ET0735 - DEVOPS FOR AIOT
SCHOOL OF ELECTRICAL AND ELECTRONIC ENGINEERING, SINGAPORE POLYTECHNIC

LABORATORY 4: INTRODUCTION TO RASPBERRY PI

Objectives

By the end of the laboratory, students will be able to

- Introduction to the Raspberry Pi
- Configure the Raspberry Pi in "Headless Mode"
- Adapt Raspberry Pi OS configuration to auto-start a Python script

Activities

- Introduction to Raspberry Pi and Development board hardware
- Setting up remote connection to Raspberry Pi
- Configure VS Code remote debug session with Raspberry Pi

Review

- Raspberry Pi is configured in "Headless Mode"
- VS Code is configured to remotely debug and run Python code on the Raspberry Pi
- Successfully run a Python demo application on the Raspberry Pi Development board to test the LED, LCD, Buzzer, Servo Motor and DC Motor

Equipment:

- Windows OS laptop
- Raspberry Pi + Development Board
- USB-C 5 volt power adapter
- 9 volt power adapter
- USB to Ethernet adapter

1 Setting up of Remote Connection to Raspberry Pi

1.1 To develop applications on the Raspberry Pi, the first step is to connect remotely from your laptop to the Raspberry Pi.

1.2 Power up your Raspberry Pi and the Development Board and connect the Ethernet Network cable by connecting the following

1. DC 5 volt supply using USB-C connector to the Raspberry Pi
2. Ethernet Network cable
3. DC 9volt supply to the Development as shown in the diagram below.

