

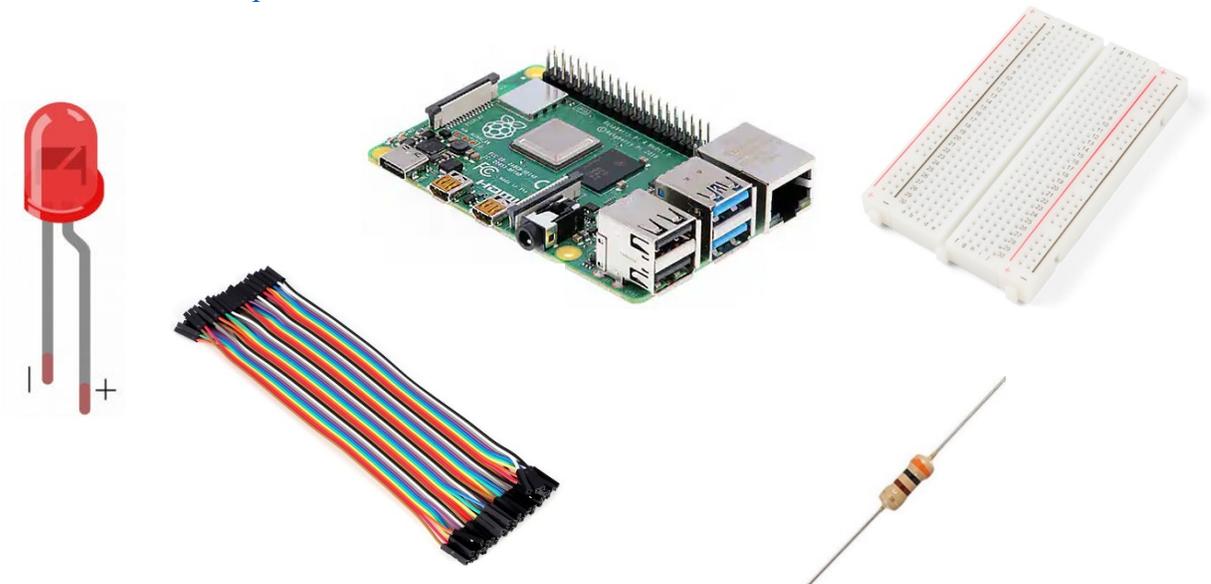
Module 03: Lab Experiment

Controlling IoT LED Using Raspberry Pi and Python

Overview: In this lab experiment, you will learn how to use Raspberry Pi to control a Light Emitting Diode (LED) using a Raspberry Pi's General-Purpose Input Output (GPIO). The GPIO allows you to connect your Raspberry Pi to the outside IoT world through the use of PINs that may be set to input or output, and are controlled through Python code.

Hardware and Software Components:

- [Raspberry Pi Device](#)
- [LED](#)
- [Resistor 330 ohm](#)
- [Breadboard](#)
- [Male/Female Jumper Wires](#)



