Internet of Things

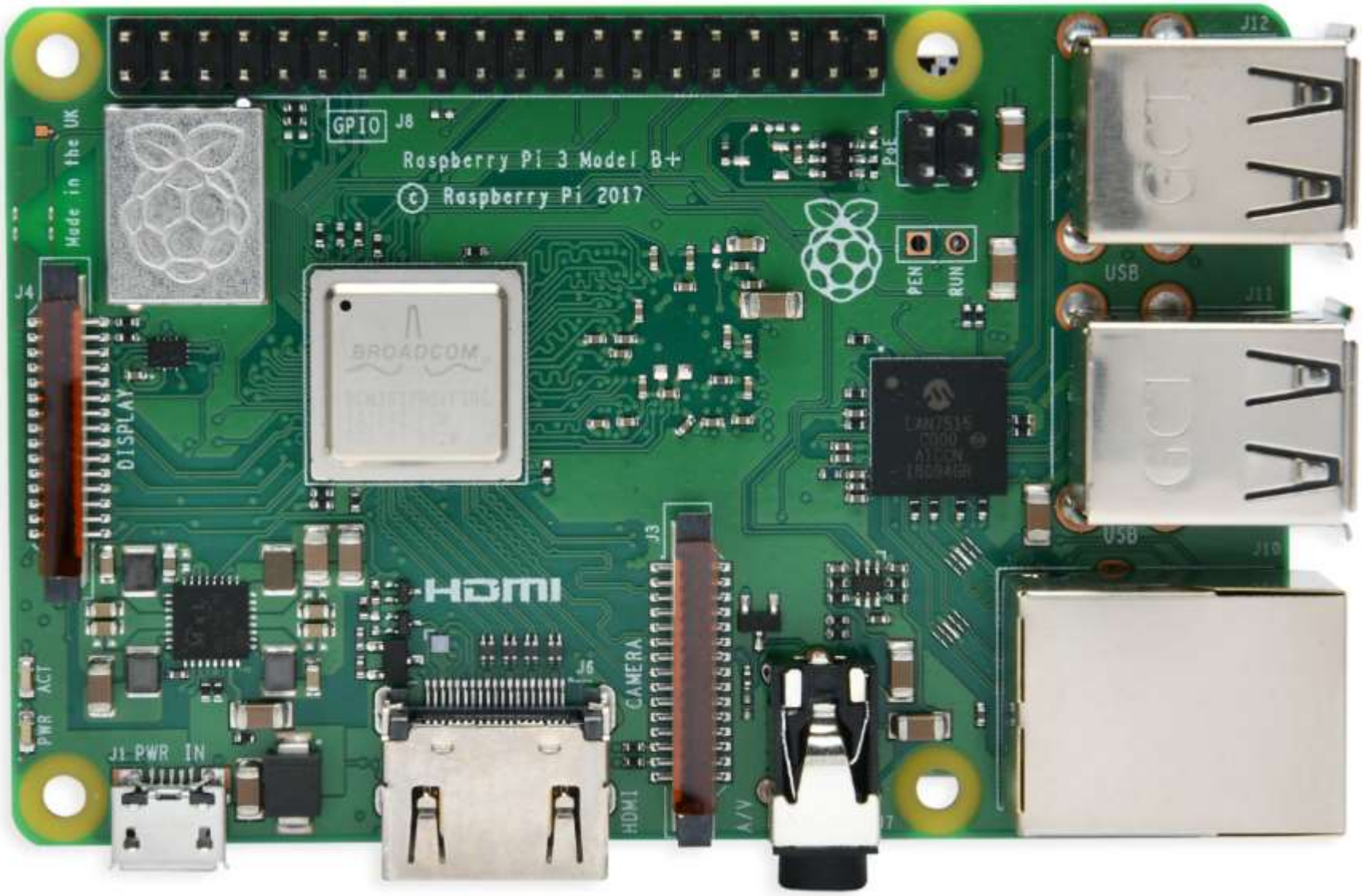
Simple Application on RPi

Abdallah El Ghamry

Raspberry Pi

- The Raspberry Pi is capable of doing all the things you'd expect from a computer.
- Everything from browsing the internet and playing games, to watching movies and listening to music.
- Raspberry Pi is known as a single-board computer, but that doesn't mean it's not powerful.
- Raspberry Pi can do anything a bigger computer can do.
- Over the years, the Raspberry Pi has evolved, increasing its memory, improving its performance, and adding features.

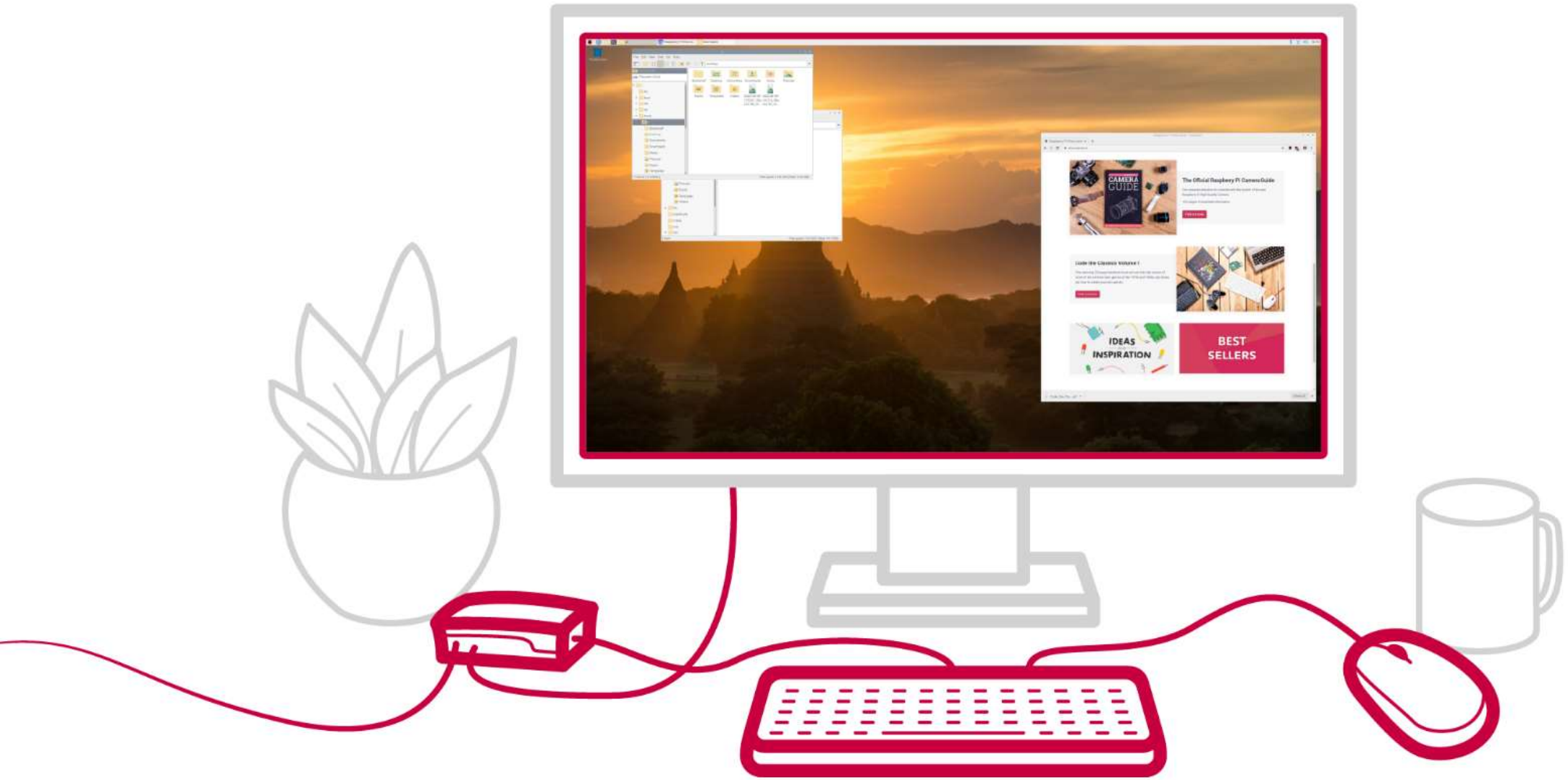
Raspberry Pi 3 Model B+



Raspberry Pi 3 Model B+: Specifications

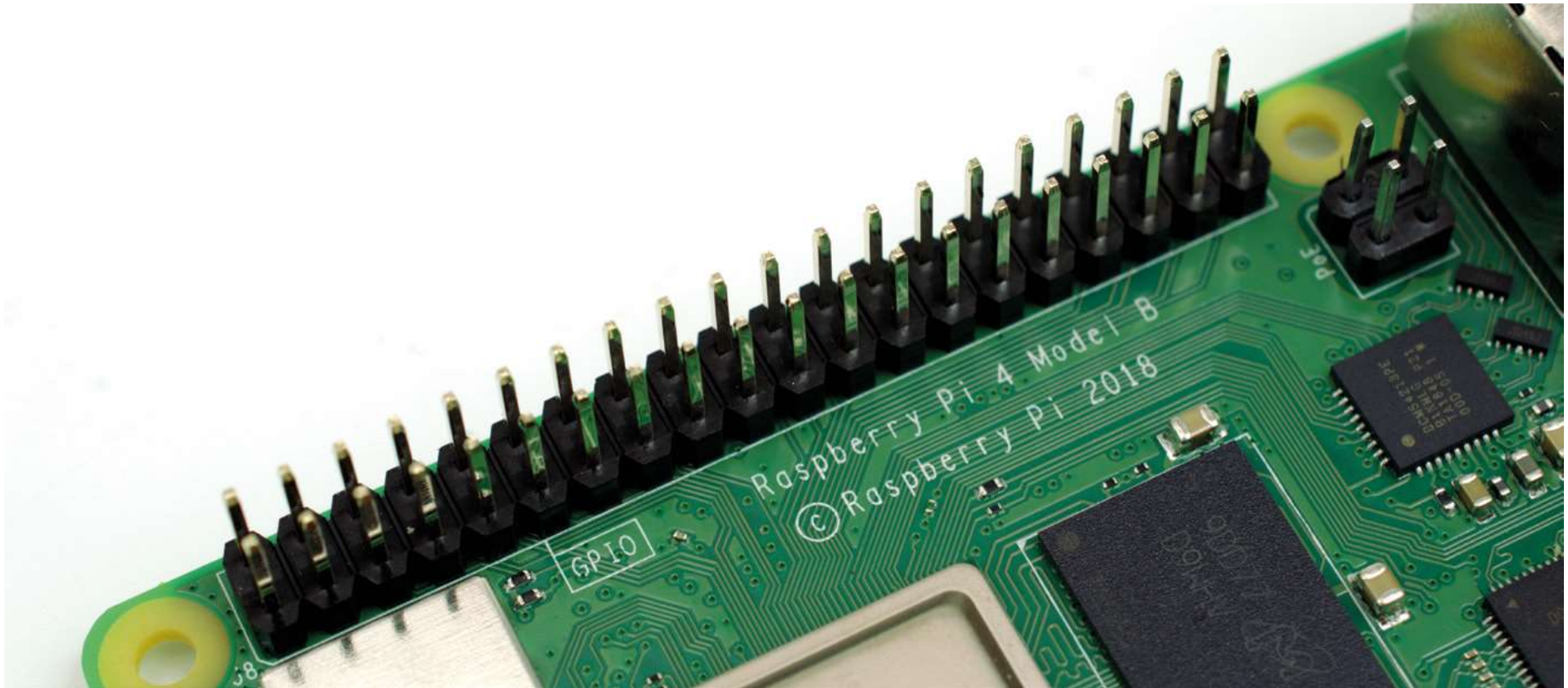
Processor	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz
Memory	1GB LPDDR2 SDRAM
Connectivity	▪ 2.4GHz and 5GHz wireless LAN, Bluetooth 4.2, BLE ▪ Gigabit Ethernet over USB 2.0 ▪ 4 × USB 2.0 ports
Access	Extended 40-pin GPIO header
Video & Sound	▪ 1 × full size HDMI ▪ MIPI DSI display port ▪ MIPI CSI camera port ▪ 4 pole stereo output and composite video port
SD Card Support	Micro SD format for operating system and data storage

Raspberry Pi as PC



Raspberry Pi: GPIO

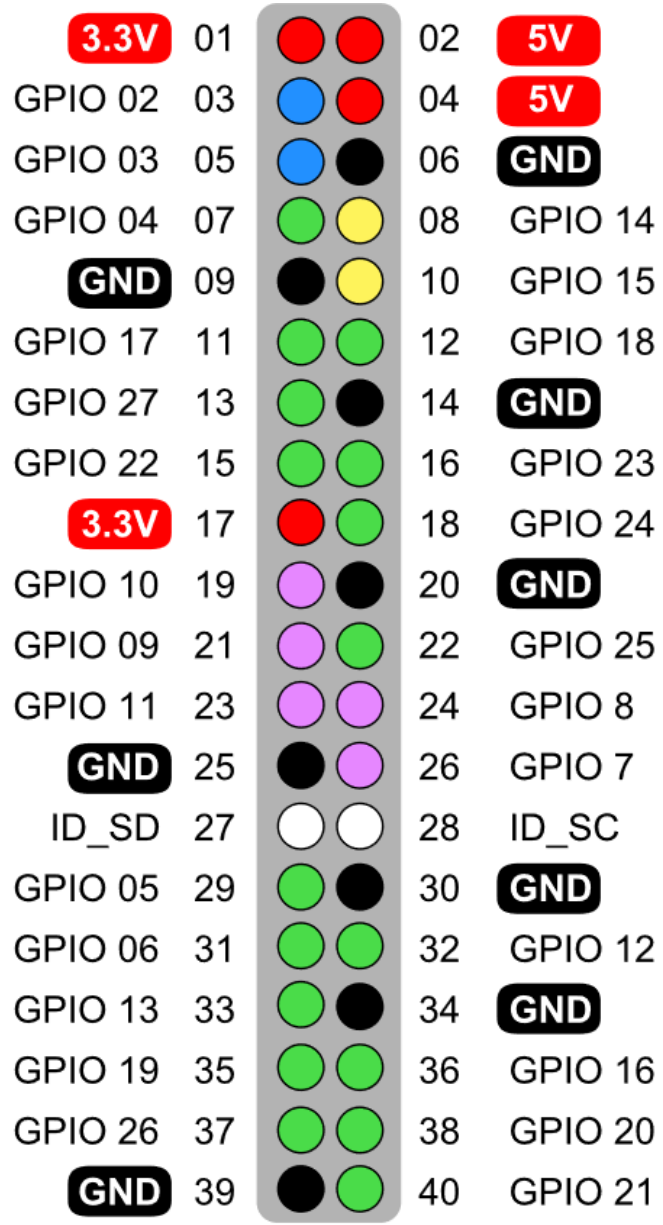
- The GPIO (General-Purpose Input/Output) header is a feature of the Raspberry Pi used to talk to additional hardware such as LEDs and buttons and sensors.