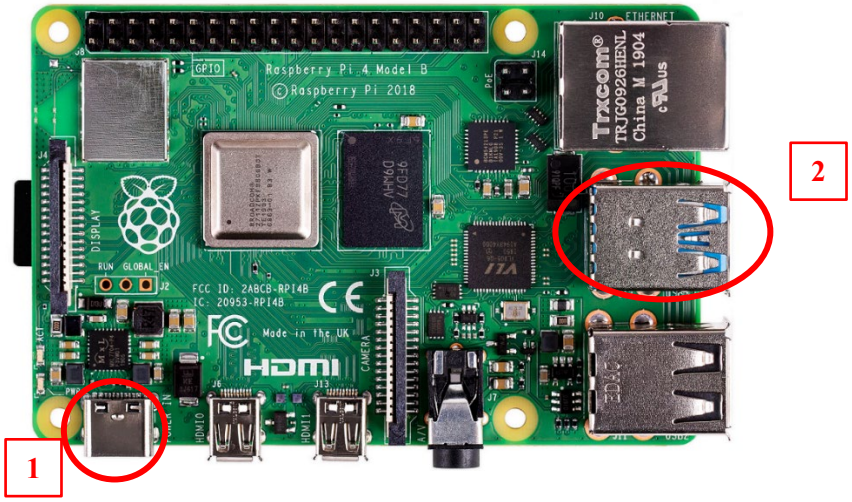

Figure 1 – Raspberry Pi 4

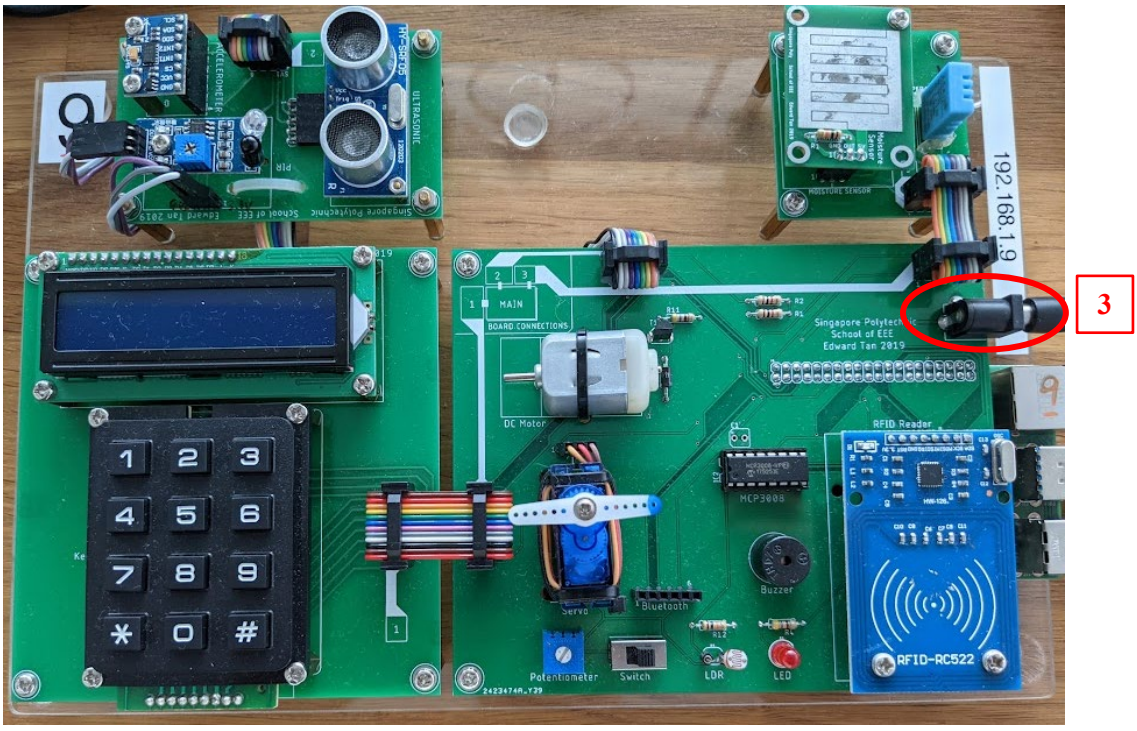

Figure 2 – Raspberry Pi Development Board

1.3 Right click on the Windows Start menu below and select “Network Connections”,

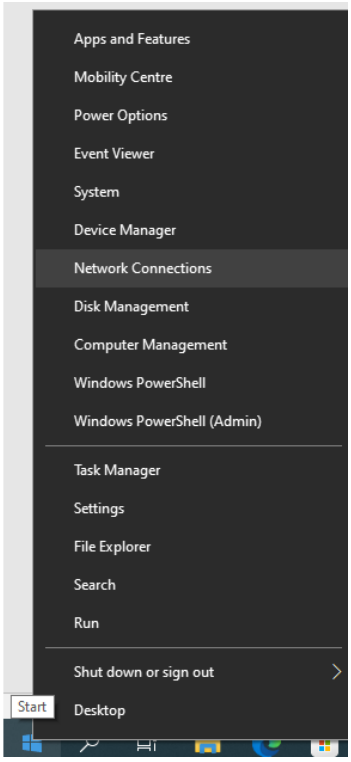

Figure 3 – Select Network Connections

1.4 Select “Ethernet” below and then select “Change adapter options”

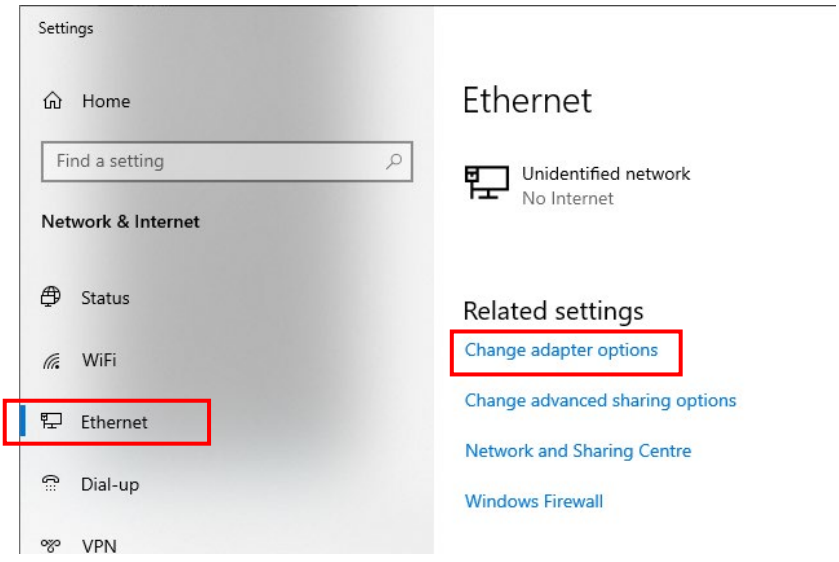

Figure 4 – Windows “Ethernet” settings

1.5 Select the Ethernet adapter that is physically connected to the Ethernet port that is connected to the Raspberry Pi below then right click and select “Properties”

