CSE 461 : Introduction to Robotics

Lab Report

Lab No: 01
Lab Task: Using push buttons to control LEDs.

Group Information

Lab Section: 02
Group Number: 02

Group Members:

Name	Student ID
Nowshin Sumaiya	21301276
Md. Raisul Islam	21301406
Samir Yeasir Emon	21301413
Sakib khan	21301278
Farhan Labib	21301238

Task Description:

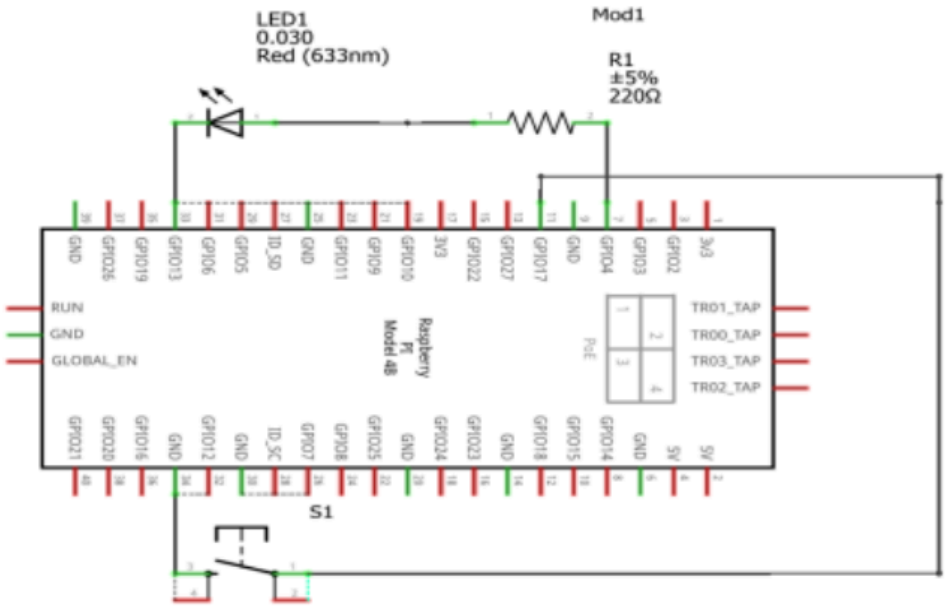
First of all, We started by setting up the Raspberry Pi by inserting the micro SD card that contained the operating system and attaching it to a power source, monitor, keyboard, and mouse. The Raspberry Pi 4 will next be connected to an LED on the breadboard to begin the hardware setup, using the provided circuit diagram. Furthermore, we will connect the cathode terminal of the LED and one leg terminal of the push-button to the series short terminal of the breadboard. After that, this short terminal will be connected to the Raspberry Pi's ground pin. GPIO pin 4 on the Raspberry Pi will be linked to the LED's anode terminal. We'll connect the push button's remaining leg terminal to Raspberry Pi 4 GPIO pin 17. Finally, we'll use the Raspberry Pi 4's button to control the LED by writing simple Python code.

Components Used:

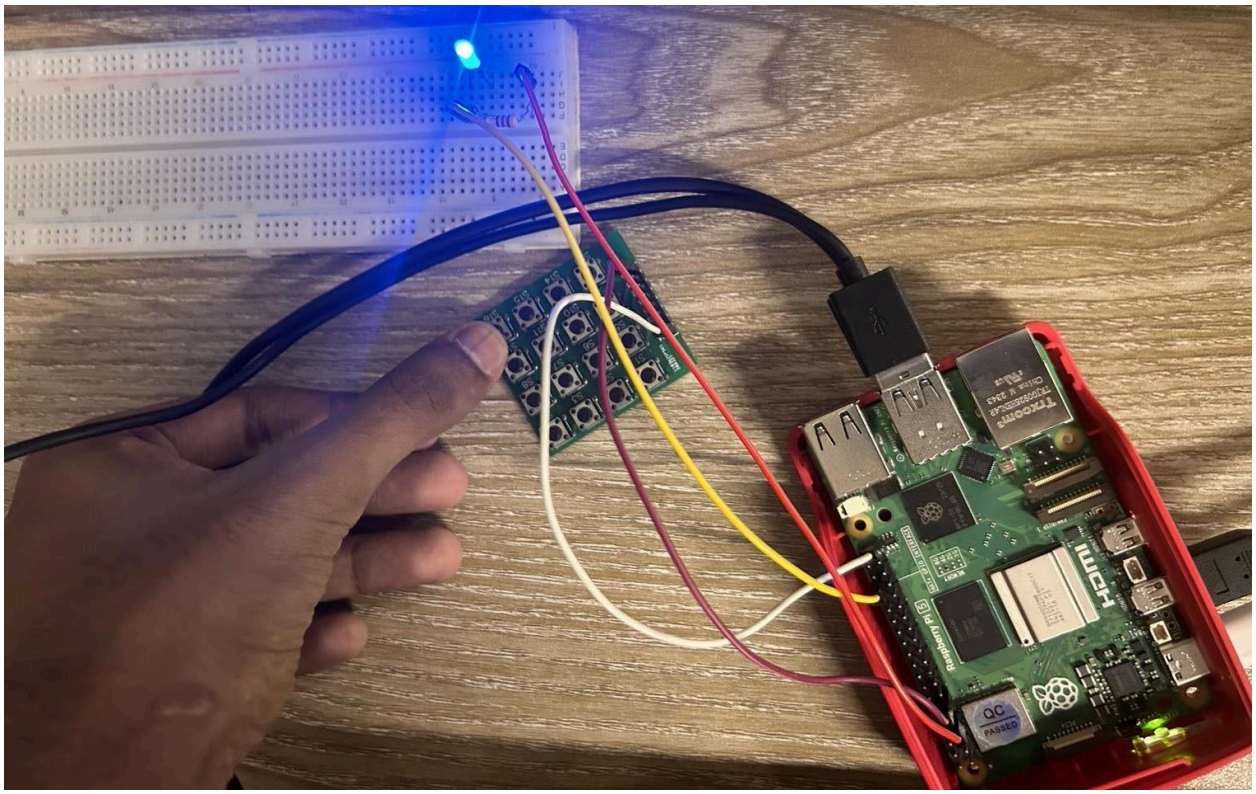
Raspberry Pi 4, Jumper Wires, LED,Push Buttons, Breadboard,A resistor of 220 ohms, SD Card, Mouse, Keyboard, VGA Cable.