Raspberry Pi: GPIO

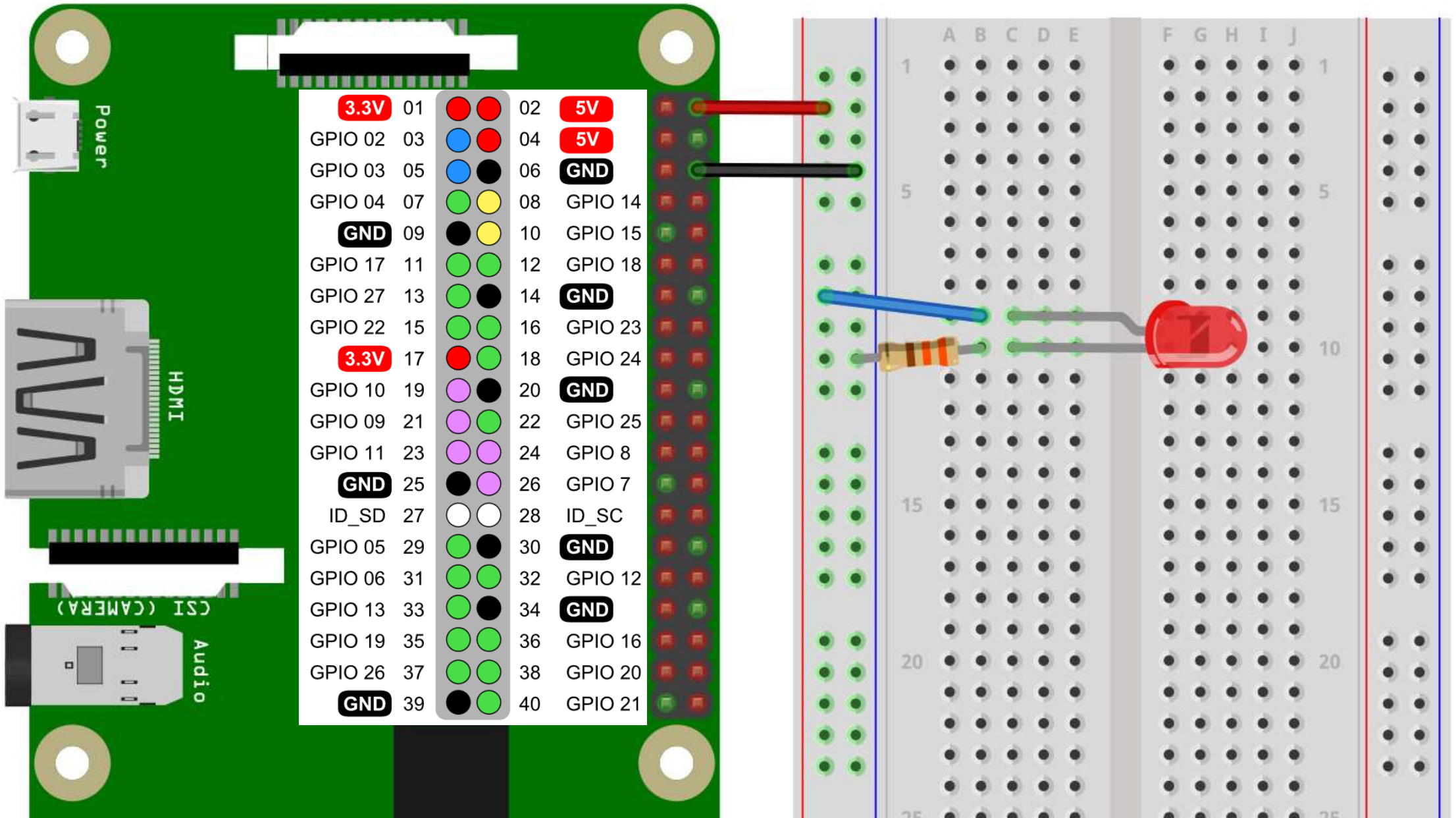
Alternate Function							Alternate Function
	3.3V PWR	1		2	5V PWR		
I2C1 SDA	GPIO 2	3		4	5V PWR		
I2C1 SCL	GPIO 3	5		6	GND		
	GPIO 4	7		8	UART0 TX		
	GND	9		10	UART0 RX		
	GPIO 17	11		12	GPIO 18		
	GPIO 27	13		14	GND		
	GPIO 22	15		16	GPIO 23		
	3.3V PWR	17		18	GPIO 24		
SPI0 MOSI	GPIO 10	19		20	GND		
SPI0 MISO	GPIO 9	21		22	GPIO 25		
SPI0 SCLK	GPIO 11	23		24	GPIO 8		SPI0 CS0
	GND	25		26	GPIO 7		SPI0 CS1
	Reserved	27		28	Reserved		
	GPIO 5	29		30	GND		
	GPIO 6	31		32	GPIO 12		
	GPIO 13	33		34	GND		
SPI1 MISO	GPIO 19	35		36	GPIO 16		SPI1 CS0
	GPIO 26	37		38	GPIO 20		SPI1 MOSI
	GND	39		40	GPIO 21		SPI1 SCLK