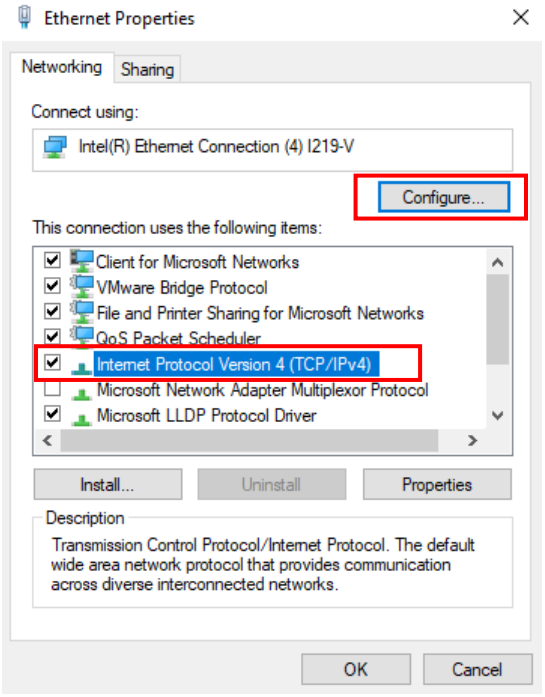

Figure 5 – Ethernet Configuration

1.6 Select “Internet Protocol Version 4 (TCP/IPv4) then click “Properties” and configure the following settings below,

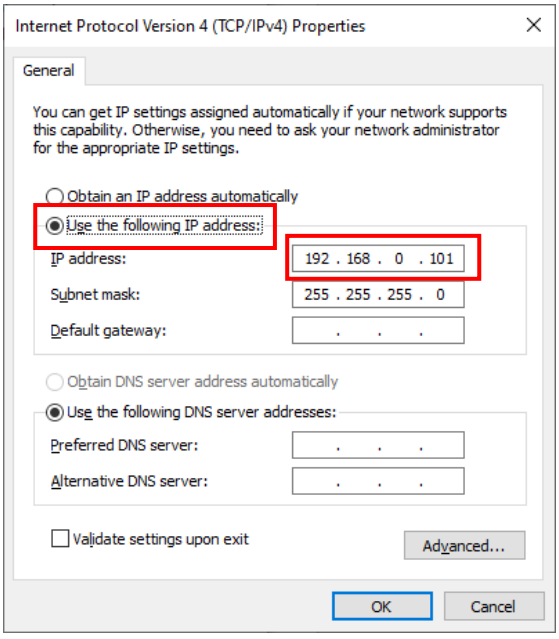

Figure 6 – IP configuration Settings

1.7 Check that the connection between your laptop and the Raspberry Pi is working by running a "ping" command below at the Windows Command Prompt below to ping the Raspberry Pi which has an IP Address = 192.168.0.100

If the Network connection was successfully setup on your laptop and the Raspberry Pi then you should see a similar result above when the "ping" has 0% loss packets.

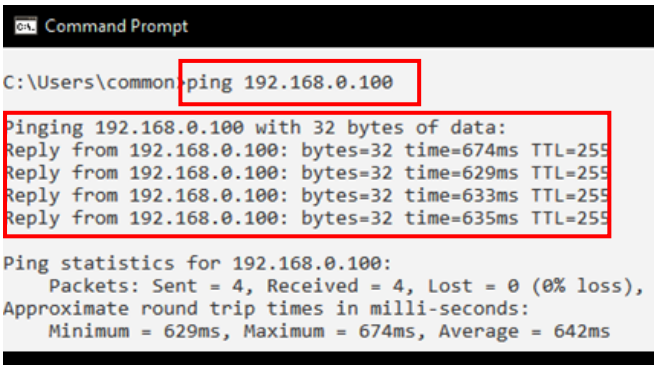

Figure 7 – Ping communication check between PC and Raspberry Pi

2 Installation of Putty

We will use the PuTTY tool to allow us to connect to the Raspberry Pi and remotely run Linux command from our laptop via the SSH protocol

2.1. Download PuTTY for Windows from https://www.putty.org/ and install it in your laptop.

2.2. Connect the Ethernet cable from your laptop to the Raspberry Pi. If your laptop does not have a physical Ethernet port, please request to loan a USB to Ethernet Adapter from your lecturer.

