**Python IP connection script**

The given Python script enables Windows to connect to a Raspberry Pi through a SSH connection. The Python script can be further modified to perform tasks other than turning on the LED. To do this, modify the code within the quotation marks under 'python\_scripts'.

1. On your laptop, copy and paste the following code in VScode inside a virtual environment, name the file as LED\_SSH.py and pip install paramiko.

**pip install paramiko**

1. For this section, you will need 2 RPI CPEs. One connected to the monitor, mouse and keyboard while the other is connected to your laptop.
2. On your laptop, refer to Lab 6 part 1, do the steps 1-6.
3. Inside the Raspberry Pi Desktop, refer to Lab 6 part 1, do the steps 7-13.
4. After you obtained wwan0’s public IP address, replace the highlighted part with it.

import paramiko

# Replace these with your Raspberry Pi's IP address, username, and password.

raspberry\_pi\_ip = "100.66.52.21"

username = "pi"

password = "raspberry"

# Establish SSH connection

ssh = paramiko.SSHClient()

ssh.set\_missing\_host\_key\_policy(paramiko.AutoAddPolicy())

ssh.connect(raspberry\_pi\_ip, username=username, password=password)

# Raspberry Pi Python script to Turn on the LED

python\_script = """

import RPi.GPIO as GPIO #import RPi.GPIO module

from time import sleep #used to create delays

GPIO.setmode(GPIO.BCM) #choose BCM mode

GPIO.setwarnings(False)

GPIO.setup(22,GPIO.OUT) #set GPIO 22 GPIOas output

GPIO.output(22,1) #output logic high/'1'

"""

python\_script2 = """

import RPi.GPIO as GPIO #import RPi.GPIO module

from time import sleep #used to create delays

GPIO.setmode(GPIO.BCM) #choose BCM mode

GPIO.setwarnings(False)

GPIO.setup(22,GPIO.OUT) #set GPIO 22 GPIOas output

GPIO.output(22,0) #output logic Low/'0'

"""

while True:

    Input\_value=input("Enter '1' to turn LED on or '0' to turn LED off ('Q' to quit):")

    # Exit

    if Input\_value=='Q':

        break

    # Send the script and execute it on the Raspberry Pi

    if Input\_value=='1':

        stdin, stdout, stderr = ssh.exec\_command("python -c '{}'".format(python\_script))

        print("LED ON!!!")

    if Input\_value=='0':

        stdin, stdout, stderr = ssh.exec\_command("python -c '{}'".format(python\_script2))

        print("LED OFF!!!")

# Close the SSH connection

ssh.close()

1. On your laptop, do a ping test to check if you can ping the public IP address.
2. If yes, continue to connect the LED to the 5G RPI CPE via GPIO pins.

The diagram below shows the GPIO pins layout of the Raspberry Pi. Do note that the 5G Hat is mounted directly on top of the Raspberry Pi. So, the GPIO pins seen on the 5G Hat is the same as the diagram below. We are using pin GPIO 22 which is D22 on the 5G Hat.

A diagram of a circuit board

Description automatically generated

1. Follow the diagram below to connect the LED to the 5G RPI CPE. Put a resistor as well.

A diagram of a circuit

Description automatically generated

1. On your laptop, run the LED\_SSH.py in VScode.
2. Enter ‘1’ to turn on LED, ‘0’ to turn off LED and ‘Q’ to quit.
3. **Troubleshooting**: Make sure you can ping the public IP address of the wwan0 interface, ensure that the LED and resistor is connected correctly to the 5G Hat GPIO pins.