Raspberry Pi: GPIO



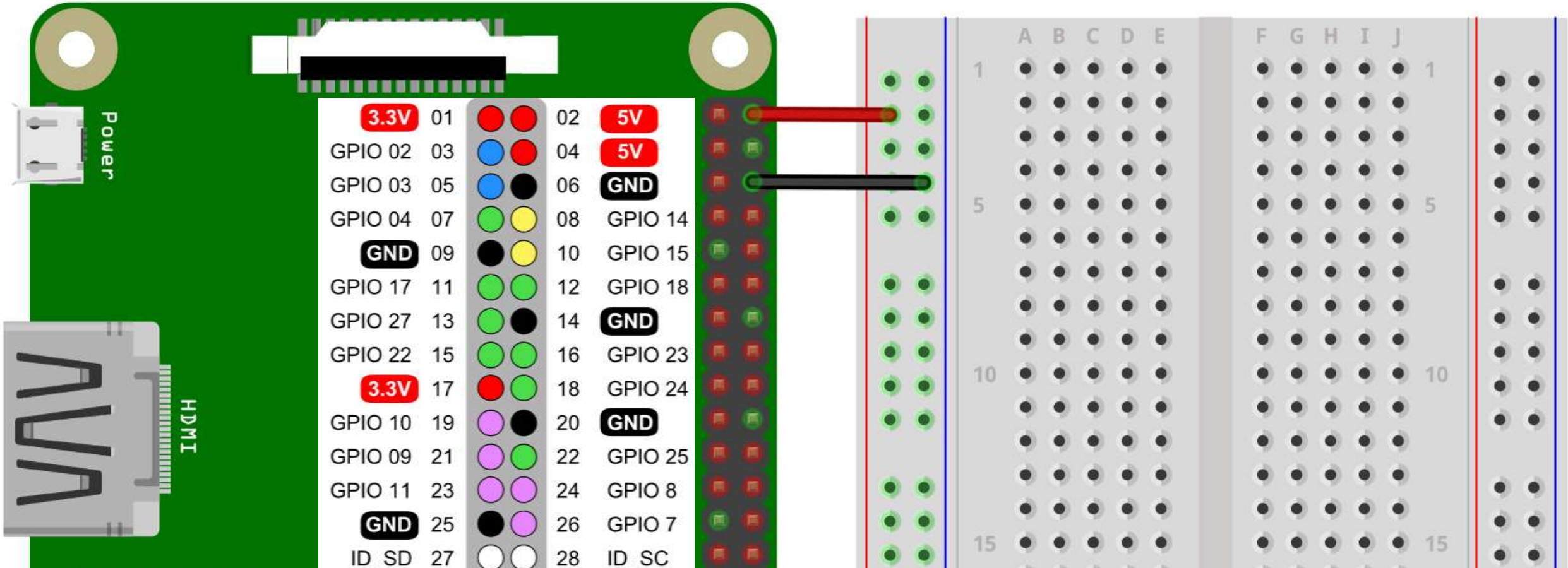
Raspberry Pi: GPIO Template

Turning on an LED



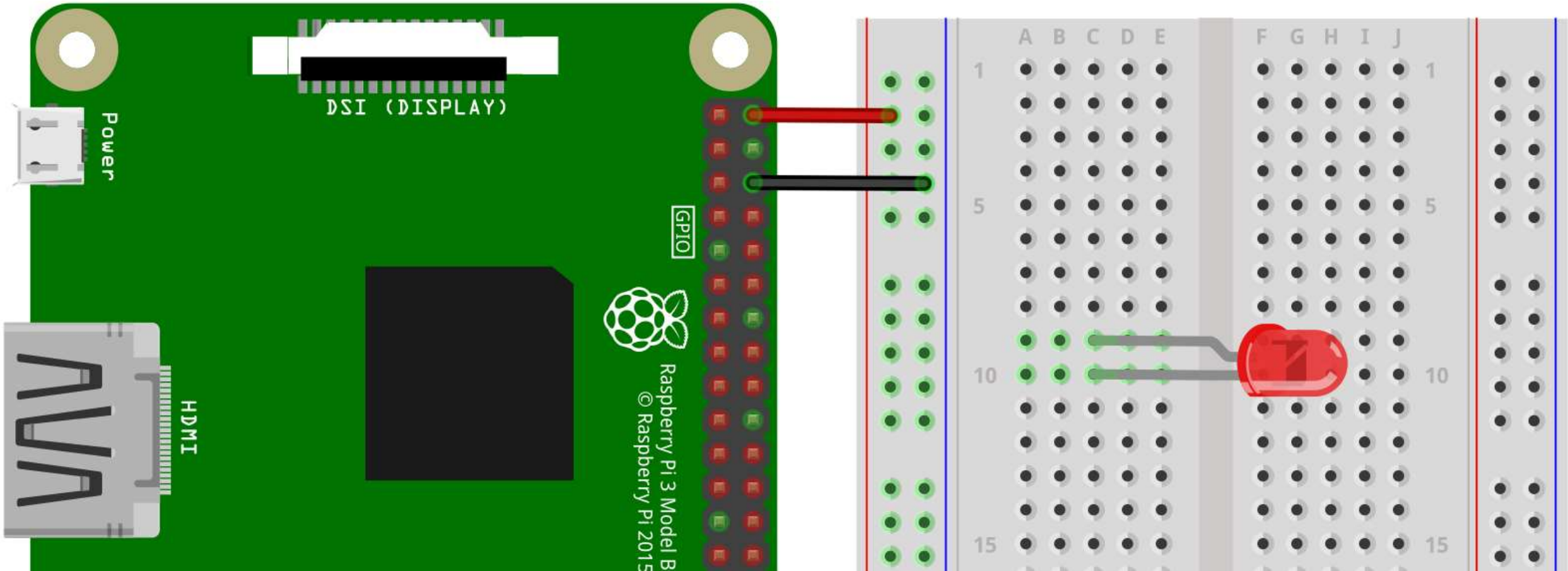
Turning on an LED: Steps

1. Connect breadboard power (+) and ground (-) rails to Arduino 5V and ground (GND), respectively.



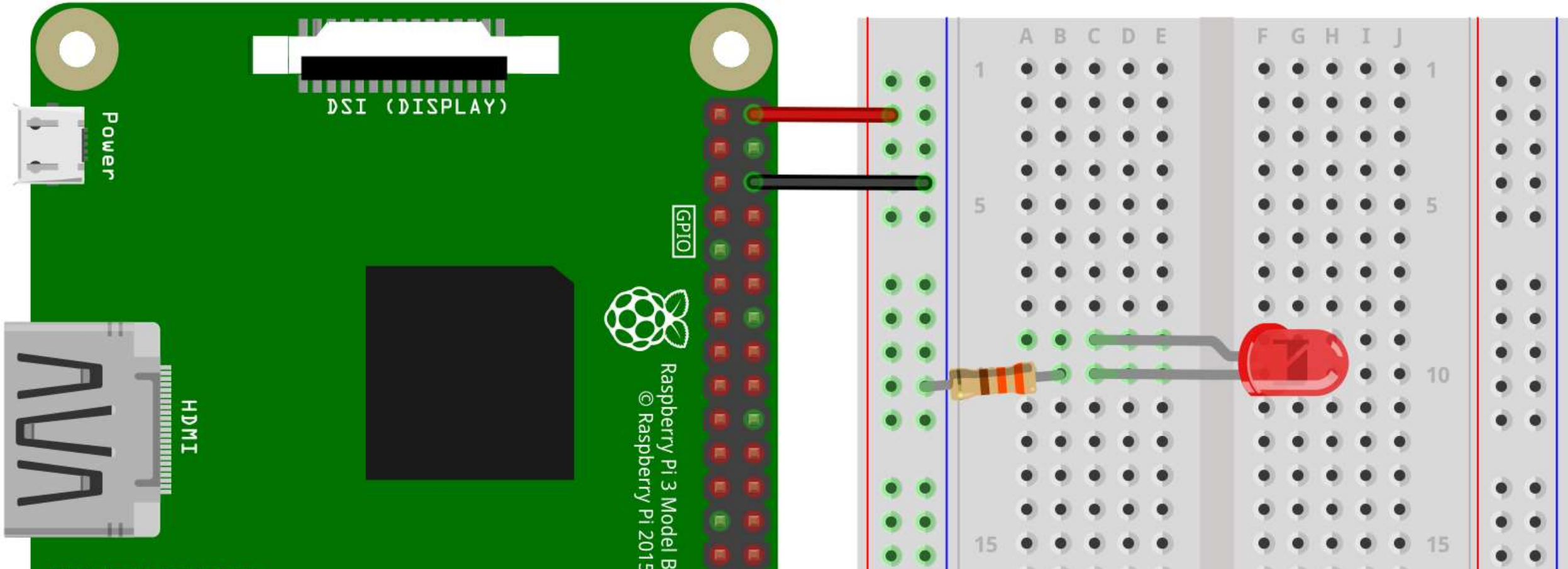
Turning on an LED: Steps

2. Plug the LED into two different breadboard rows.



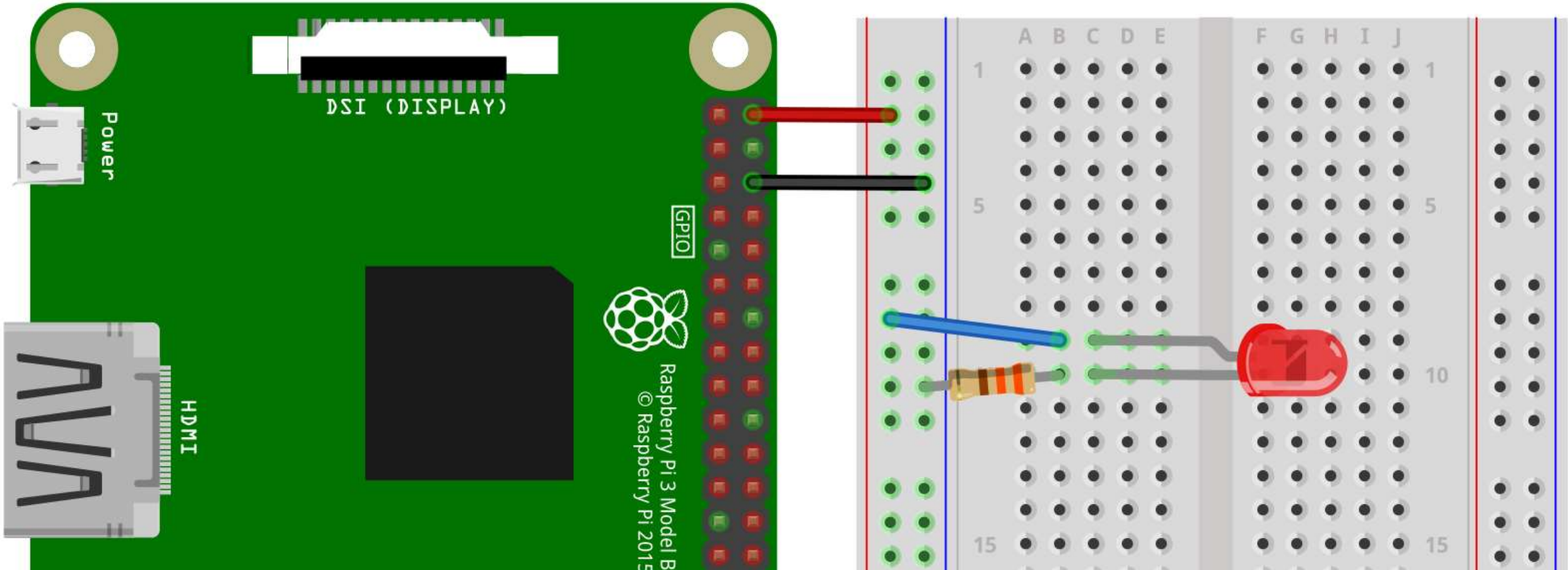
Turning on an LED: Steps

3. The cathode (shorter leg) connects to one leg of a resistor of 330Ω, and the other resistor leg to the ground.

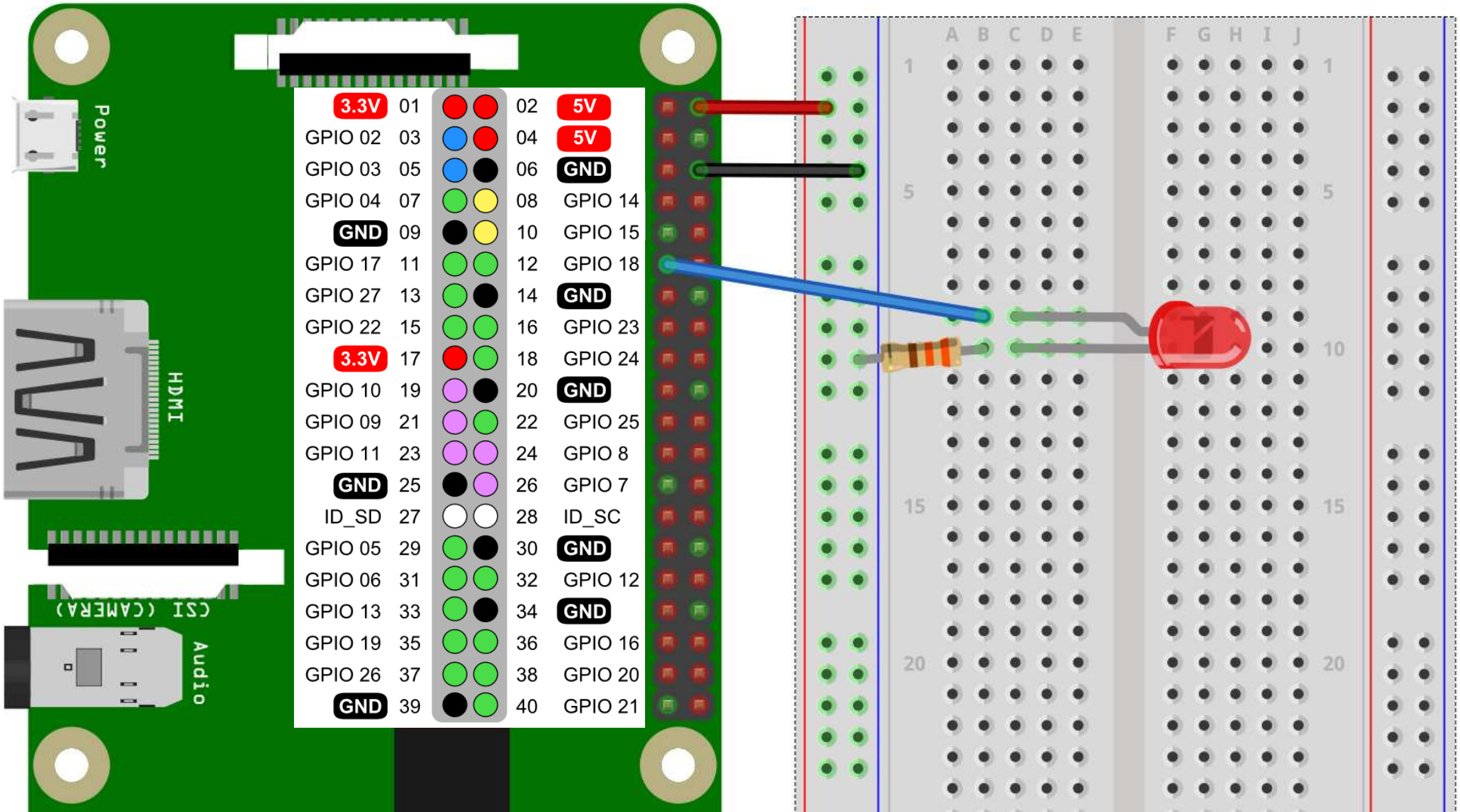


Turning on an LED: Steps

4. Wire up the LED anode (longer leg) to the power.

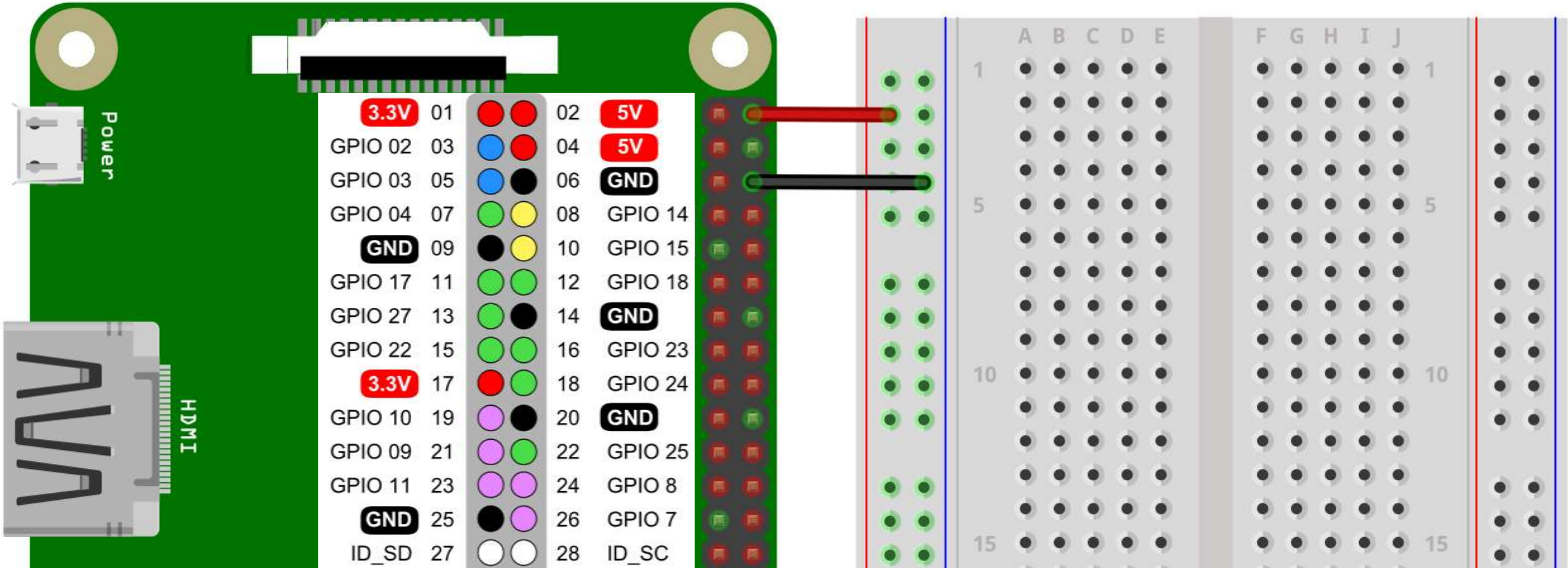


Your First Raspberry Pi Project: Blinking an LED



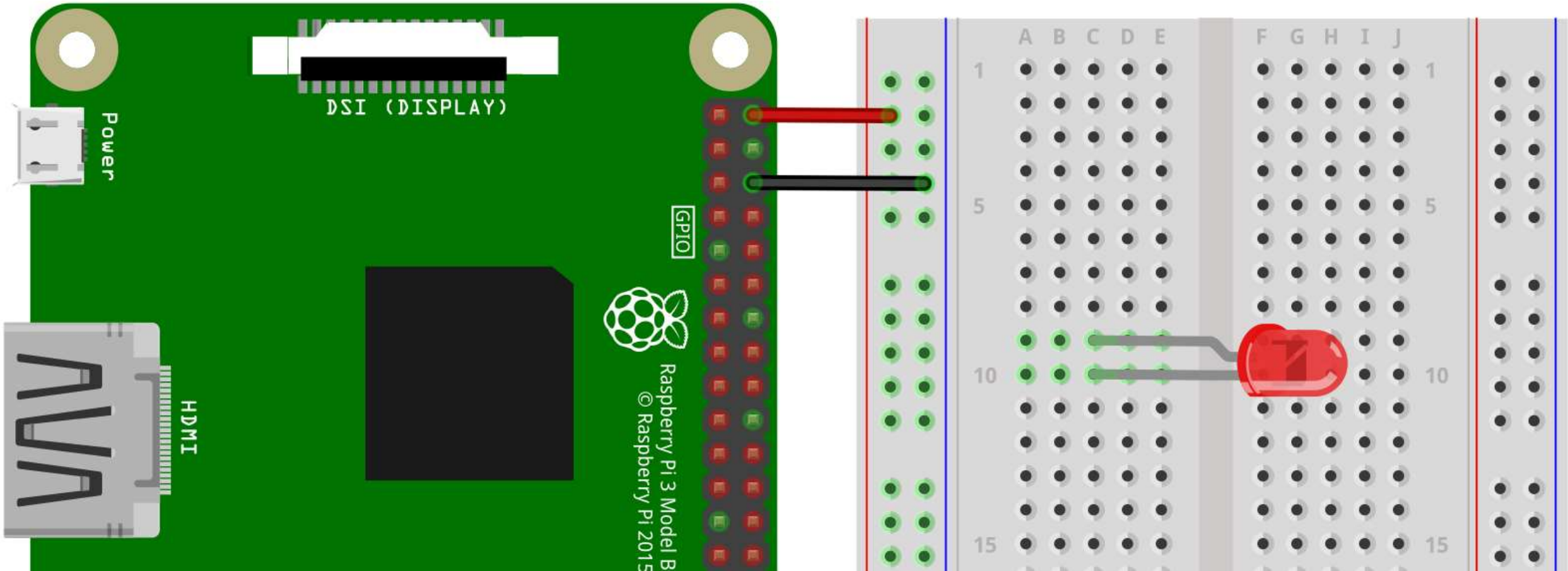
Your First Raspberry Pi Project: Steps

1. Connect breadboard power (+) and ground (-) rails to Arduino 5V and ground (GND), respectively.