2.3. Run Putty and Connect to the Raspberry Pi using the IP address 192.168.0.100

2.4. If this is the first time you are connecting to the Raspberry Pi via Putty, then you should see the prompt window "PuTTY Security Alert" to accept new Security keys.

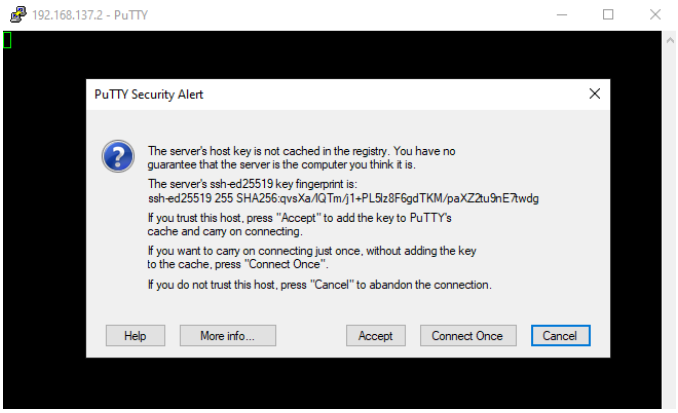

Figure 8 – Ping communication check between PC and Raspberry Pi

2.5. After accepting the security key, enter the username as “pi”. When prompted for password, enther “password”. Note that unlike most of other software, when you enter the password, no ‘*’ will be shown.

Figure 9 – PuTTY login prompt

2.6. After entering the username and password, the SSH remote terminal session will start and show the console output (Figure 10).

2.7. You are now ready to start running Linux commands remotely via the PuTTY SSH Terminal.

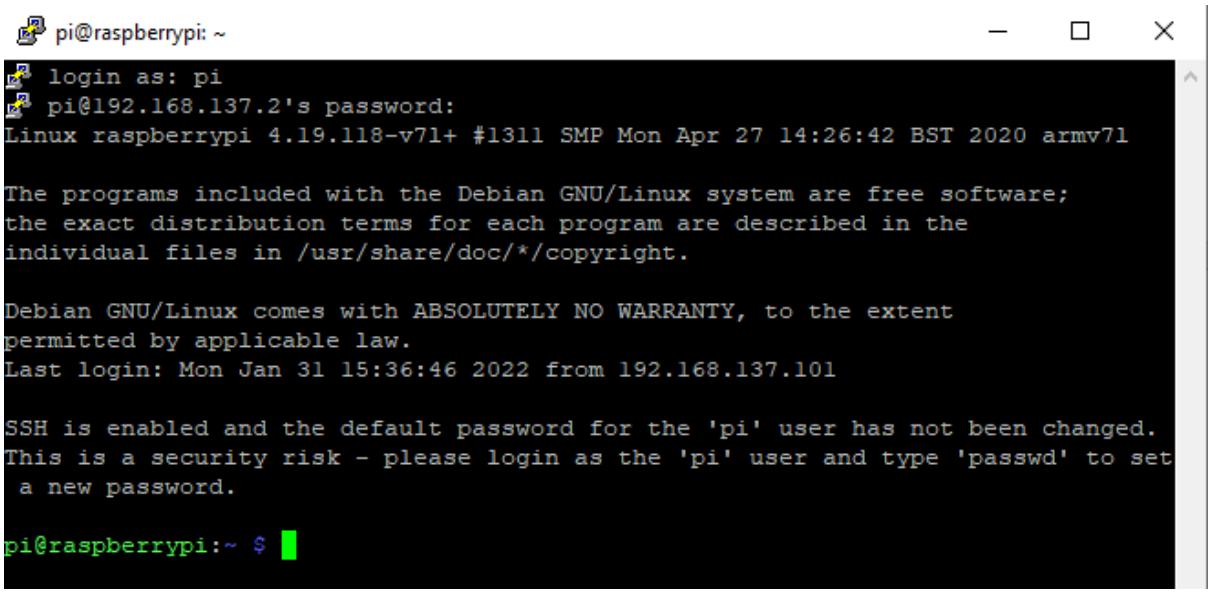

Figure 10 – PuTTY Remote Terminal

3 Installation of VNC Viewer

We will now install VNC Viewer which is a Remote Desktop tool that allows us to connect to the Raspberry Pi remotely and view the Raspberry Pi OS Desktop interface.

