Project Report
CSCI 43300

Riya Shrestha
Zoe Yang

1. Introduction

In this project, we will be learning to use a single node system, Raspberry Pi to build simple applications using Python programming language. We will be learning about how to control the blinking and brightness of led, how to control simple functions using a button and how to build simple applications for sensors. We will also be learning how to use different resistors, LEDs and sensors in circuit designs.

Hardware platform used: Raspberry Pi 4
Software platforms used:

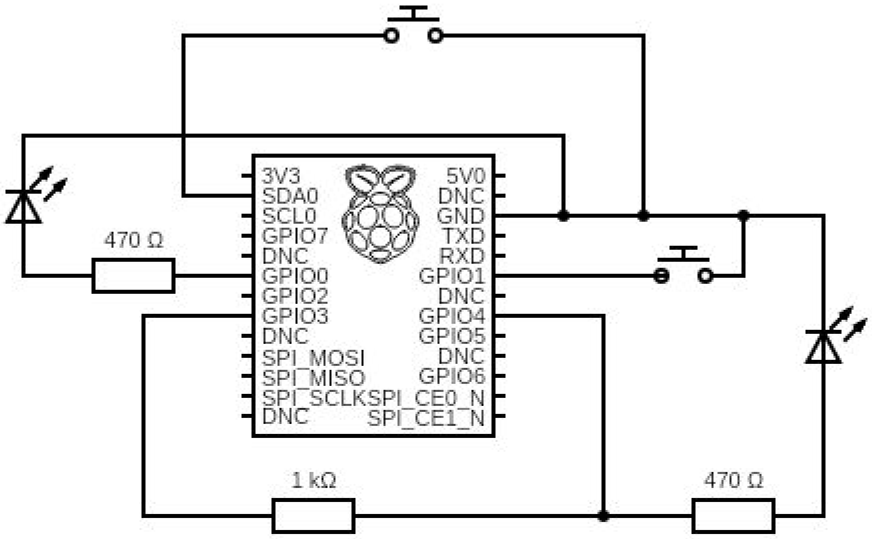
- Raspberry Pi OS
- Python programming language

We also used the temperature sensor DS18B20, different resistors (5.1k ohms, 220 ohms, etc). Different LEDs, buttons, breadboard, female to male wires and jump wires were also used in this project.

2. Project design

Step 1:
We start by setting up a simple circuit with a 220 ohms resistor and a led connecting in series. Then we ran the program “Blink” which was provided in the project requirement which made the led blink once a second.

Step 2:

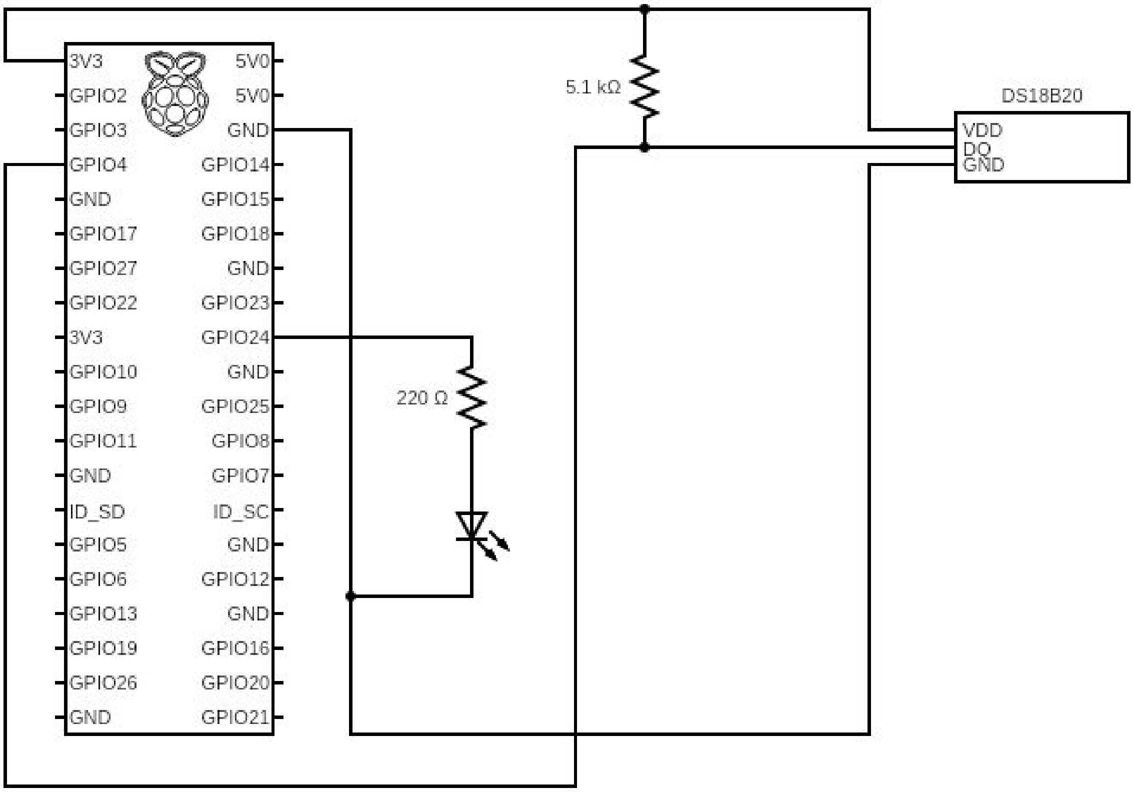


This is the circuit diagram for our step 2.

Generally, there are two modes: frequency mode and brightness mode. As it is described, in frequency mode, led 1 will initially blink once per second for 3 seconds, and then it will blink 0.25s faster every 3 blinks. A “for” loop is used to loop through 4 frequency levels (1s, 0.75s, 0.5s, 0.25s) and when reaching the highest frequency, it will go back to 1 blink/second and continue increasing frequency. When button 1 is pressed, led 1 will stop increasing frequency and stay at the current frequency until button 1 is pressed again.

Button 2 is set for switching the mode. When button 2 is pressed, it will switch to brightness mode, which led 2 will start to blink and led 1 is off. There are two levels of brightness (bright and dim). Led 2 is initially set as bright, and when button 1 is pressed, it will turn dim, and when pressed again it will turn bright. Basically, two resistors (R2 & R3) are connected in series. R2 is connected to GPIO 4, as shown in the diagram, and R3 connecting in series with R2 is on the other side connecting to GPIO 3. So when GPIO 4 is power on an GPIO 3 is off, Led 2 is only connected with one 470 ohms resistors, so it will be bright; when GPIO 3 is on and GPIO 4 is off, Led 2 is connected with two resistors (together 1k + 470 = 1470 ohms), so it will be dim.

Step 3:
First, we need to enable 1-wire interface in Raspberry Pi Configuration OS setting. Then we build our circuit as given in the circuit diagram below.



After we build the circuit and turn the power on, we will find the serial number of our temperature sensor DS18B20 under /sys/bus/w1/devices file which looks like 28-XXXXXXXXX. The file w1_slave under the sensor (28-XXXXX) consists of the current temperature value. On the second line of that file, we can see some value like t=26129 which means the temperature is 26.129 degree Celsius. In our program, we first read the w1_slave file and saved the current temperature value in the variable “tempdata”. Then we converted the temperature to Fahrenheit by applying the formula:

temperature = (temperature/1000) * 9/5 +32

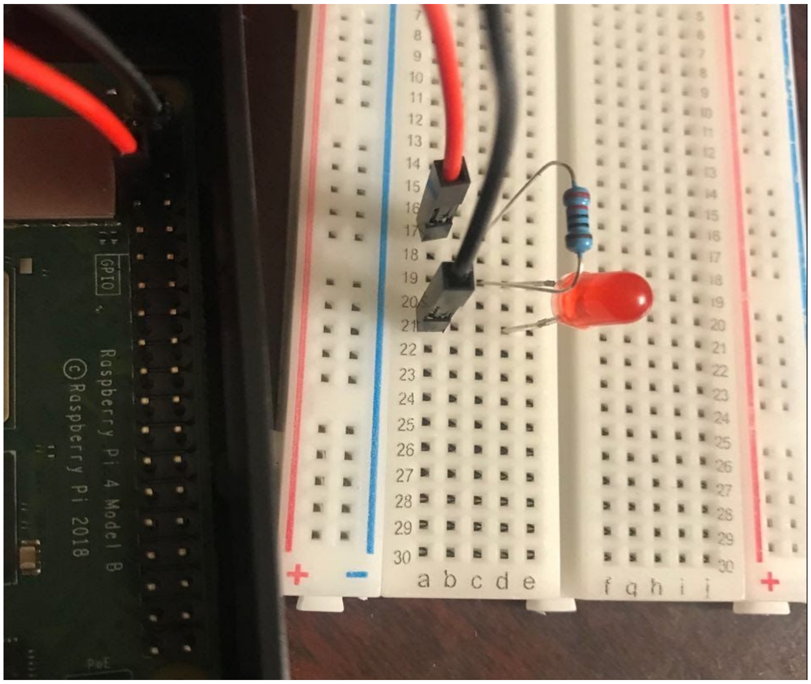
After rounding the temperature in Fahrenheit upto two decimal places, we inserted the condition to check if the temperature is greater than a given threshold value, 75 in our case. If the temperature is greater than the given threshold value, the LED turns on until the temperature drops below the given threshold value. The program stops running if there is a keyboard interaction (Ctrl + C) at any time.

3. Implementation

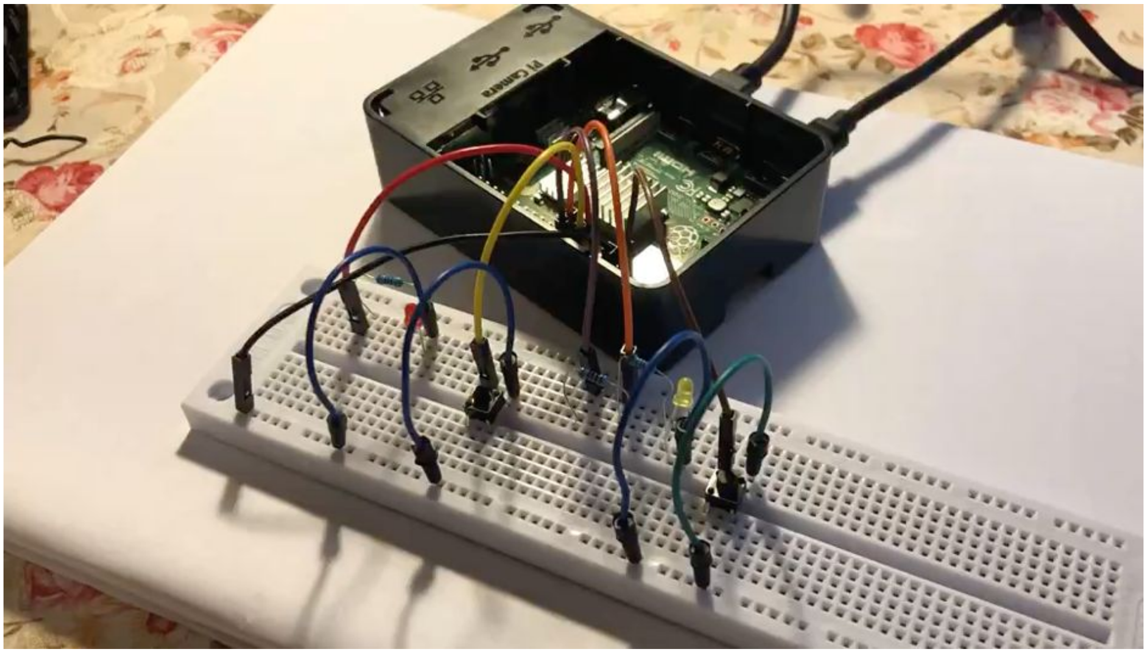
We used the programming language Python for our entire project. We imported the gpiozero, Rpi.GPIO and time libraries for this project. Our project works correctly for each step as defined in the project assignment. The first step makes the LED blink in certain patterns. The second step controls the brightness and frequency of the LED using buttons. The third step uses the DS18B20 sensor to check the temperature constantly and sends the output to turn the LED on if the temperature increases above the threshold.