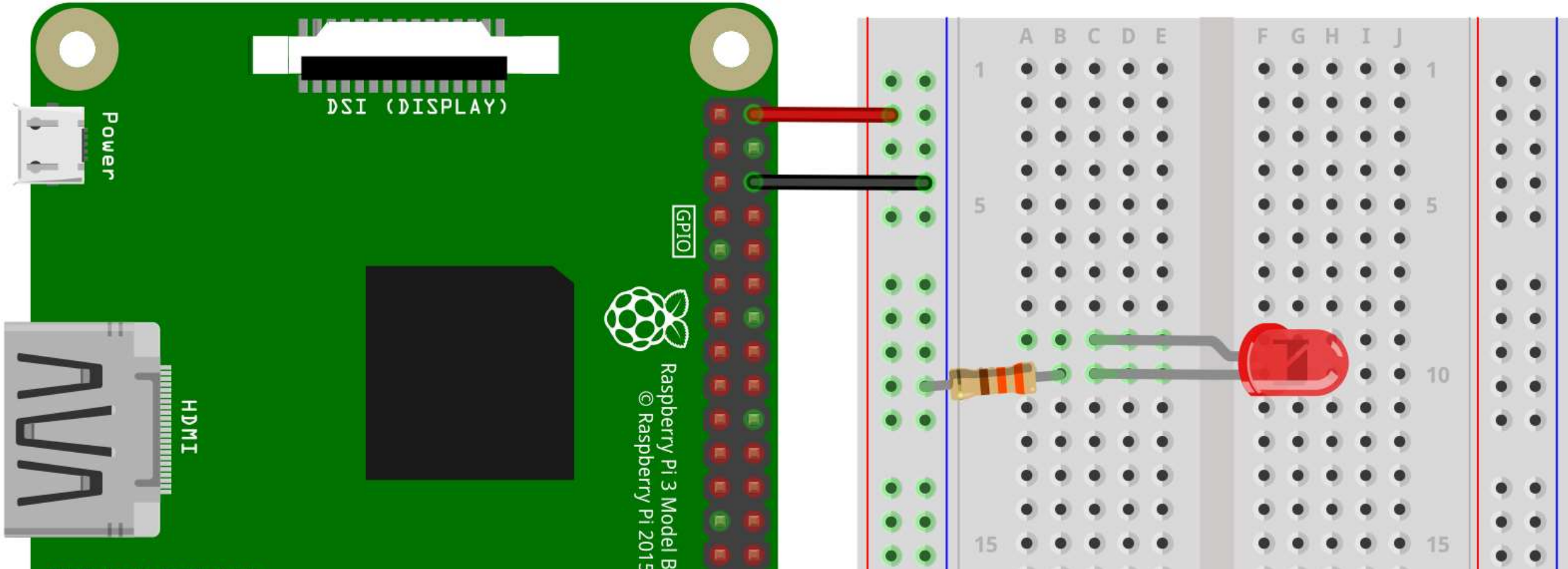
Your First Raspberry Pi Project: Steps

2. Plug the LED into two different breadboard rows.



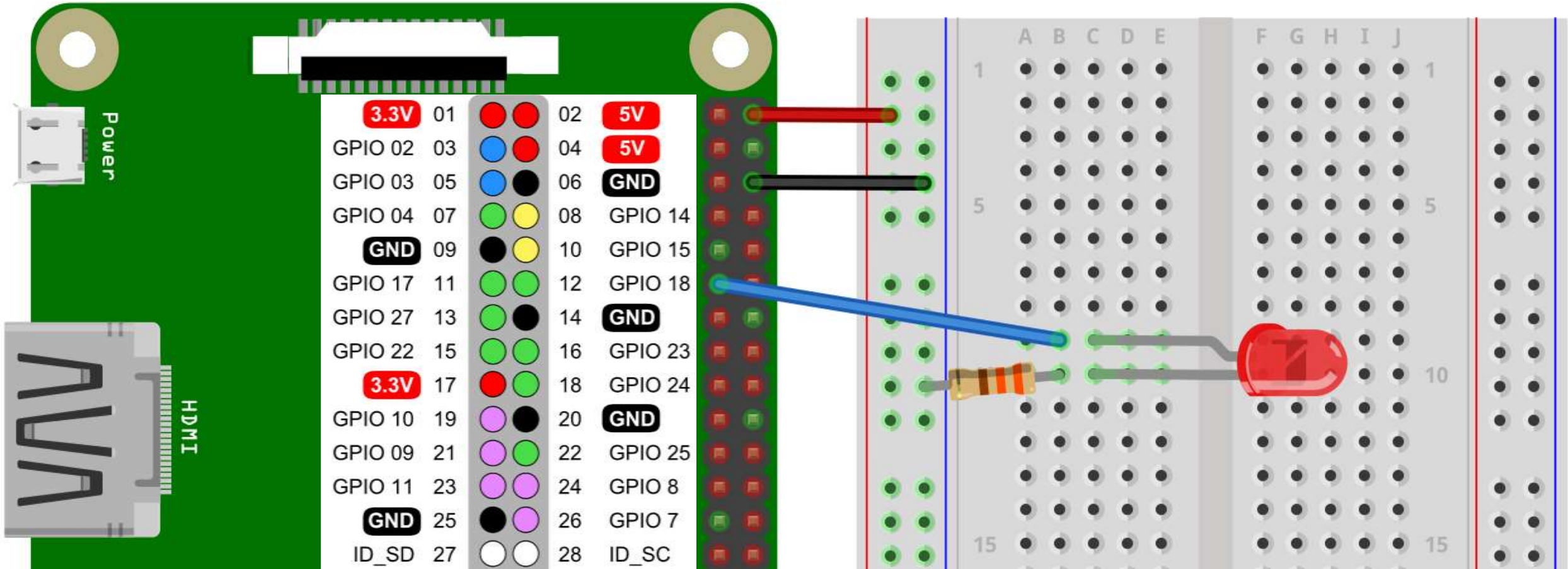
Your First Raspberry Pi Project: Steps

3. The cathode (shorter leg) connects to one leg of a resistor of 330Ω, and the other resistor leg to the ground.



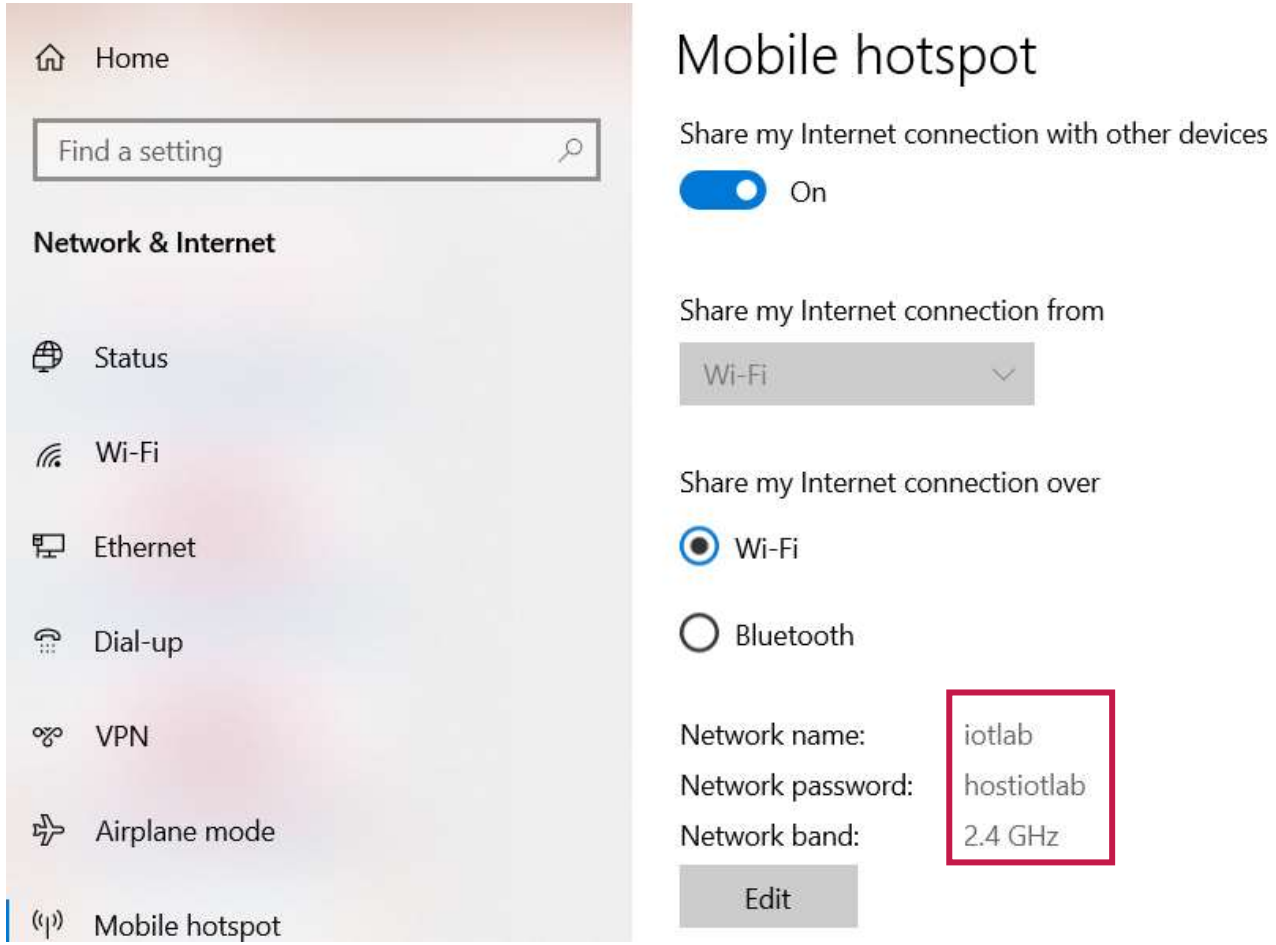
Your First Raspberry Pi Project: Steps

4. Wire up the LED anode (longer leg) to Arduino pin 11 (GPIO 17).



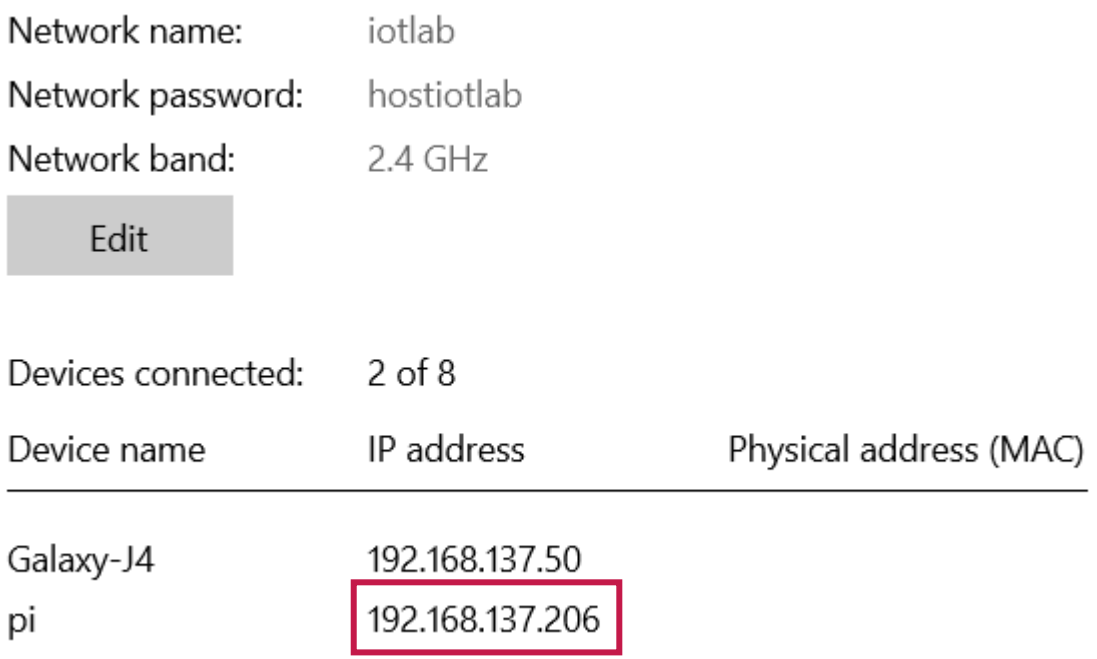
Your First Raspberry Pi Project: Remote Access

- Open Mobile hotspot, and make sure your Wi-Fi info is correct.
- Make sure to turn on the hotspot.



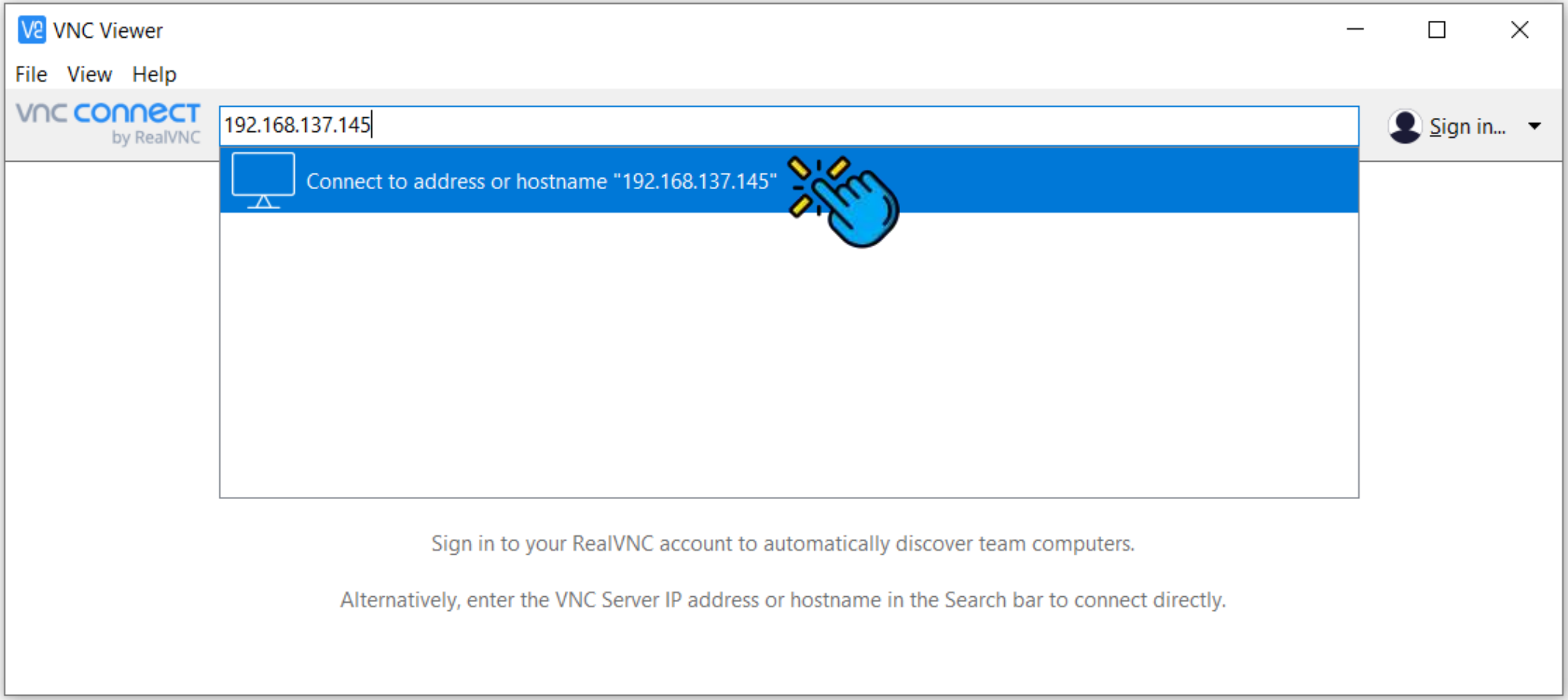
Your First Raspberry Pi Project: Remote Access

- After connecting your Raspberry Pi to power, it will be connected to your Wi-Fi automatically and have an IP address.
- Open Mobile hotspot, and copy that IP address.



