…etc), their functions, and how they are implemented in our system (Both hardware, and software wise). In addition to the main parts, we also got introduced to the Multimeter, and its proper way of use.

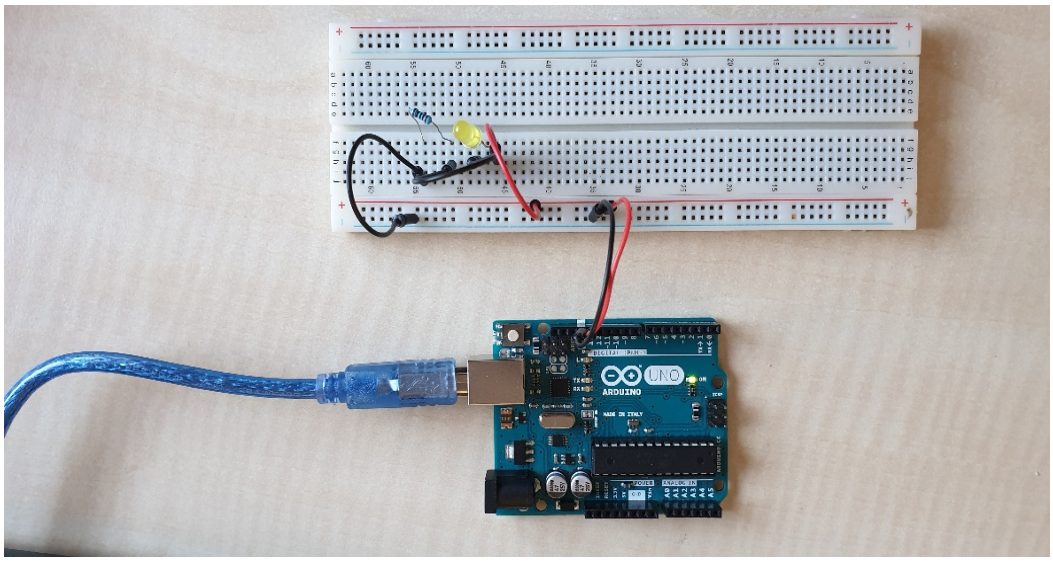
Execution:

Task 1.1/1.2:

In this task, I used the code provided by the basic examples integrated in the Arduino IDE. I uploaded the code to the Arduino board, and then unplugged it (To remove the power source while working with the circuit), then assembled the needed circuit. After that I connected the circuit to the Arduino, and the Arduino to the computer once again to act as a power source.

In the circuit, I connected the positive LED leg to pin #13 in the Arduino board, and connected the negative leg to a 100 Ohms resistor (to ensure our led not getting the full 5V, which will cause a high current going through the LED, which would cause it to burn down), and then to GND (Ground) on the Arduino.

The program defines pin #13, and thus our LED, as an output source. And inside the program's loop, we give this pin high voltage, and then cut it off every second. In other words, the LED turns on and then off every second.



MP4
Task 1_1.mp4

Task 1.3:

Following the steps stated in manual report, I put each cord into the right terminal (Red into HzVΩ and the black one into COM) of the multimeter. I then set the wheel to Ω position for resistance measurement and tried using different measurement basis. The one that was best fitting for finding the needed resistance was 2k Ohm basis. In which I found the following results:

0.15 (150 Ohm) => When setting the basis to 20k, I got 0.01 (Less accurate, because the basis is too large for the resistor)

0.10 (100 Ohm) => When setting the basis to 20k, I got 0.01 (Same result as 150, so we can't differentiate using this basis)