3.1. Download VNC Viewer for Windows from the link below

https://downloads.realvnc.com/download/file/viewer.files/VNC-Viewer-6.22.315-Windows.exe?_ga=2.227493657.524886960.1651897950-1801372009.1651897950

3.2. After installing VNC Viewer, create a new connection instance to the Raspberry Pi from "File □ "New Connection"

3.3. In the "New Connection", set the field "VNC Server" to the static IP Address of the Raspberry Pi = **192.168.0.100:5900**

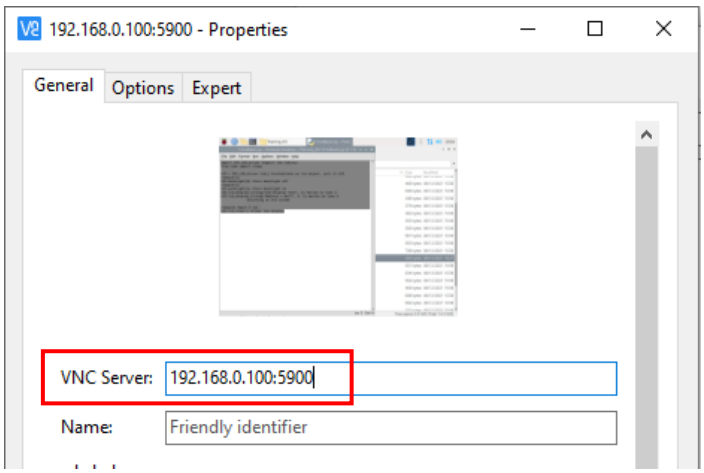

Figure 11 – VNC Viewer to Raspberry Pi connection

3.4. Using the same "user name = pi" and password = "password", start a VNC Remote Desktop connection to the Raspberry Pi and you should see the Raspberry Pi desktop below.

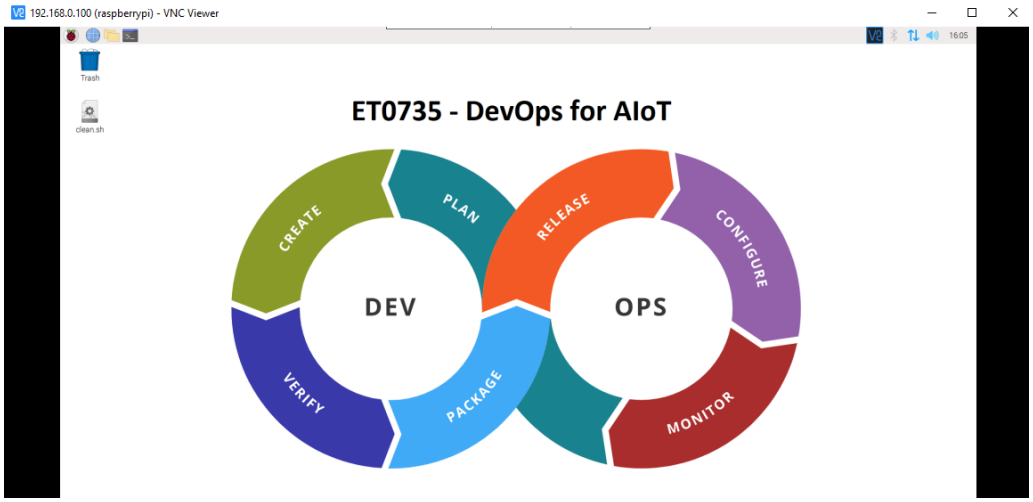

Figure 12 – VNC Viewer Raspberry Pi Remote Desktop

4 Introduction to Linux Commands

The Raspbian OS running on the Raspberry Pi is a Linux based OS, therefore you will need to be familiar with several Linux commands to develop applications on the Raspberry Pi.

Since we are developing on the Raspberry Pi in "headless" mode, ie: without the connection of a physical keyboard, mouse and display, all the Linux commands will be executed remotely via a SSH session using Putty in Windows or Terminal in Mac OS.

Although we are using the Raspberry Pi in "headless" mode, we still can access Raspian OS display and issue commands in the terminal directly by connecting through VNC. So it is not necessary to execute linux commands through Putty's SSH connection.