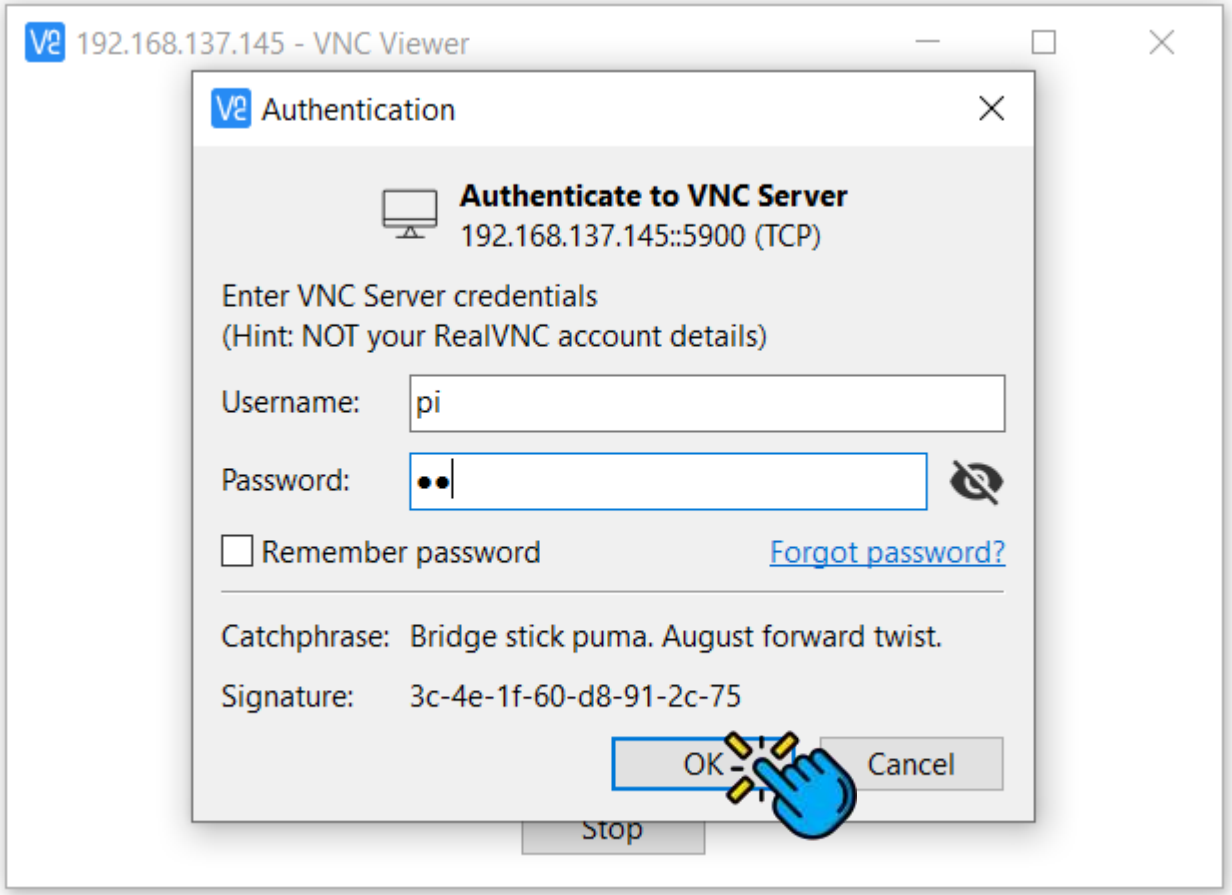
Your First Raspberry Pi Project: Remote Access

- Open VNC Viewer, and Enter the IP address of your RPi.



Your First Raspberry Pi Project: Remote Access

- Enter your username and password, and click Ok.



Your First Raspberry Pi Project: Code

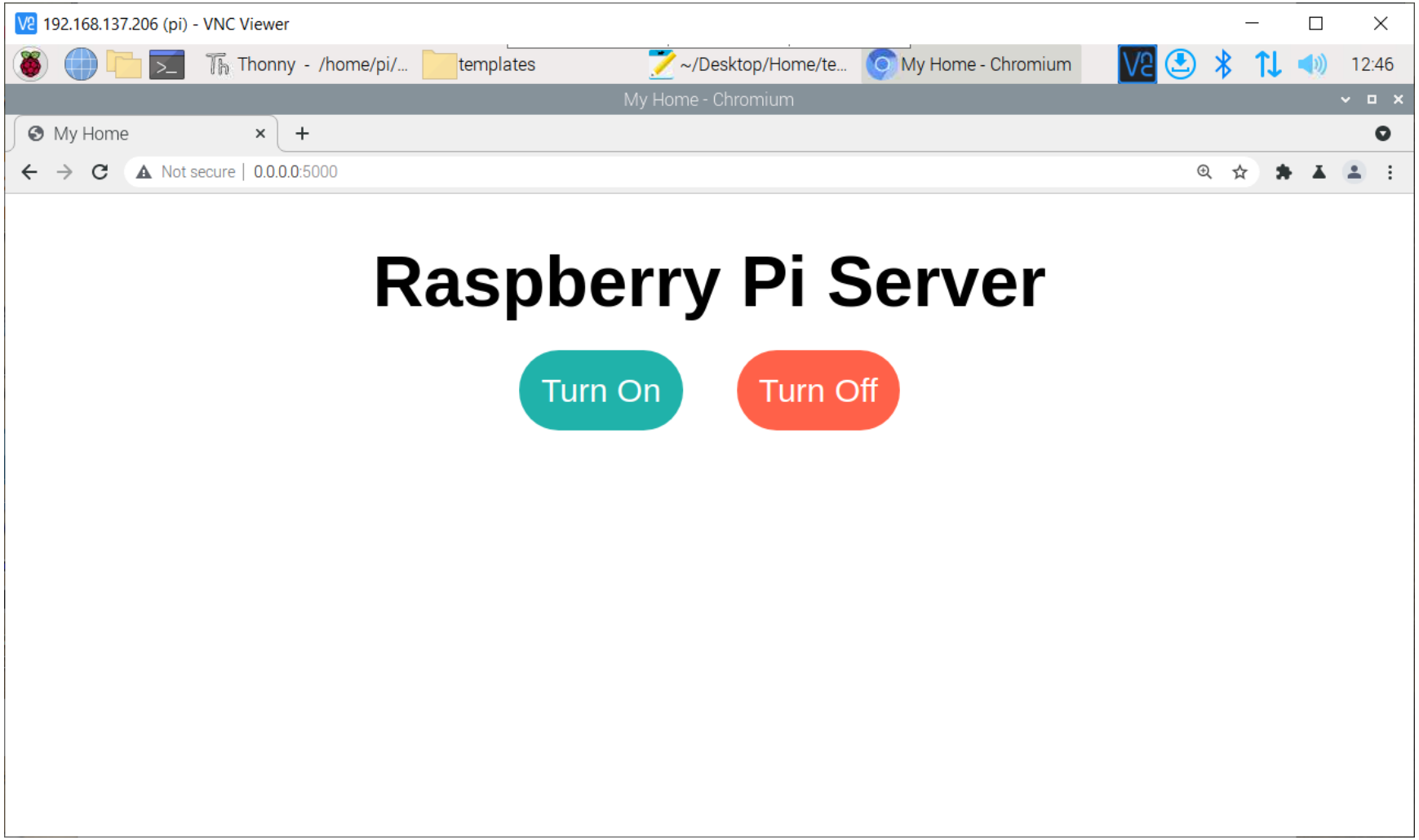
```
from RPi import GPIO                    # Import GPIO
from time import sleep                  # Import sleep function

pin = 11                                # Set pin number
GPIO.setmode(GPIO.BOARD)                # Use board pin numbering
GPIO.setup(pin, GPIO.OUT)               # Set pin 11 as output

while True:
    GPIO.output(pin, 1)                 # Turn the LED on
    sleep(1)                            # Wait for a second

    GPIO.output(pin, 0)                 # Turn the LED off
    sleep(1)                            # Wait for a second
```

Your First Raspberry Pi Project: Alternative Code

```
from RPi import GPIO                    # Import GPIO
from time import sleep                  # Import sleep function

pin = 17                                # Set pin number (GPIO 17)
GPIO.setmode(GPIO.BCM)                  # Use GPIO pin numbering
GPIO.setup(pin, GPIO.OUT)               # Set GPIO 17 as output

while True:
    GPIO.output(pin, 1)                 # Turn the LED on
    sleep(1)                            # Wait for a second

    GPIO.output(pin, 0)                 # Turn the LED off
    sleep(1)                            # Wait for a second
```

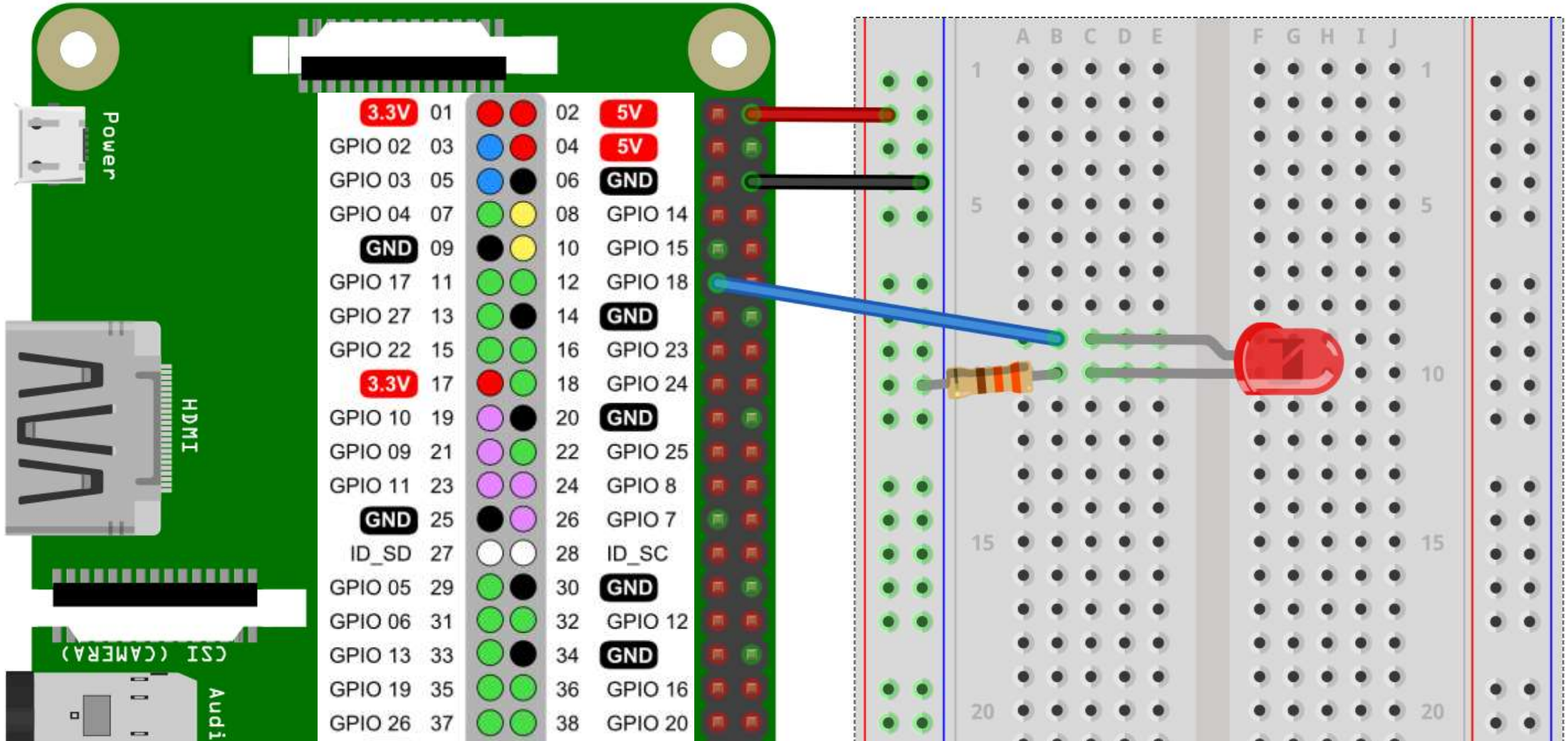
Smart Home

- We want to build a simple app to turn on/off an LED via a webpage.



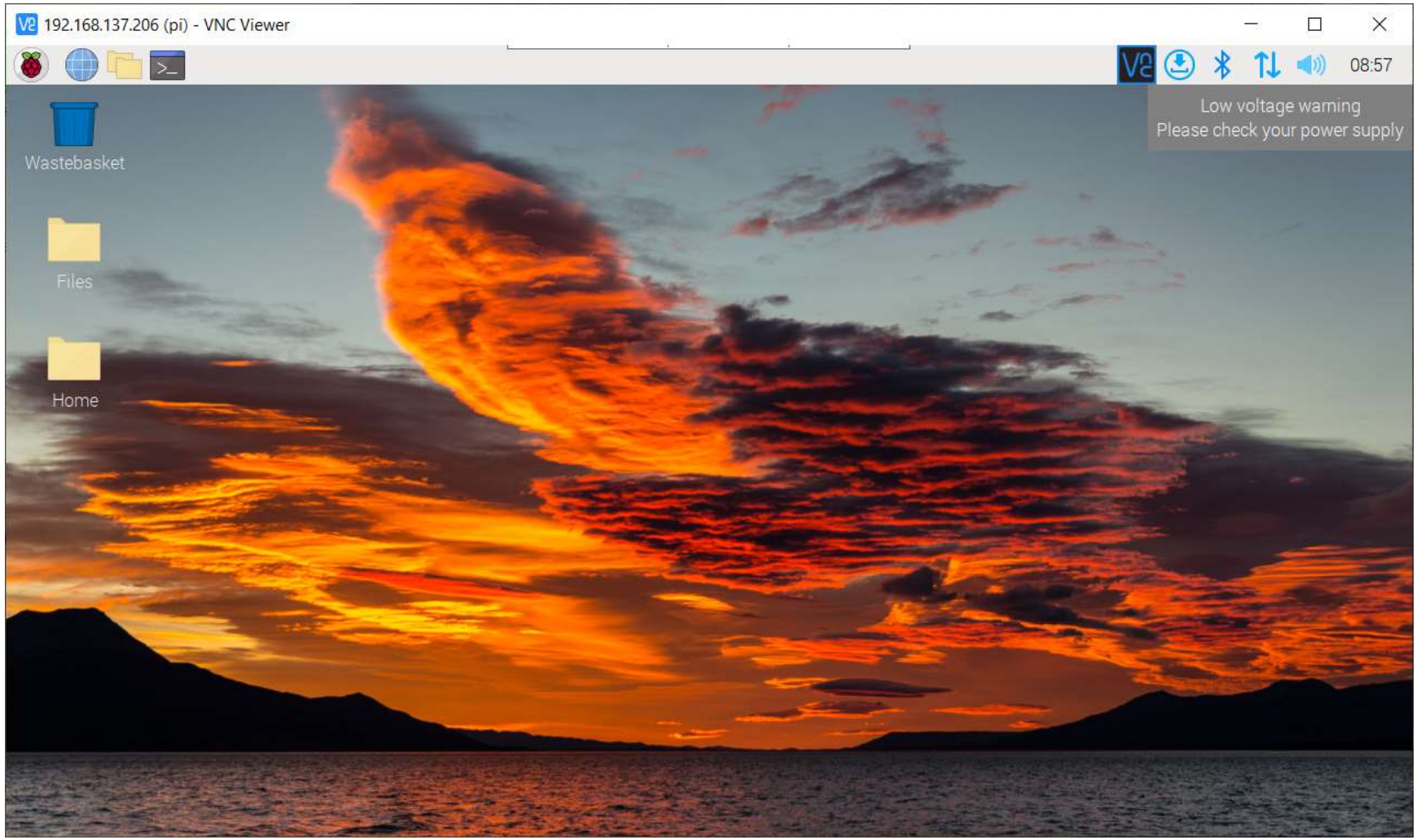
Smart Home: Circuit

- We will use the same connections.