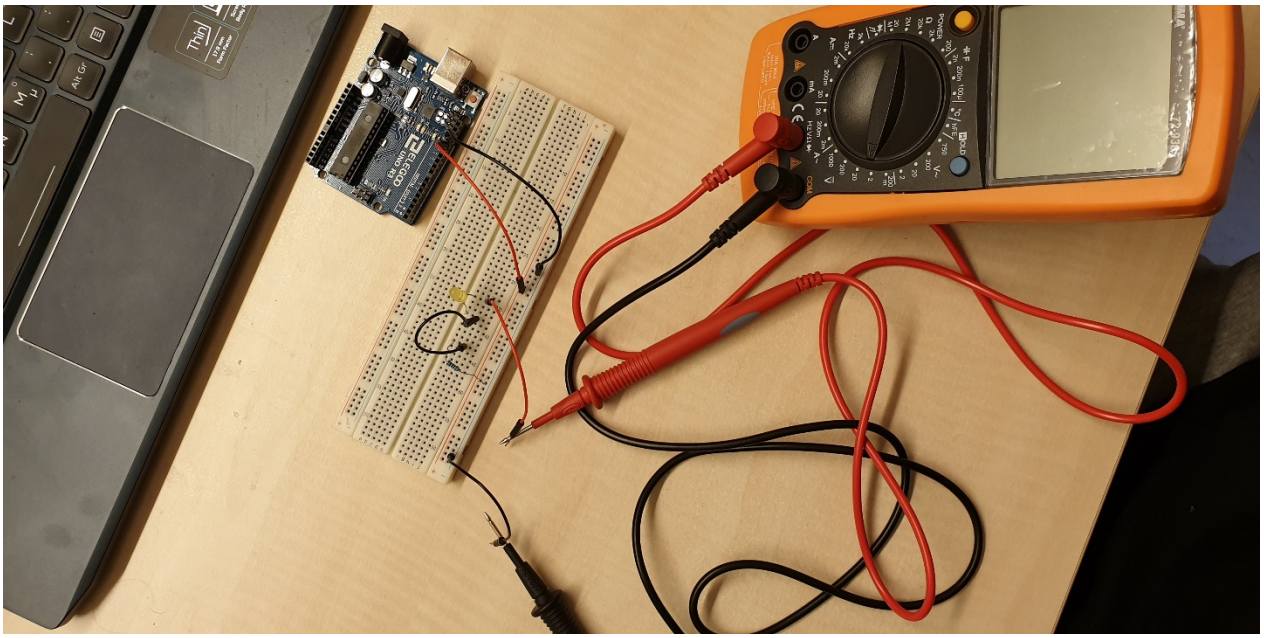
0.001(10 Ohm) => When setting the basis to 200 instead, I got the readings 10-12 Ohms

Task 1.4:

I connected a similar circuit to the one in Task 1.2 and connected the multimeter cords (leads) to the terminals like stated in Task 1.3.
Then switched the multimeter wheel (rotary switch) to the bottom right corner (Labelled DC Voltage Measurement on the Lab manual), specifically to 2 as measuring accuracy (Highest value measured is 2V).

I connected one end to the ground (GND), the other one to the positive leg of the LED, to test the voltage that goes through the LED, after the effect of the resistor, and the LED's own resistance (Vforward).

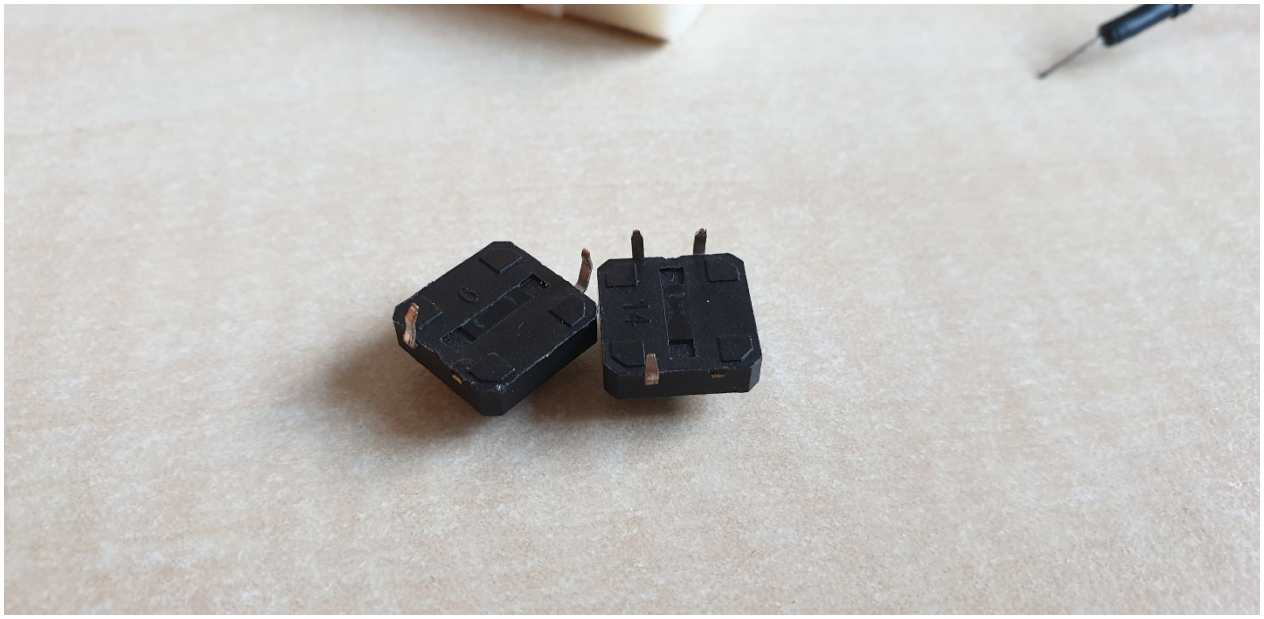
And after plugging the Arduino to the power source (Which, sadly, isn't shown in the picture), my reading was 0.027V.