Some snapshots of the project:

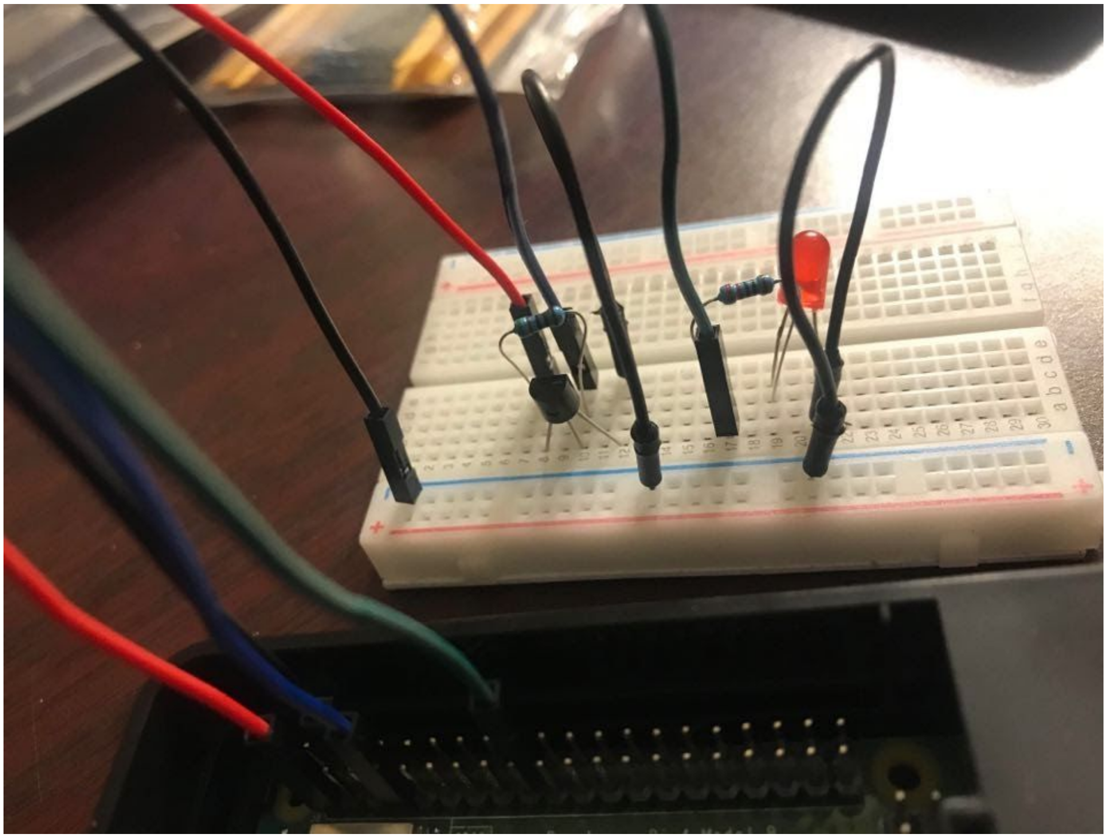
Step 1:



Step 2:



Step 3:



4. Experiences

One thing that we have learnt from this project is that we need to start earlier. Because of taking the class and collaborating with group members remotely, we need to prepare our equipment by ourselves and would meet unexpected issues from time to time. Starting earlier will allow us more time to take care of these problems.

We ran into problems with using the sensor DS18B20. Our sensor was not getting detected by the Raspberry Pi. In the end we figured out there were some connection issues. From this problem, we learned that we need to understand the pin connections of the sensors properly so that it functions correctly. We also learned how to connect the LEDs in the right way which is anode to the resistor and cathode to the ground.

Another challenge we faced was to use the Raspberry Pi terminal remotely.

5. Video demo

Demo for Step2:
https://drive.google.com/file/d/1BFfamgBn9p-Ag9Sdfdi6NyUJ1wX_VEFc/view?usp=sharing

Demo for Step3:
https://drive.google.com/file/d/1HfWEGi8T9Jy4oob343TvWyHx5B6TwbCG/view?usp=sharing