# Overview

Let’s use a button as an input device on your Raspberry Pi. When the button is pressed, the General Purpose Input/Output (GPIO) connected to the button will change to low level (0V). You can detect the state of the GPIO using a Python script so that you can determine when the user has pressed the button.

# Step 1: Connect It

Use the following wiring diagram to connect the sensor to your Raspberry Pi



**Important Notes:**

* The wire colors do not matter. Just use whatever is available at your station.
* **Make sure that you are plugging the wires into the correct location on both the Pi and the Sensor (especially the power and ground)**
* The wires move around a lot, so make sure they are firmly in place when you are testing

# Step 2: Code It

We have provided you with a Python code template that can run this sensor/device. The code is located at:

**/home/pi/Documents/rpi-iot-demos/button.py**

Use the Geany editor to open this file (NOTE: this will be slow, so be patient). Once it is open, take some time to read the code to see how it works.

Once you are ready, do the following:

1. Find the function called OnStateChange().
   1. This function is called whenever the Raspberry Pi detects that the button has been pressed or depressed.
   2. The buttonPin parameter is set to 0 when the button is pressed, and 1 when it is not pressed
2. Modify this function so that it prints a message to the console whenever the button is pressed.

def OnStateChange(buttonPin):

if buttonPin == 0:

print("Button Pressed")

if buttonPin == 1:

print("Button Depressed")

# Step 3: Run / Test It

1. To run the script, open a Linux console and navigate to the folder with your code
   1. Helpful Linux Commands:  
      **ls** lists the contents of the directory  
      **cd <folder\_name>** opens a folder (don’t type the < > characters)  
      **cd ..** exits the folder you are currently in  
      **cd ~** takes you back to your home folder
2. Run the python program by typing the following:

**python3 button.py**

1. Press the button and see what happens!
   1. If it doesn’t work, make sure that your wiring is correct and that you see lights on the sensor / device
2. **To exit the program, press Ctrl-C in the terminal**

# Step 4: Make It Control Something

To make your program send MQTT messages, modify your main program to look like the following:

# --------------------------------------------

# Main Program Starts Here

# --------------------------------------------

if \_\_name\_\_ == '\_\_main\_\_':

comm.connect()

setup()

try:

print("Program Running. Waiting for a Button Press")

loop()

except KeyboardInterrupt:

destroy()

Now, go to the OnStateChange() function you modified earlier. You can use the comm.send function to send a message when the button is pressed, etc.

comm.send("CHANNEL NAME GOES HERE", "MESSAGE GOES HERE")