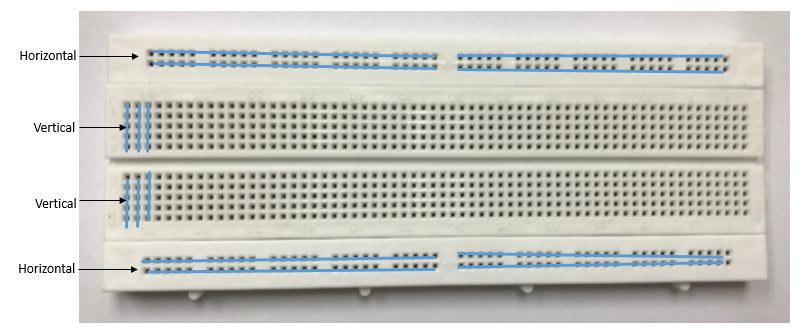
**LAB 1-1 SETTING UP YOUR BREADBOARD AND CONNECTING DEVICES TO IT**

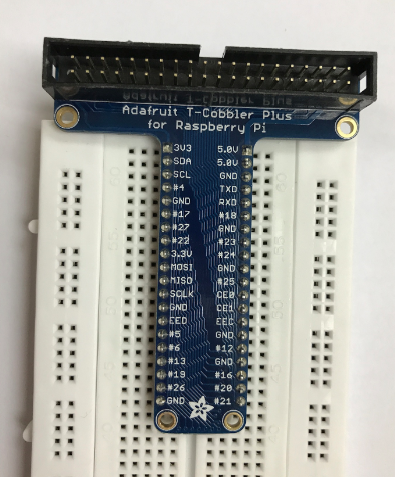
|  |  |
| --- | --- |
| Description | In this lab, you will setup your breadboard to be connected to the Raspberry Pi using the Adafruit T CobblerPlus Kit and connect an LED light and Push Button switch to it and write Python programs to make these devices work. |
| What you will learn | * How to setup your breadboard to be connected to the Raspberry Pi * How to connect an LED light and a Push Button switch to the Pi using the breadboard. * How to write Python programs to make these devices work |
| What you need | * Adafruit T Cobbler Plus and Breakout Cable * Breadboard and Jumper wires * LED light (red, green, yellow) * Push Button switch |
| Duration | 45 minutes |

**Connecting your Cobbler Kit to your Breadboard**

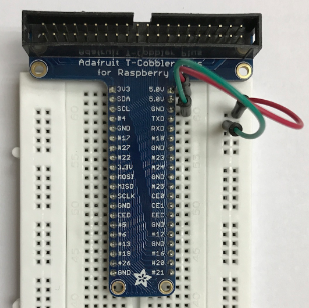
1. Your breadboard is connected internally as follows:



1. Connect your Cobbler Kit to your breadboard based on the diagram below:

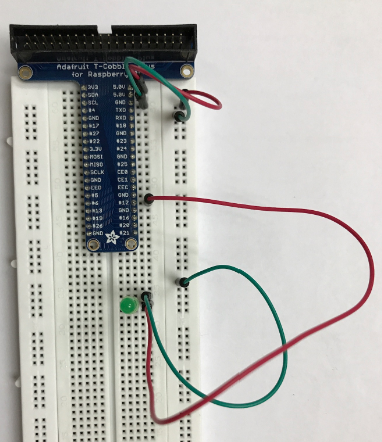


1. It is conventional to use the horizontal rows below and above to connect to Gnd and 5V respectively.



**Connecting and programming LEDs**

1. Connect an LED to your breadboard and Cobbler kit as follows:
   1. Connect **Pi Cobbler #12** to **Positive end of LED (LONG end)**
   2. Connect **Pi Cobbler GND** to **Negative end of LED (SHORT end)**



1. Connecting to VNC Viewer
   1. Turn on your Pi and wait a few mins
   2. Connect to SSID from your Raspberry Pi, eg pi723\_1)
   3. Wireless password : raspberry
   4. Launch VNC Viewer
   5. Connect to 192.158.1.8
   6. Username: pi
   7. Password: raspberry
2. Create a new Python script named **ledlights.py** and save it in the **/home/pi directory**

Codes are available on your Pi Desktop/P1Codes

import RPi.GPIO as GPIO

import time

from time import sleep

#dismisses error messages

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM)

#Creating a blinking method

def blinking(pin):

#turns on LED for 2 seconds

GPIO.output(pin, GPIO.HIGH)

time.sleep(2)

#turns off LED for 2 seconds

GPIO.output(pin, GPIO.LOW)

time.sleep(2)

return

#states where the LED pin is

LED\_PIN = 12

GPIO.setup(LED\_PIN, GPIO.OUT)

#Calling the Blinking method over a span of 20 seconds

for i in range (0, 20):

blinking(LED\_PIN)

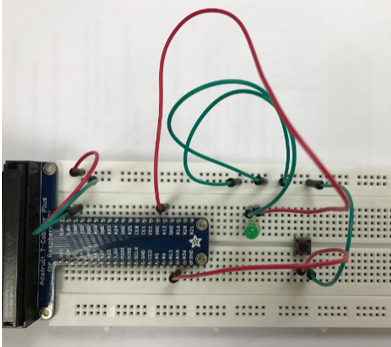
GPIO.cleanup()

**Notes on this program**

* import RPi.GPIO as GPIO allows us to make use of the Pi’s GPIO pins
* GPIO.setwarnings(False) prevents the program from displaying unimportant warning messages during code execution
* GPIO.BCM indicates that you refer to the Pi’s GPIO pins by the Broadcom SOC channel number
* LED\_PIN = 12 states that the LED is using Pin 12 on the Cobbler Kit
* GPIO.output(pin, GPIO.HIGH) turns on the LED, while GPIO.output(pin, GPIO.LOW) turns the LED off

**Connecting Push Button to Illuminate LEDs**

1. Add the push button to the existing connections on the breadboard:
   1. Connect **Pi Cobbler #13** to **left end of push button**
   2. Connect **GND from Pi Cobbler** to **right end of push button**



1. Create a Python script names **pushbtnled.py** and save it in the **/home/pi directory**

import RPi.GPIO as GPIO

import time

from time import sleep

#dismisses error messages

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM)

#states where the LED pin is

LED\_PIN = 12

GPIO.setup(LED\_PIN, GPIO.OUT)

#states where the Push Button pin is

PUSH\_BUTTON = 13

GPIO.setup(PUSH\_BUTTON, GPIO.IN, pull\_up\_down=GPIO.PUD\_UP)

while True:

if GPIO.input(PUSH\_BUTTON) == False:

GPIO.output(LED\_PIN, GPIO.HIGH)

time.sleep(1)

GPIO.output(LED\_PIN, GPIO.LOW)

GPIO.cleanup()