The following are some common Linux commands that are useful,

Linux Command	Description	Example
pwd	Displays the current directory path	
ls	Lists all the files and folders in the current directory	
mkdir	Makes a new directory at the current or specified path	
cd	Changes the current path to the specified directory path	
sudo	Changes the user permissions level to the root user which may be required to run some Linux commands	"sudo shutdown now"

Table 1

Exercise 1

In the next section of this lab we will clone and run a sample Python application on the Raspberry Pi.

To prepare for this, we first need to create a new directory in the Raspberry Pi where a Github repository will cloned in the next part of this lab,

In the next section of this lab, we will need to upload the local VS Code project to the Raspberry Pi. Therefore, we first need to create a new folder on the Raspberry Pi.

1. Before working on the exercises on the Raspberry Pi, please double click on the bash script file "clean.sh" on the Desktop of the Raspberry Pi using the VNC Viewer shown below

2. When the pop-up box appears, select "Execute" and check that the folder at /home/pi/ET0735 is removed

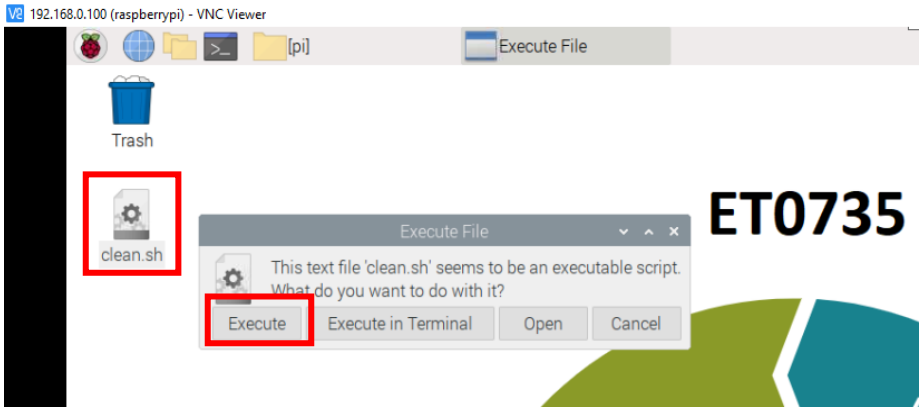

Figure 13

3. Using PuTTY connect to the Raspberry Pi terminal and enter a Linux command to create the directory **/home/pi/ET0735/Lab4**

Write down the Linux command/s you used below

2. Change to the directory **/home/pi/ET0735/Lab4** and check that the directory change is successful.

Write down the Linux command/s you used below

5 Cloning Github repository containing Git submodules

The Lab 4 Github repository contains a git submodule 'hal' which contains the Hardware Abstraction Layer (HAL) code and provides Application Programming Interfaces (APIs) that allow us to access the sensors and actuators connected to the Raspberry Pi on the development board.

5.1. To clone all the files in a Git repository including files directories with Git submodules, we need to add the additional parameter "--recurse-submodules" to the Git "clone" command that we have been using in the previous labs.

5.2. On your laptop, make sure that you are in 'C:\Local_Git_Repository'(use cd command to navigate). Clone the Lab4 repository for Lab 4 from the URL https://github.com/ET0735-DevOps-AIoT/Lab4 using the slightly modified Git clone command below,

```
git clone --recurse-submodules https://github.com/ET0735-DevOps-AIoT/Lab4.git
```

5.3. After cloning the Lab 4 repository, check all the files have been cloned into your local Git repository including the Git submodule folder at **src/hal**. Make sure that the src/Hal folder is not empty and it contains the python driver files for the devices in the Raspberry Pi.

6 Setup of VS Code to Connect to Raspberry Pi

To develop and test Python code on the Raspberry Pi, we need to configure VS Code to connect remotely to the Raspberry Pi and allow for remote debugging and testing.

In the previous labs, we configured VS Code to use the Python interpreter installed on our local machine. However, we are now developing Python applications on the Raspberry Pi, therefore we will need to use the Python interpreter on the Raspberry Pi and not the interpreter on our laptops.

But before that, we need to install the extensions required to connect to the Raspberry Pi. Two extensions are required to work with Raspberry Pi.

(i) **SFTP extension** to transfer files from our laptop to the Raspberry Pi

(ii) **Remote-SSH extension** to work remotely from laptop to Raspberry Pi.

6.1. Launch VS Code software on your laptop.

6.2. When prompted to choose project, click "Open" and navigate to c:\Local_Git_Repository\Lab4

6.3. In the side panel on the left, click on "Extensions" icon and search for "SFTP". Click on the SFTP extension shown below (Figure 14) and install.