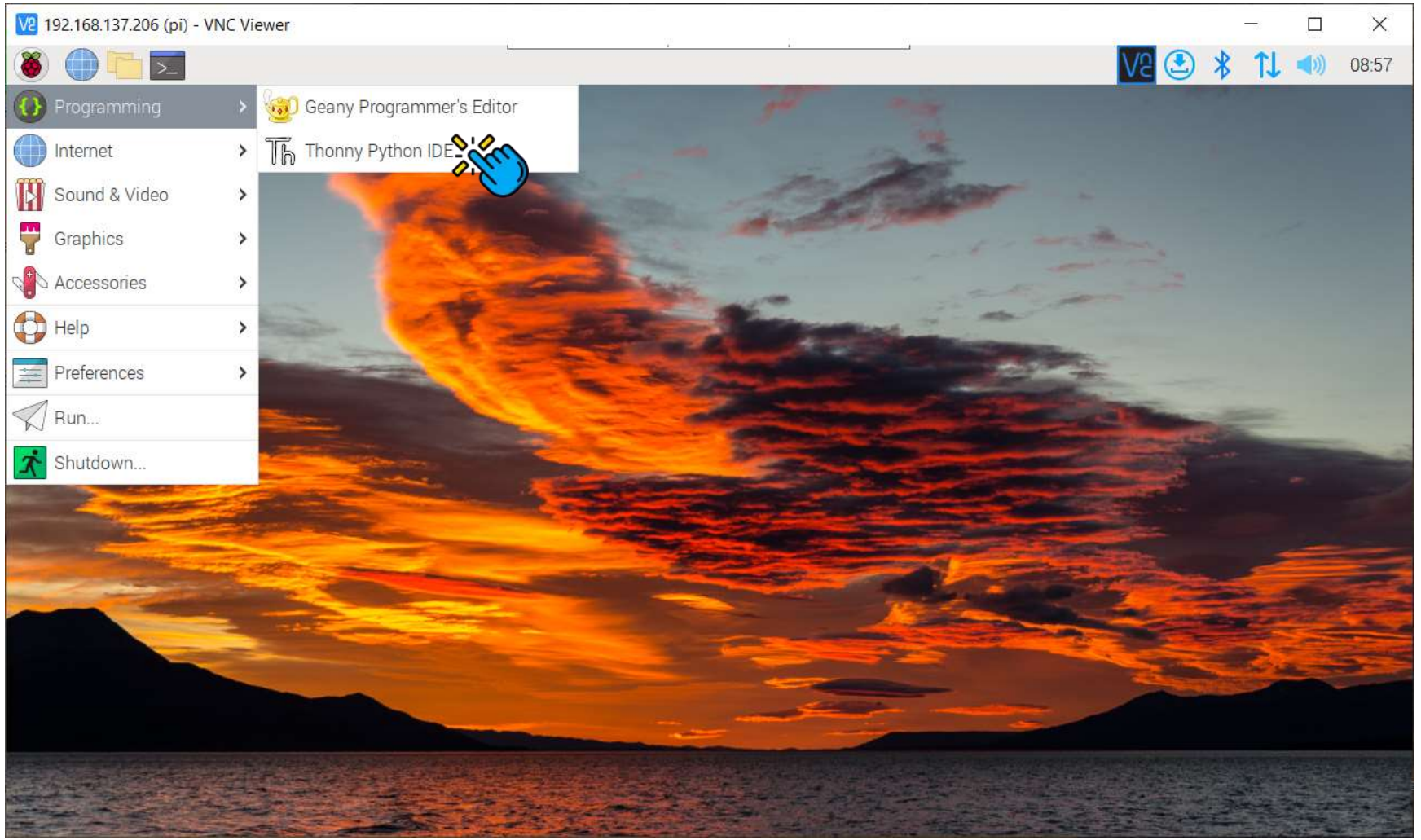
Smart Home: Project Files

- Create a new folder named Home.



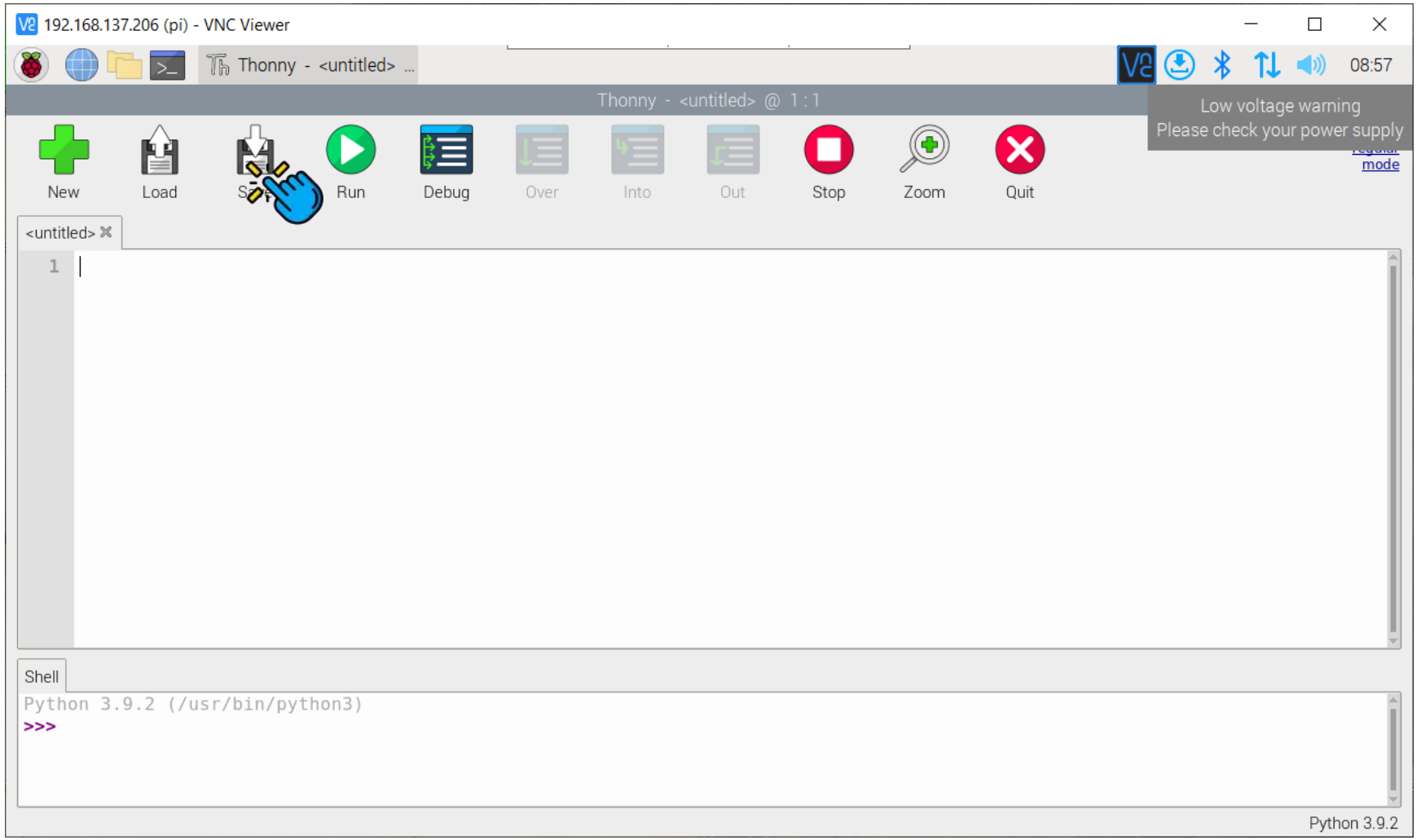
Smart Home: Project Files

- Open Thonny Python IDE to write some python code.



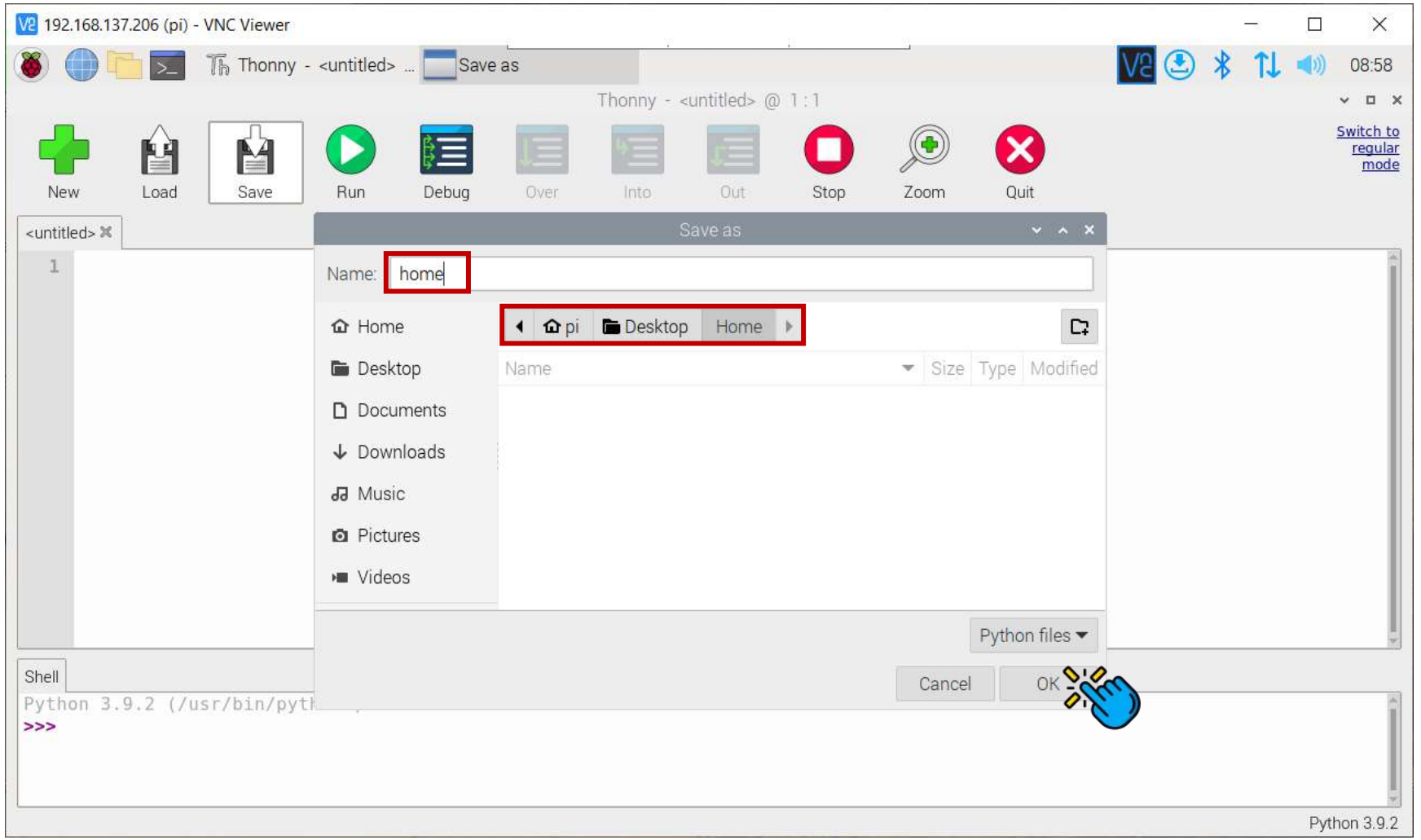
Smart Home: Project Files

- Click Save.



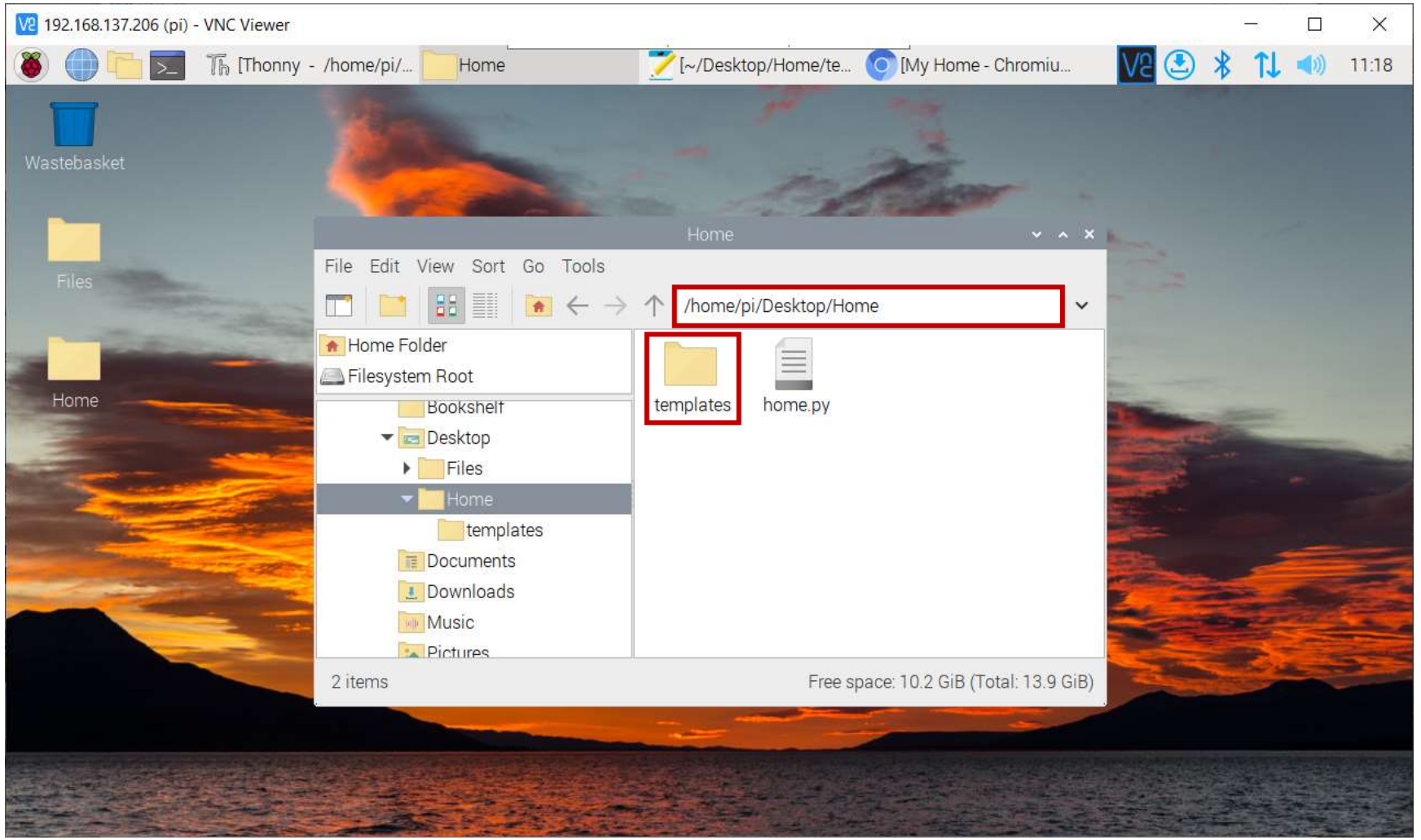
Smart Home: Project Files

- Save your file as `home.py` in the Home folder.