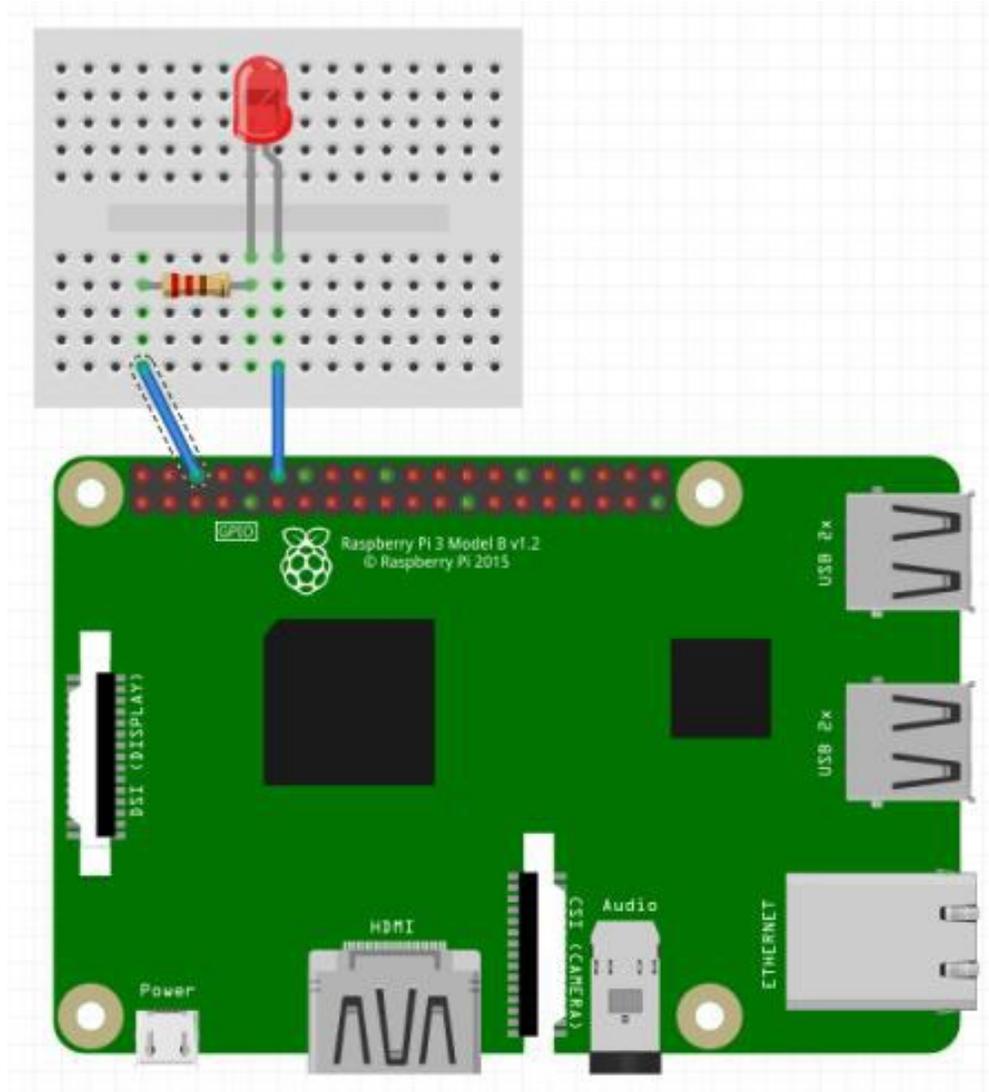
1. Step 1: Connecting the Circuit

Ensure that the Raspberry Pi is turned off until the circuit is connected to avoid accidentally damaging any components.

To build our physical circuit, start by inserting the hardware components into our breadboard. Please refer to the below figure to guide you through the process.