Task 1.5:

In this task, I rotated the multimeter's switch to the Diode Test sections (The one with the musical note symbol), then put the chords of the multimeter on two of the button's pins and switched them around.
The multimeter beeped whenever I put them on two pins across of each other, meaning that each two pins across of each other are connected.

Note: The only issue I had with this task, and the following one, was having broken button pins the first time I tried doing this Lab, which required me waiting for the next week's session to be able to do it again.



Task 1.6:

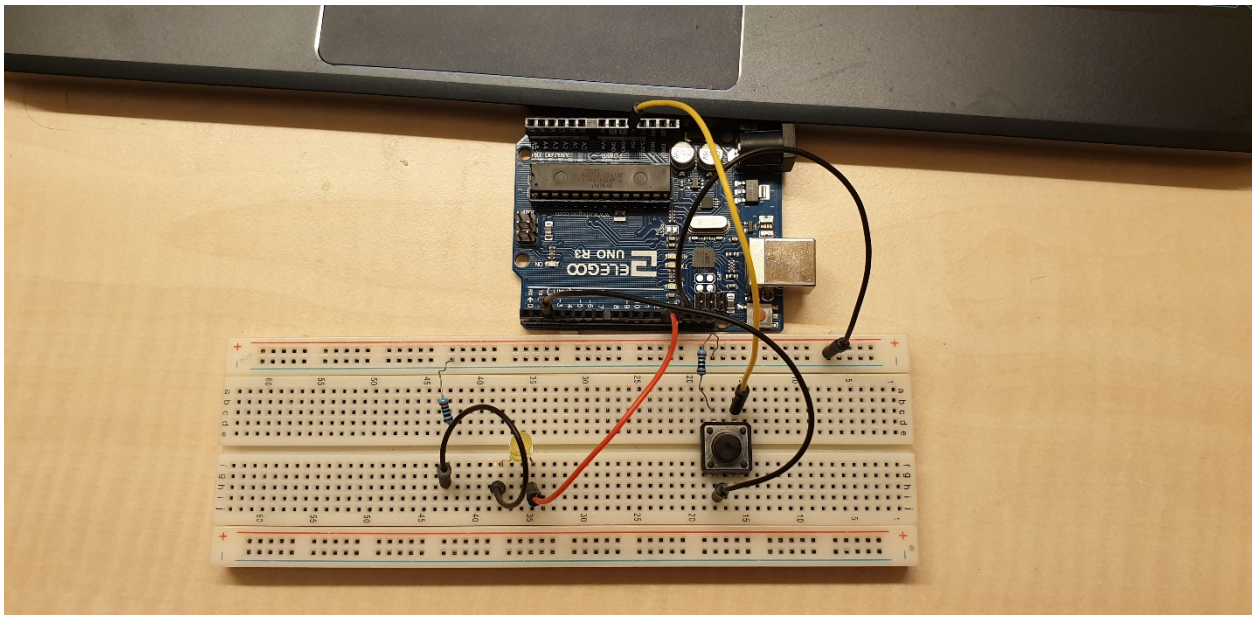
I connected two opposite pins of the button (The ones connected together) with a resistor that goes to the ground, and the other pin to pin #2 on the Arduino.

And connected the other side with 5V from the Arduino board. So that when the button is pressed and the pins are connected, the voltage will go through the button, and pin #2 will have high voltage, which we use as input (High voltage means button is pressed. Low voltage means button is not pressed)

The rest of the circuit is just a normal LED circuit similar to the one I made in the previous tasks. (LED connected to pin #13)

In the program, when the pin #2 has HIGH as input (meaning the button is pressed), we put the LED output as HIGH. And whenever it isn't, we set it to LOW.