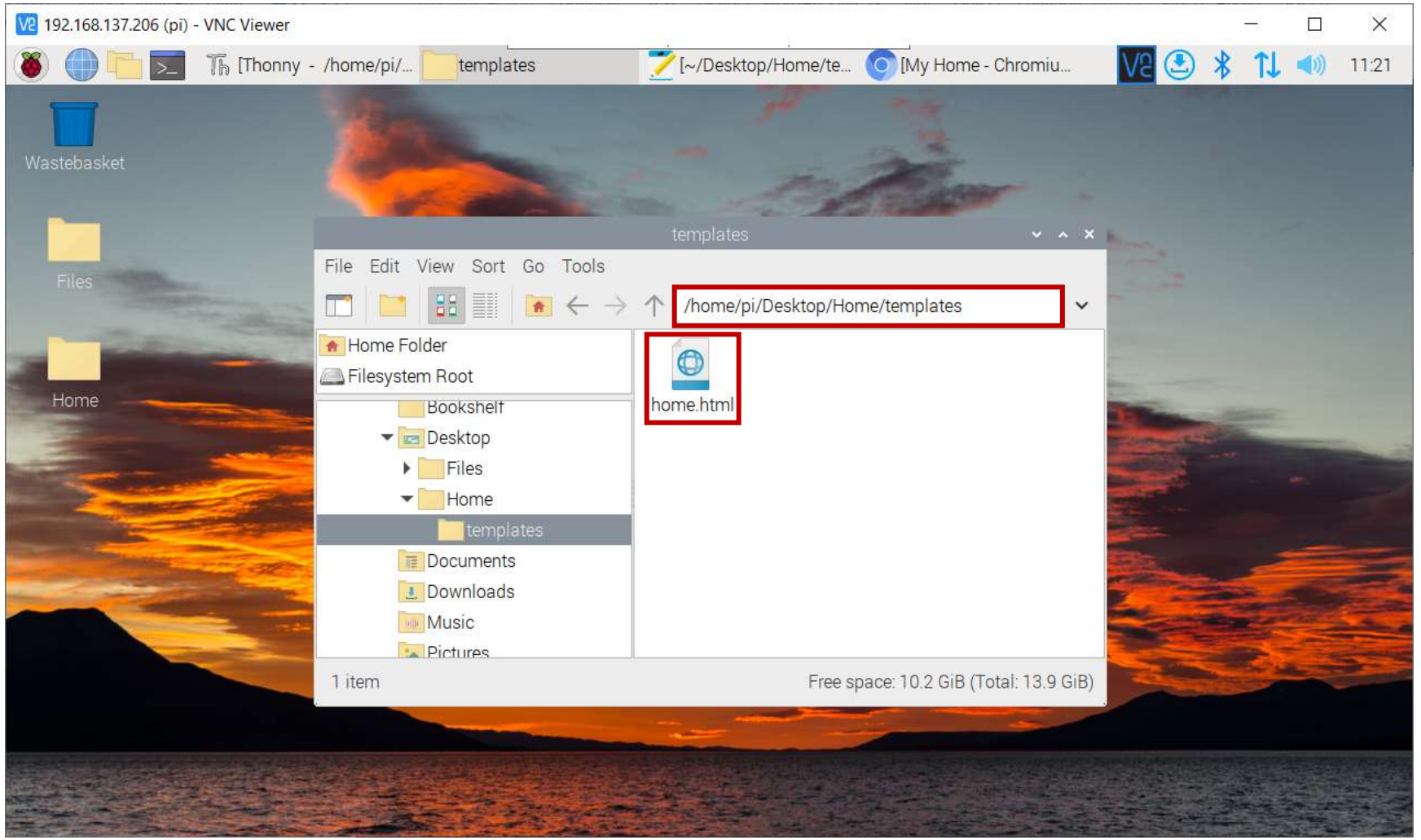
Smart Home: Project Files

- Create a new folder named `templates` in the `Home` folder.

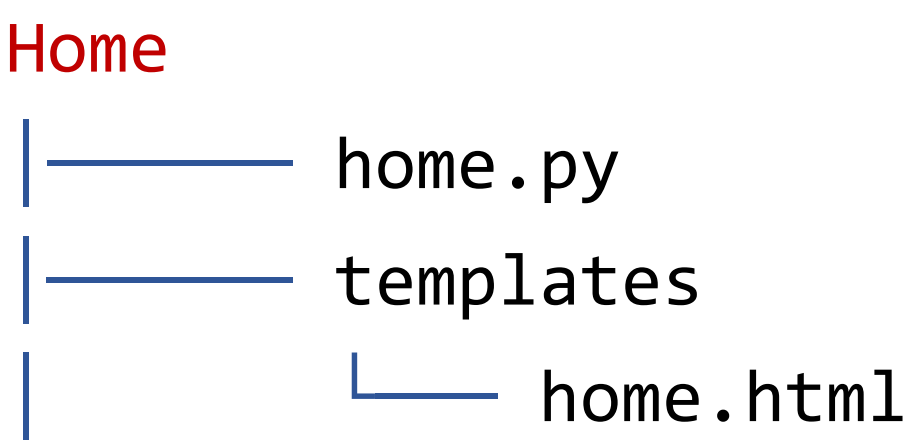


Smart Home: Project Files

- Create a new file named `home.html` in the `Home/templates` folder.

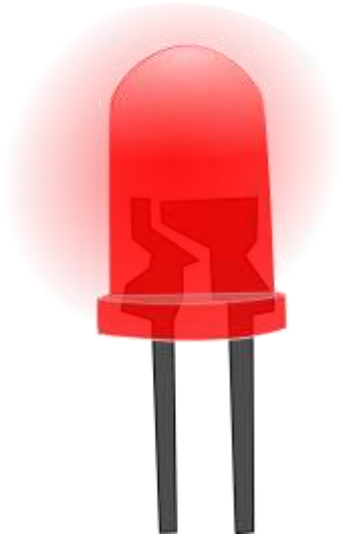


Smart Home: Project Hierarchy



Smart Home: Simple Idea

- When the user enters `0.0.0.0:5000/on`, the LED turns on.
- When the user enters `0.0.0.0:5000/off`, the LED turns off.



0.0.0.0:5000/on

0.0.0.0:5000/off

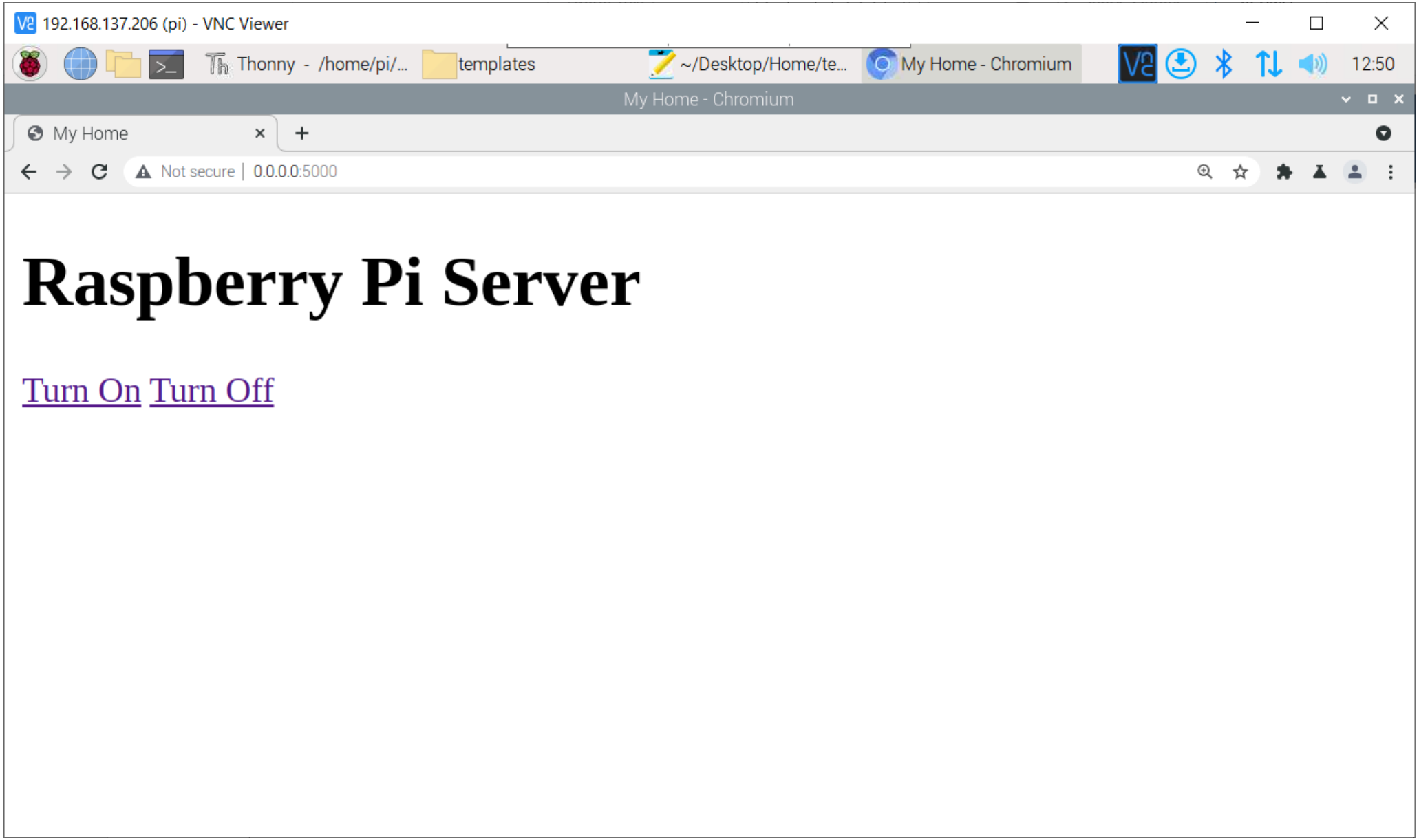
Smart Home: Python Code

```python
from RPi import GPIO                          # Import GPIO
from flask import Flask, render_template      # Import Flask

pin = 11                                      # Set pin number
GPIO.setmode(GPIO.BOARD)                      # Use board pin numbering
GPIO.setup(pin, GPIO.OUT)                     # Set pin 11 as output

app = Flask(__name__)                         # Create Flask object

@app.route('/')
def home():
    return render_template('home.html')       # Go to home page

@app.route('/on')
def led_on():
    GPIO.output(pin, 1)                       # Turn the LED on
    return render_template('home.html')       # Go to home page

@app.route('/off')
def led_off():
    GPIO.output(pin, 0)                       # Turn the LED off
    return render_template('home.html')       # Go to home page

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)        # Start the server (0.0.0.0:5000)
```

Smart Home: HTML

```
<!DOCTYPE html>

<html>
<head>
    <title>My Home</title>
</head>

<body>
    <h1>Raspberry Pi Server</h1>
    <a href="/on">Turn On</a>
    <a href="/off">Turn Off</a>
</body>
</html>
```