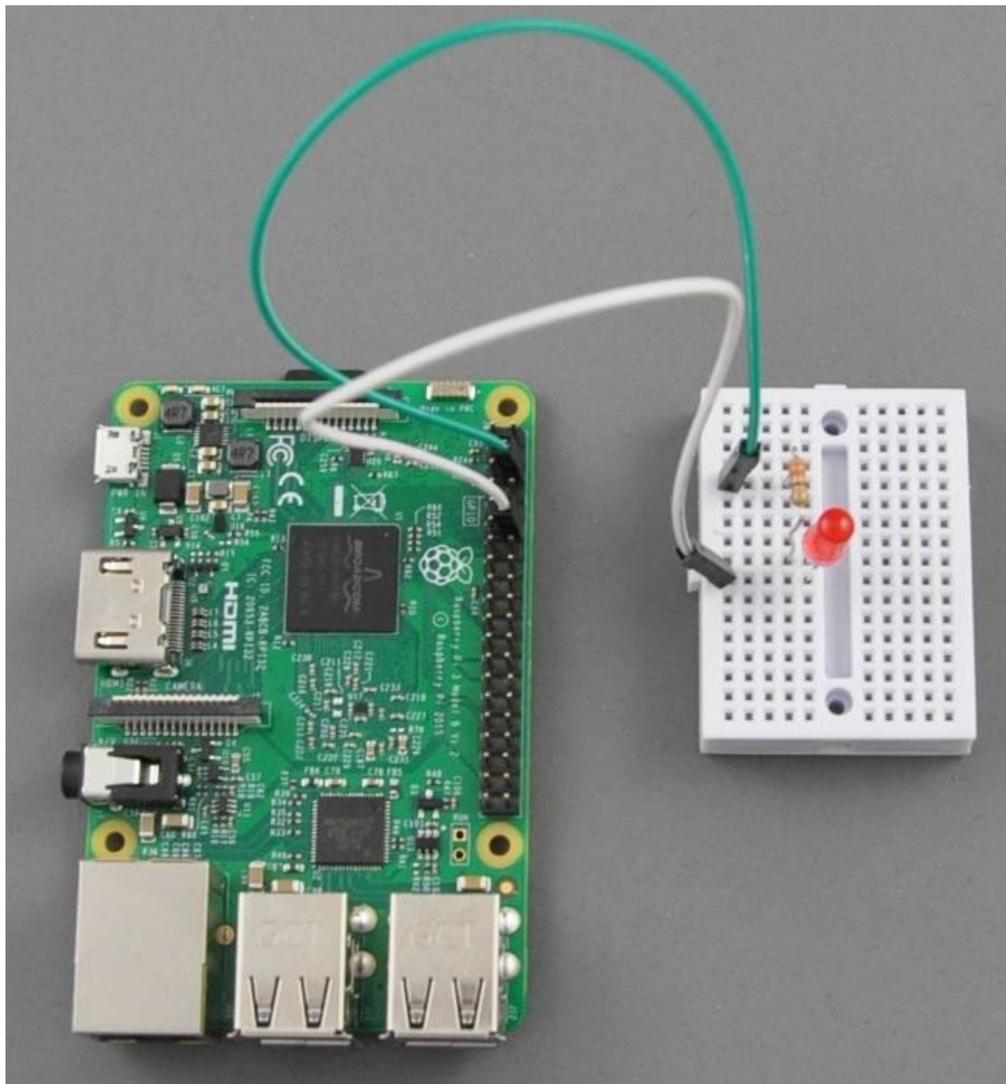
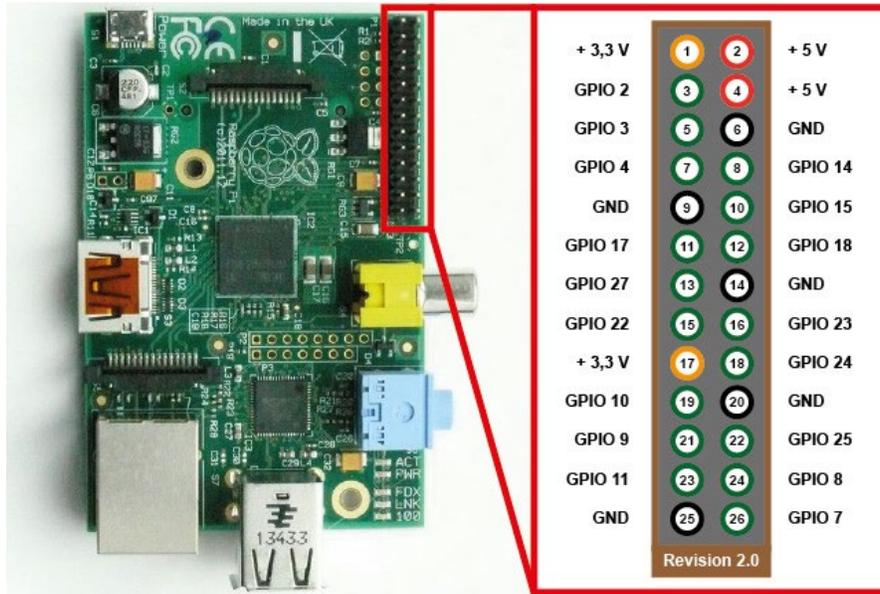
- **LED:** The LED has 2 legs. The longer leg (anode) is always connected to a positive supply. The shorter leg (cathode) is always connected to the ground.

- **Resistor:** You need a resistor to limit the amount of current in the circuit. Without the resistor, the current flow through the LED will be much larger than required, which may damage the circuit.



- Use a jumper wire to connect the GPIO's ground (Pin 6) to the '-' rail (marked in blue) on the breadboard.
- Connect the resistor from the same row on the breadboard to a column on the breadboard.
- Connect the LED with the cathode (shorter leg) in the same row as the resistor. Insert the anode (long leg) in the adjacent row.
- Use another jumper wire to connect the Pin 7 (GPIO4) in the same row as the anode (long leg) of the LED.

This completes our circuit. Now, you can turn on your Raspberry Pi.



2. Step 2: Writing the Python Code

Now, it is time to write your Python code that will turn on/off the LED.

Please follow the following steps:

- Open a terminal in your Raspberry Pi.
- Create a new file named 'led.py' by typing this command: `nano led.py`
- Then, write the below code in the `led.py` file. This code will turn your LED on for 10 seconds and then turns it off again.

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(4,GPIO.OUT)
print "LED on"
GPIO.output(4,GPIO.HIGH)
time.sleep(10)
print "LED off"
GPIO.output(21,GPIO.LOW)
```

- Exit the file by clicking on 'CTRL X'.
- Run the program in the terminal using this command: `python led.py`

Now, you will write another short Python program that will make your LED blinks.

- Open a terminal in your Raspberry Pi.
- Create a new file named 'blink.py' by type this command: `nano blink.py`
- Then, write the below code in the `blink.py` file.

```
from gpiozero import LED
from time import sleep
led = LED(4)
while True:
    led.on()
    sleep(1)
    led.off()
    sleep(1)
```

- Exit the file by clicking on 'CTRL X'.
- Run the program in the terminal using this command: `python blink.py`
- If all goes well, you should see the LED blink!

Summary: This lab experiment showed you how you could control a LED using a Raspberry Pi's GPIO and Python Script. This project is a good starting point to develop more great projects for exploring the IoT world!