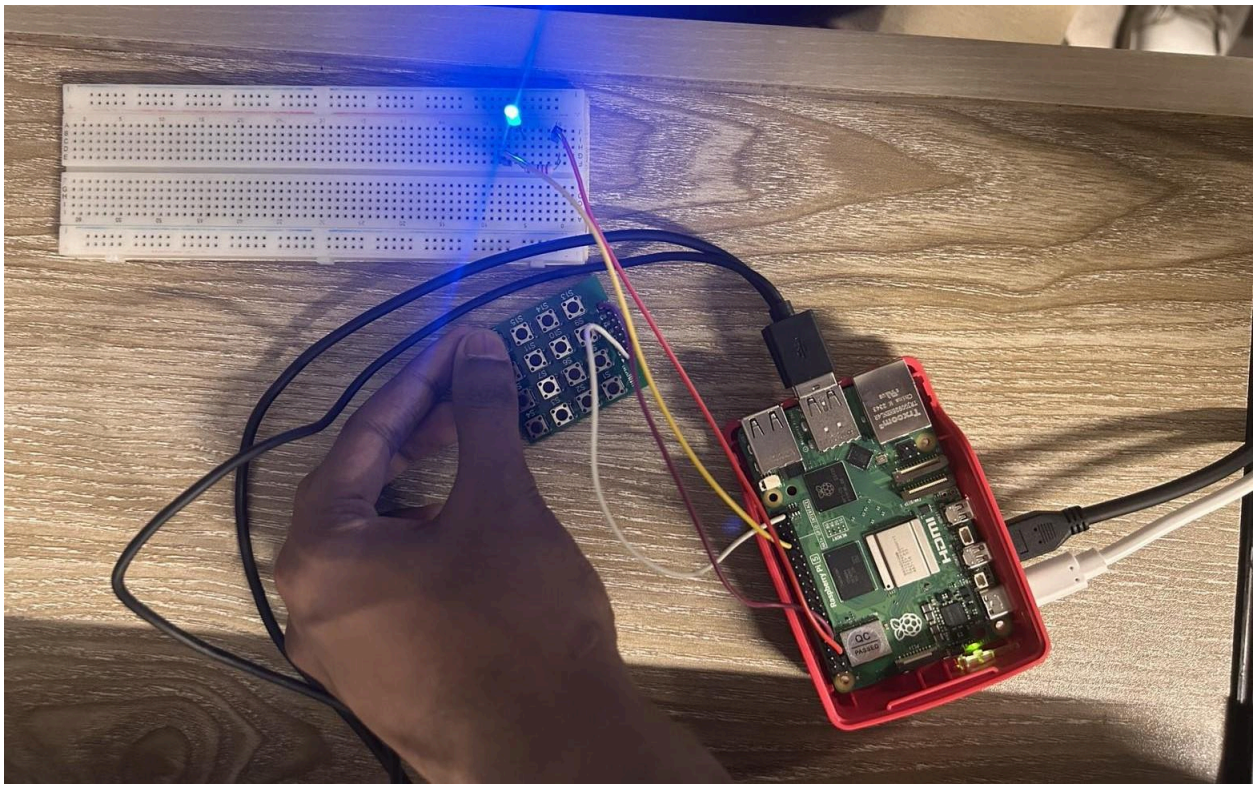
Circuit Diagram:

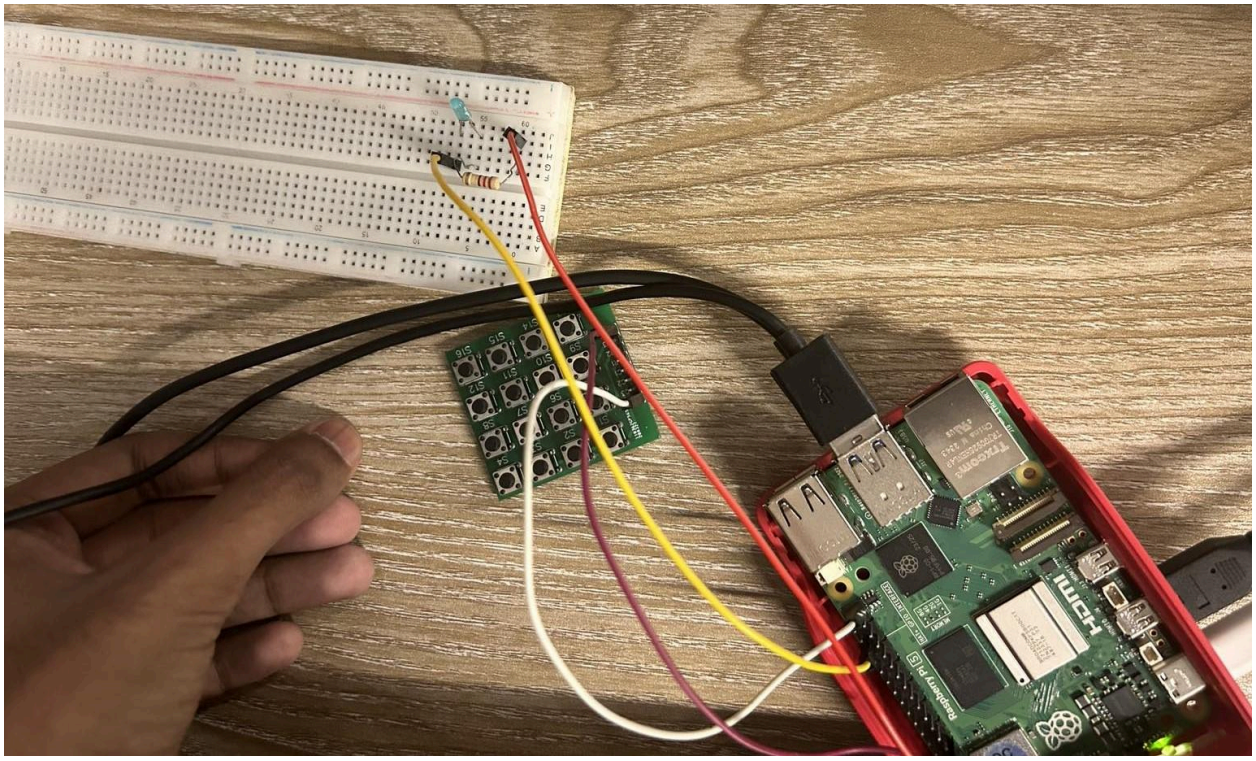
NB: Here the connection of GPIO14 will be on GND (Slight mistake correction)



Circuit Setup:







Code:

```
from gpiozero import LED
#imports LED functions from gpiozero library
from gpiozero import Button
#imports Button functions from gpiozero library
led = LED(4)
#declare the GPIO pin 4 for LED output and store it in led variable
button = Button(17)
#declare the GPIO pin 17 for Button output and store it in button variable
while True:
#initiated an infinite while loop
    button.wait_for_press()
#use the built-in function of the button to wait till press
    led.on()
#turn on the led
    button.wait_for_release()
#use the built-in function of button to wait till release
    led.off()
#turn off the led
```

Discussion / Q&A:

1. Why is there a 220 Ohms resistor in series with the LED?

Answer:

A specific voltage, often between 1.8 and 2.2 volts, is typically needed for LEDs. The LED may be harmed if a higher voltage is applied, such as the 3.3 volts supplied by the Raspberry Pi GPIO pin. A resistor is used to limit the current flowing in order to prevent this. Simply reducing the voltage through the LED.

The voltage across the LED is maintained within a safe operating range by using a 220 ohm resistor, which is about 3.3V times the current. That's why there is a 220 Ohms resistor in series with the LED.

2. Why is the push button connected from a GPIO pin on the RPI to the GND pin of the RPI instead of being connected directly to the LED and the resistor combination?

Answer:

The circuit will not function properly if the push button is connected to the LED and resistor alone, without any other components. Again, when the button is pressed, a connection is made between the GPIO pin and ground, thereby closing the circuit to the GPIO pin. This changing status can be detected by the Raspberry Pi, which can then respond appropriately by turning on or off the LED.

3.What would happen if the series 220 Ohms resistor was replaced with a 1KOhms resistor? What visual change would you see?

Answer:

Compared to a 220 Ohm resistor, a 1K Ohm resistor gives more resistance to current flow.

Consequently, there will be less voltage available for the LED due to a higher voltage across the resistor. It uses little voltage and current, which makes the LED seem dimmer. Extreme situations if the resistor value is too high, the LED may not light at all due to insufficient voltage.