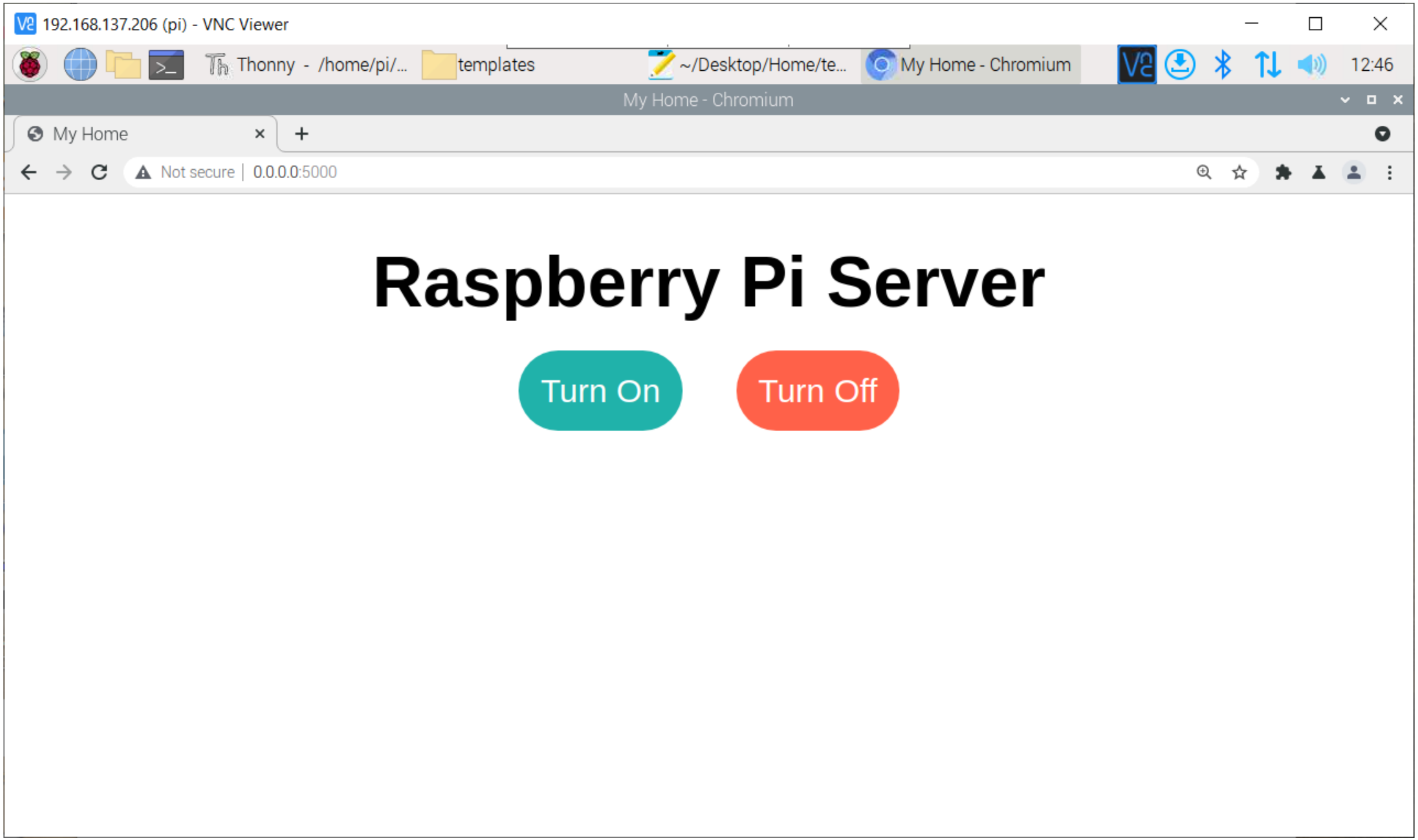
Smart Home: HTML



Smart Home: Better Look

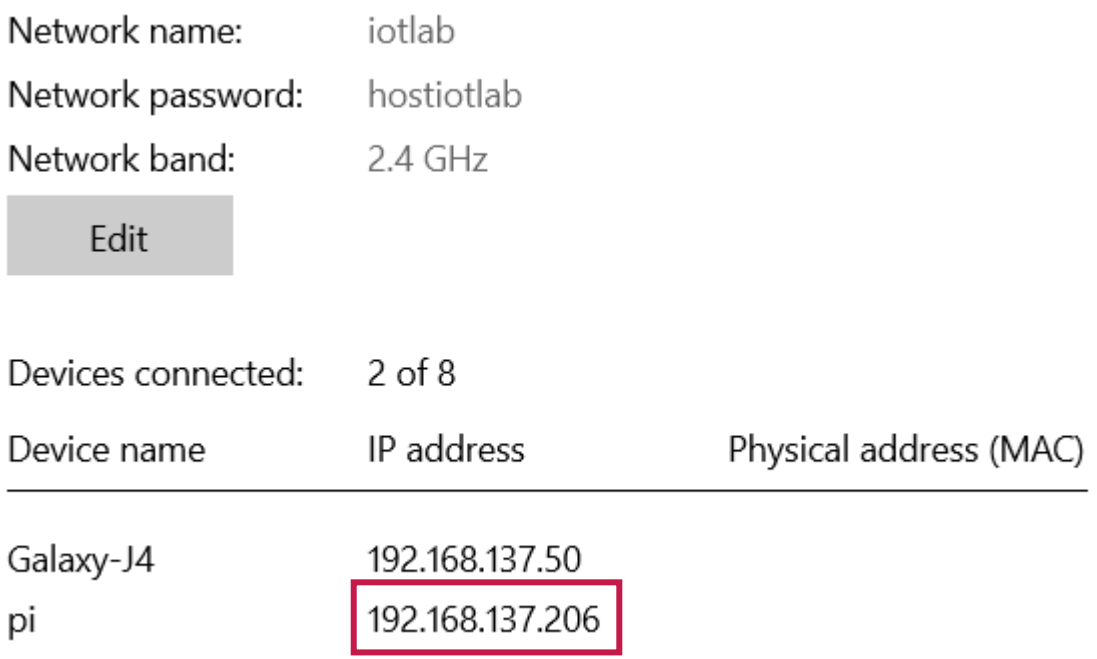
```
<!DOCTYPE html>
<html>
<head>
    <title>My Home</title>
    <style>
        body{text-align: center; font-family: Arial;}

        .btn{
                font-size: 15px;
                padding: 10px;
                margin: 10px;
                border-radius: 25px;
                text-decoration: none;
        }

        .on{background-color: lightseagreen; color: white;}
        .off{background-color: tomato; color: white;}
    </style>
</head>
<body>
    <h1>Raspberry Pi Server</h1>
    <a href="/on" class="btn on">Turn On</a>
    <a href="/off" class="btn off">Turn Off</a>
</body>
</html>
```

Smart Home: Better Look



Smart Home: Accessing Webserver From Anywhere

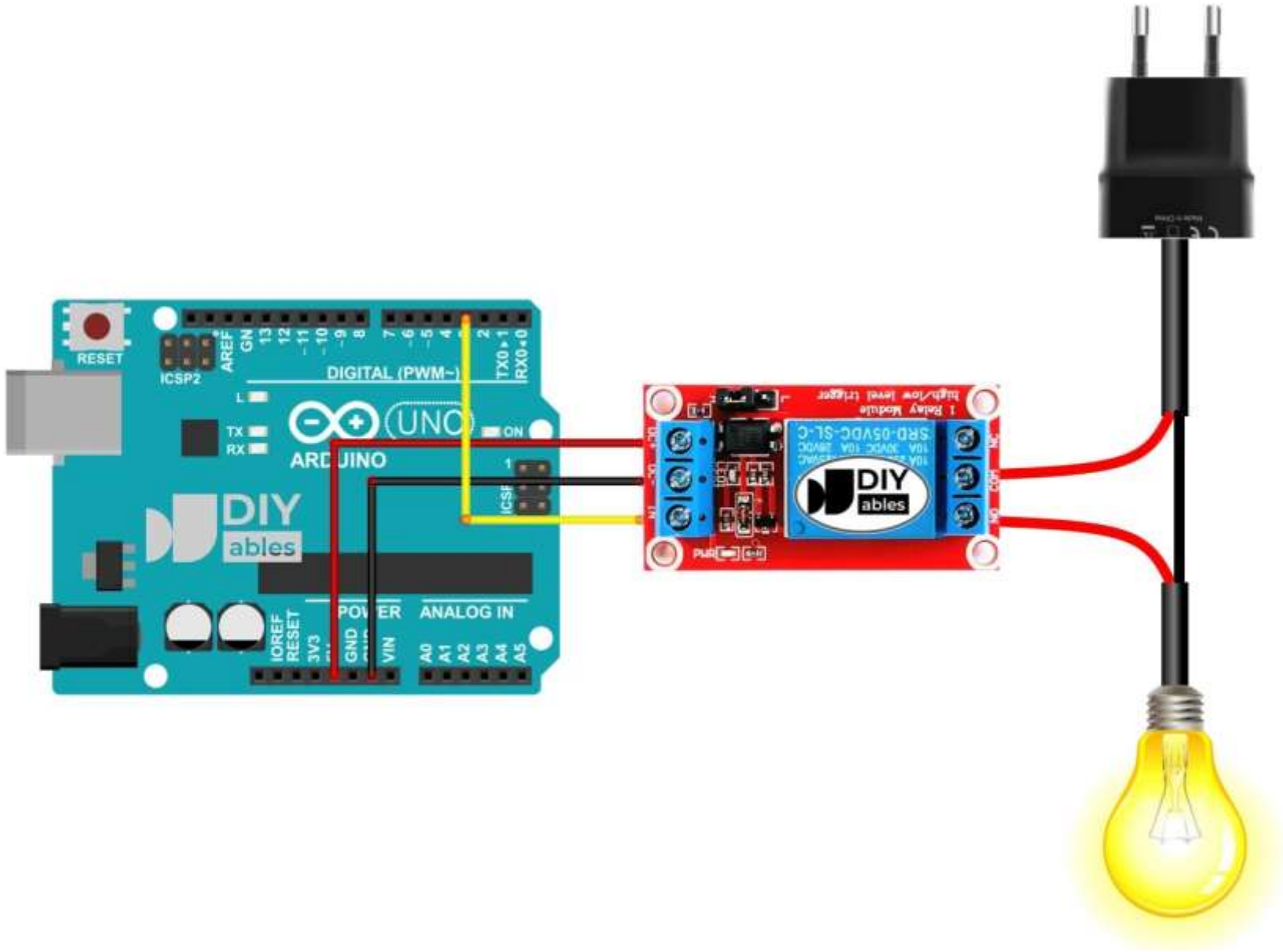
- After connecting your Raspberry Pi to power, it will be connected to your Wi-Fi automatically and have an IP address.
- Open Mobile hotspot, and copy that IP address.



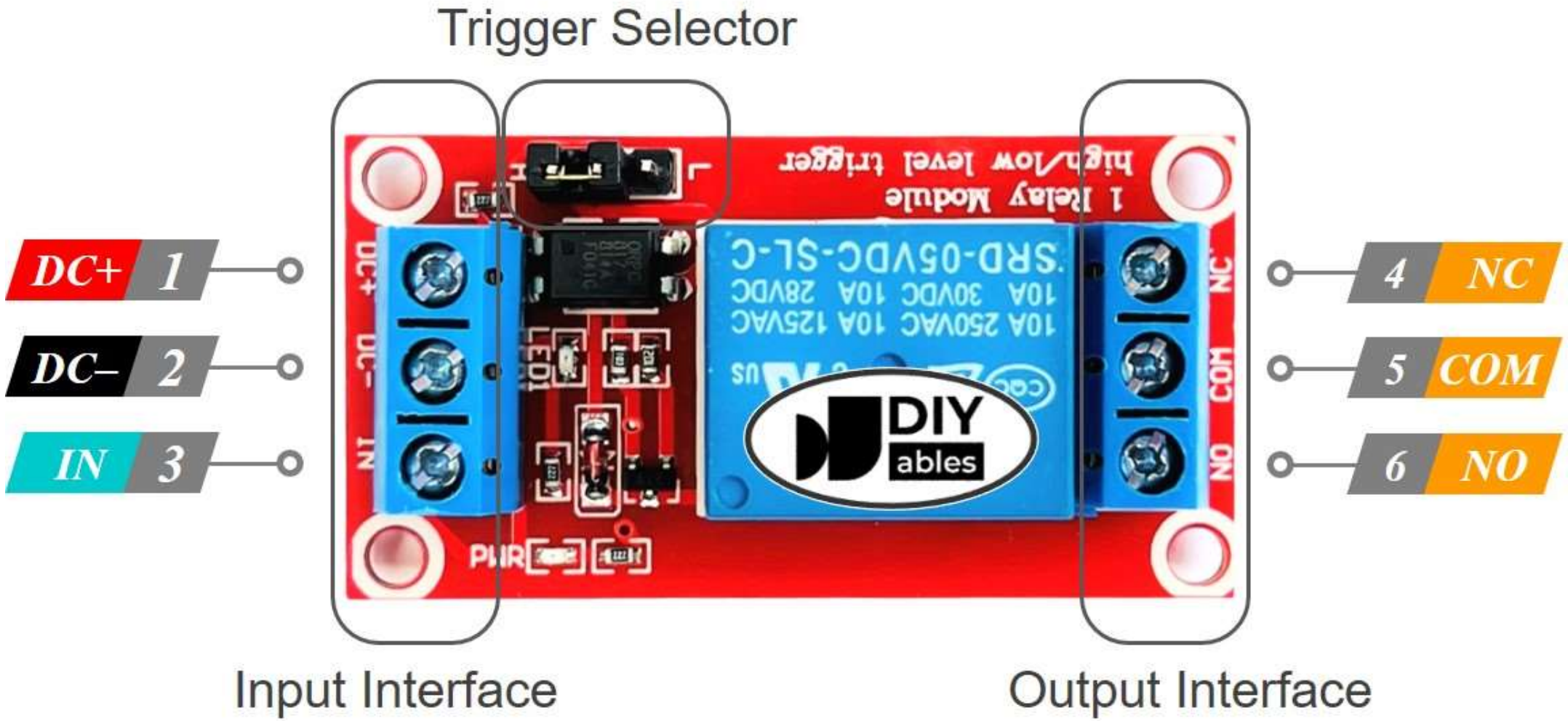
Smart Home: Accessing Webserver From Anywhere

- The copied IP is `192.168.137`.
- Open the browser and go to `192.168.137.206:5000`.

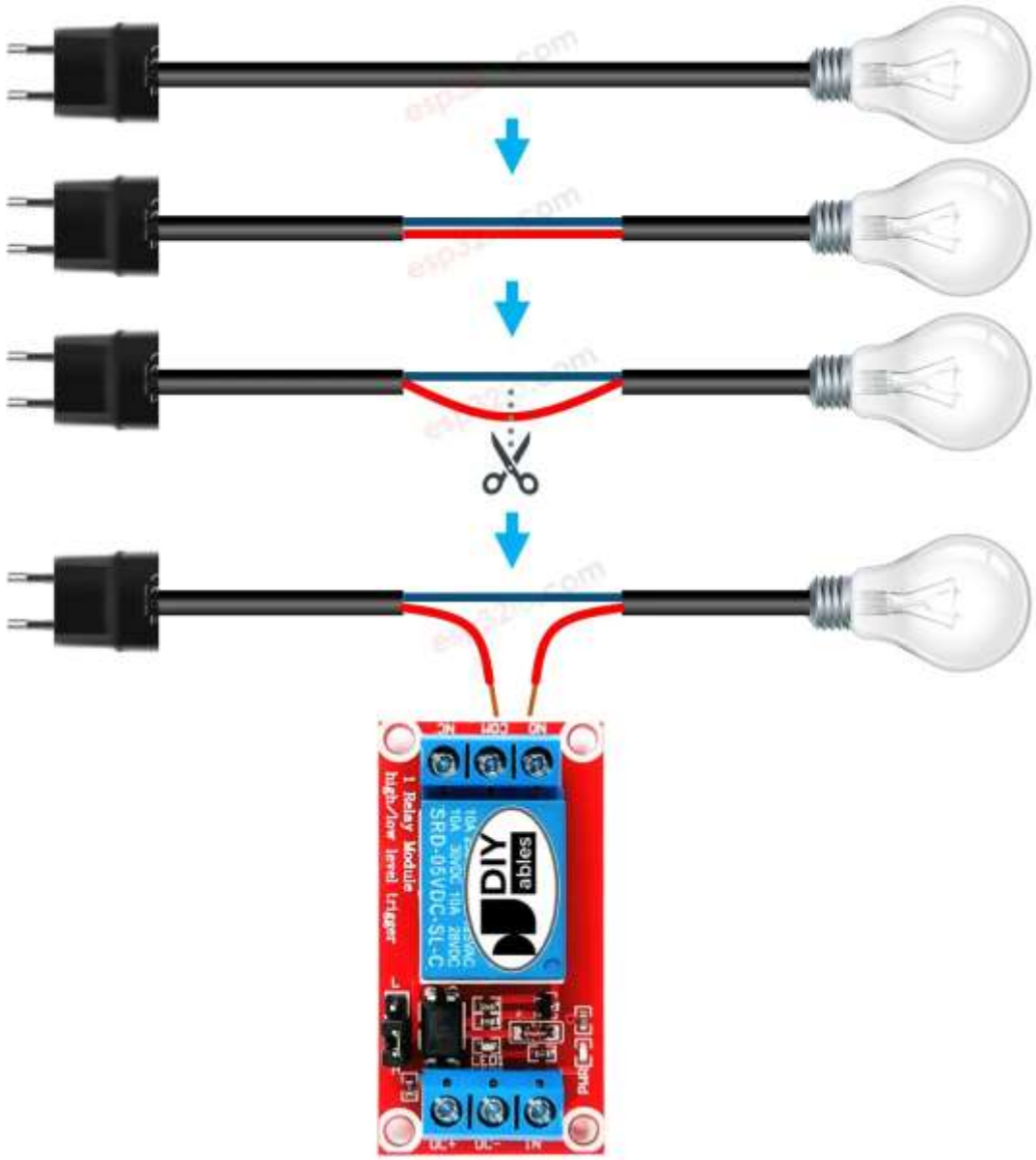
Control High Voltage Devices



Control High Voltage Devices



Control High Voltage Devices



References

- How to Use Raspberry Pi GPIO Pins
- GPIO Programming on the Raspberry Pi
- Raspberry Pi GPIO Tutorial: The Basics
- Python WebServer With Flask and Raspberry Pi
- Raspberry Pi GPIO Home Automation
- Python (RPi.GPIO) API
- RPi.GPIO Basics
- Configure Flask Server to be Visible Across the Network
- Arduino Connection with Relay Module
- ESP32 – Relay
- Relay Module with Arduino