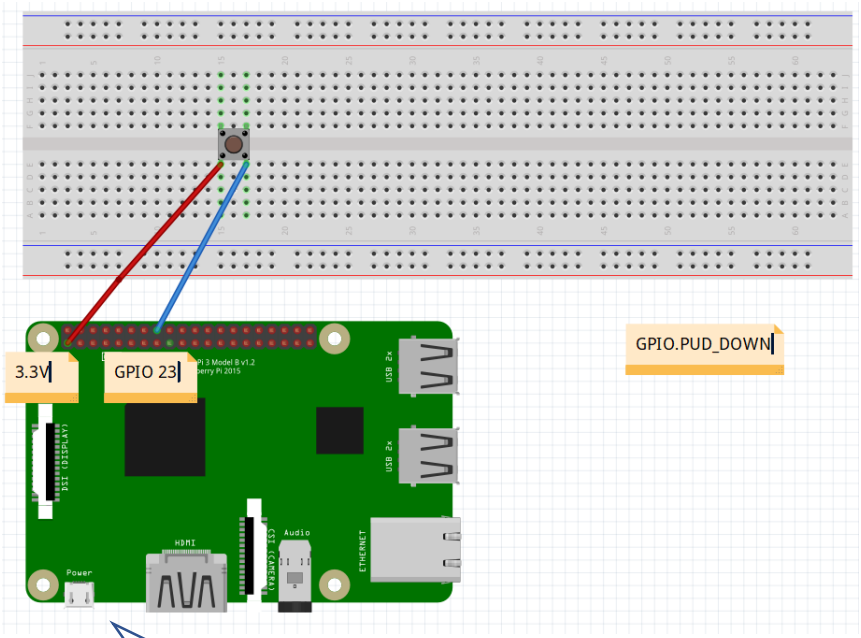
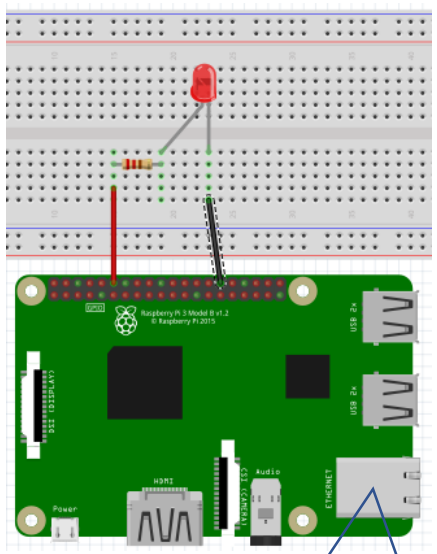
LED 원격 제어 Lab 3

(실습과제 4) switch를 이용한 led 원격 제어





서버(server.py)

클라이언트(client.py)

스위치를 누르면

json

{ “cmd” : “led”, “led_no” : 18, “act” : ”on”}

GPIO.setmode(GPIO.BCM)

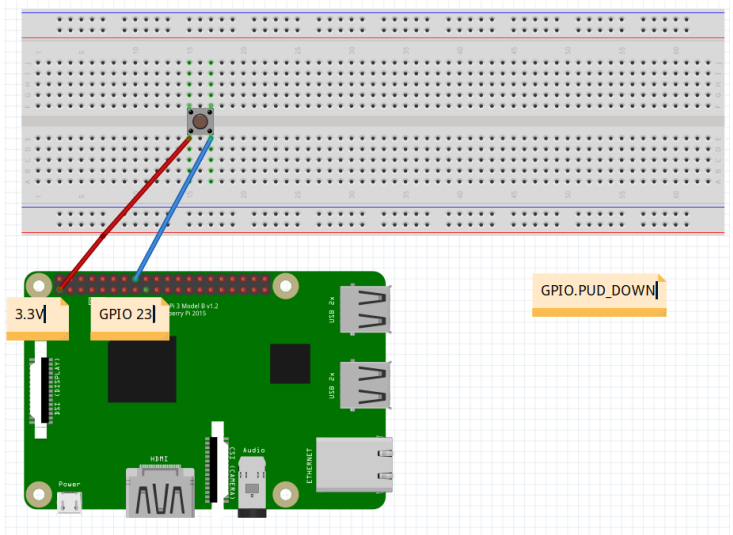
GPIO.setup(led_pin, GPIO.OUT)

GPIO.output(led_pin, True)

LED 원격제어 - client

```
import socket
import RPi.GPIO as GPIO
import json
import threading

class swThread(threading.Thread):
    def __init__(self, s):
        threading.Thread.__init__(self)
        self.csocket = s

    def run(self):
        sw_pin = 23
        before = 0
        GPIO.setmode(GPIO.BCM)
        GPIO.setup(sw_pin, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
        while True:
                after = GPIO.input(sw_pin)
                while before == after :
                        after = GPIO.input(sw_pin)
                before = after
                print('switch : ', after)

                data = {}
                data['cmd'] = 'led'
                data['led_no'] = 18
                if after == 1 :
                        data['act'] = 'on'
                elif after == 0 :
                        data['act'] = 'off'
                body = json.dumps(data)
                s.sendall(bytes(body, 'UTF-8'))
```

```
s = socket.socket()
host = '127.0.0.1'
port = 9000

s.connect((hostm port))
n = swThread(s)
n.start()

while True :
        message = input()
        if message == 'bye':
                break
s.close()
```



LED 원격제어 - server

```
import socket
import RPi.GPIO as GPIO
import json

s = socket.socket()
s.setsocketopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)

host = '0.0.0.0'
port = 9000

s.bind((host, port))
s.listen(10)

GPIO.setmode(GPIO.BCM)

while True:
    c. addr = s.accept()
    print('Got connection from', addr)

    while True :
            data = c.recv(2048)
            msg = json.loads(data.decode())
            print('received data : ', msg)

            if msg['cmd'] == 'led' :
                    led_pin = msg['led_no']
                    GPIO.setup(led_pin, GPIO.OUT)

                    if msg['act'] == 'on' :
                            GPIO.output(led_pin, True)
                    elif msg['act'] == 'off' :
                            GPIO.output(led_pin, Flase)

    c.close()
s.close()
```