So that whenever we press the button the LED lights up, and instantly turns off once we stop pressing the button.



20200402_161401.mp4

Task 1.7:

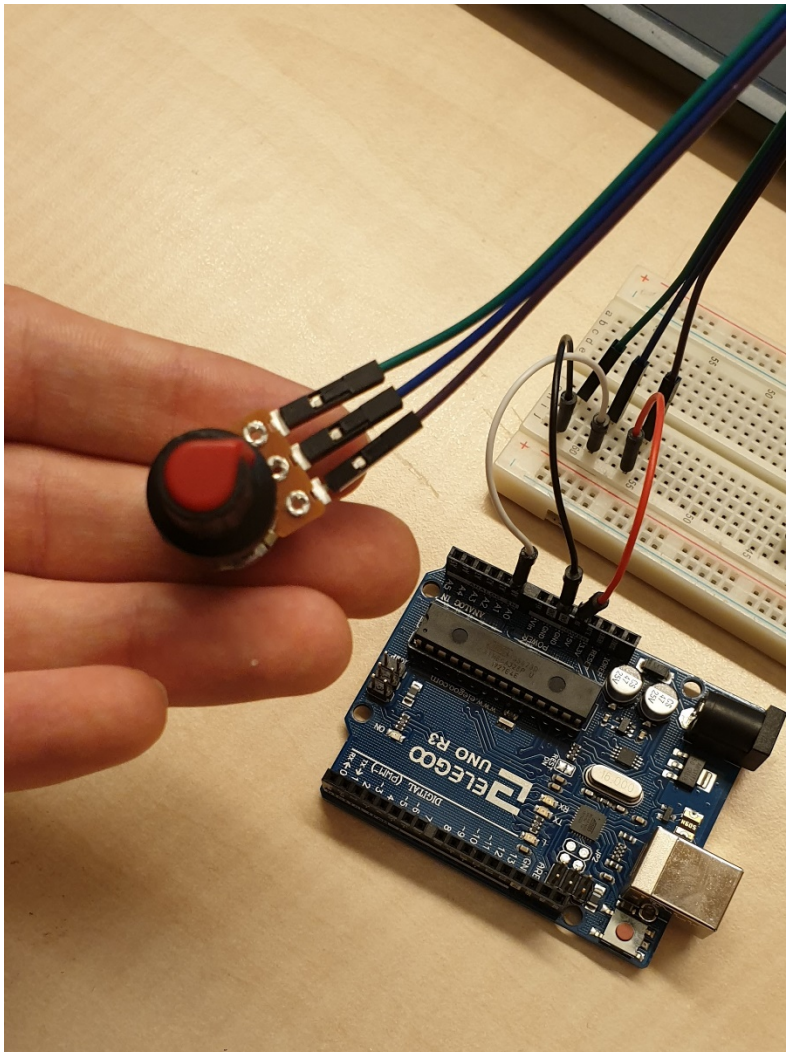
I connected Multimeter arms as stated in Task 1.3 (Red into HzVΩ and the black one into COM) and set the rotary switch to resistance measurement.
I put one arm on the right pin, and the other on the middle pin, and I rotate the knob the resistance goes up and down depending on which way I rotate it. And if I connect the Multimeter handle to left and middle pin (instead of right and middle), the resistance will be inversed and the direction of the knob that used to increase the resistance will now decrease it instead.

Task 1.8:

I connected the Potentiometer pins to the female-to-female connecter, and then connected the right pin to 5V source on the Arduino, the middle to analog input pin A0 and the left pin to GND. And then programmed the Arduino so that it would convert the raw readings from the analog pin (middle pin of the potentiometer) into voltages.

The way it works is basically that we insert high voltage (5V) into the right potentiometer pin, and we connect the left to the ground to close the circuit, and the middle pin will give us the final signal (final voltage) after the effect of the Potentiometer's resistance. But it will output it as analog output (Converts the voltage into values from 0 – 1023), which we convert in the program back to normal voltage values by multiplying by 5 and dividing by 1023.

When I rotated the knob to the right, the voltage would go down (Resistance becomes higher), and rotating to the left would make the voltage readings on the serial higher (Resistance becomes lower)



Conclusion:

The tasks were rather clear and simple. Most of the obstacles I faced were caused by faulty hardware, such that: Broken button pins, Non-working LEDs. And I easily fixed them by asking for new working parts in the next Lab session.

References:

https://www.arduino.cc/reference/en