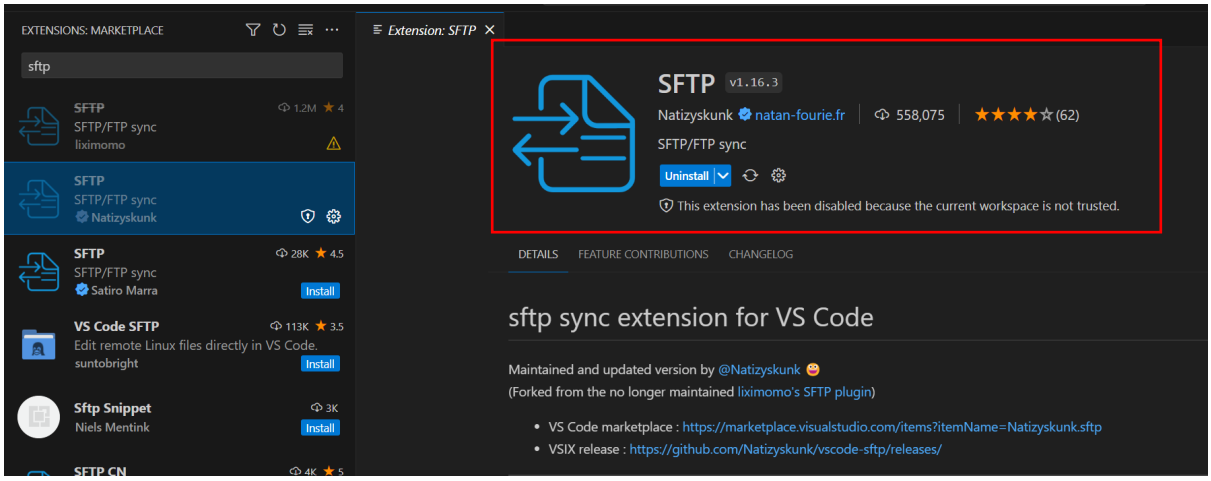

Figure 14 – Installing SFTP Extension

6.4. Next, search for "Remote – SSH" and install (Figure 15).

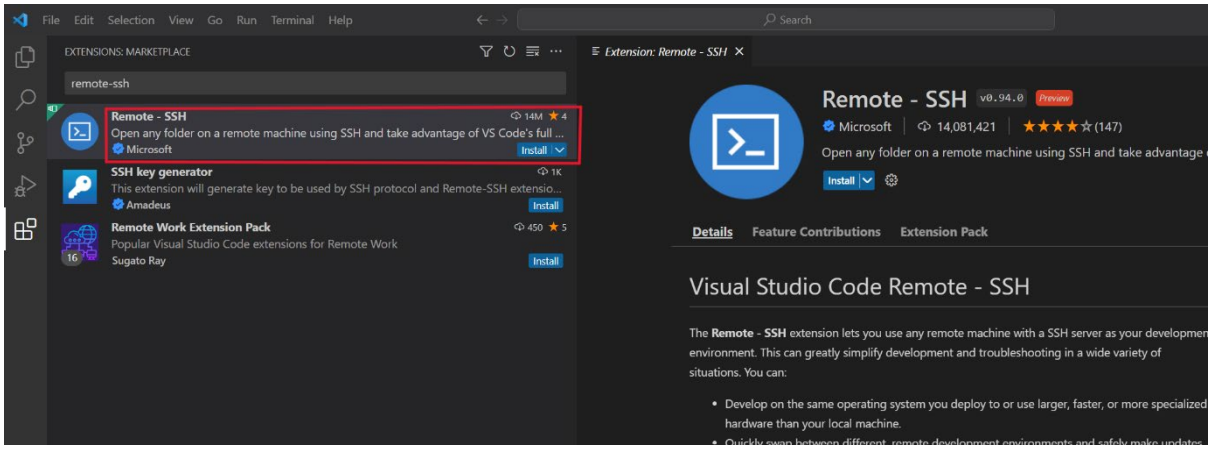

Figure 15 – Installing Remote - SSH

6.5. Return back to the Lab4 folder by clicking the "Explorer" icon on the side panel. Then, type CTRL + SHIFT + P. Select SFTP: Config (Figure 16).

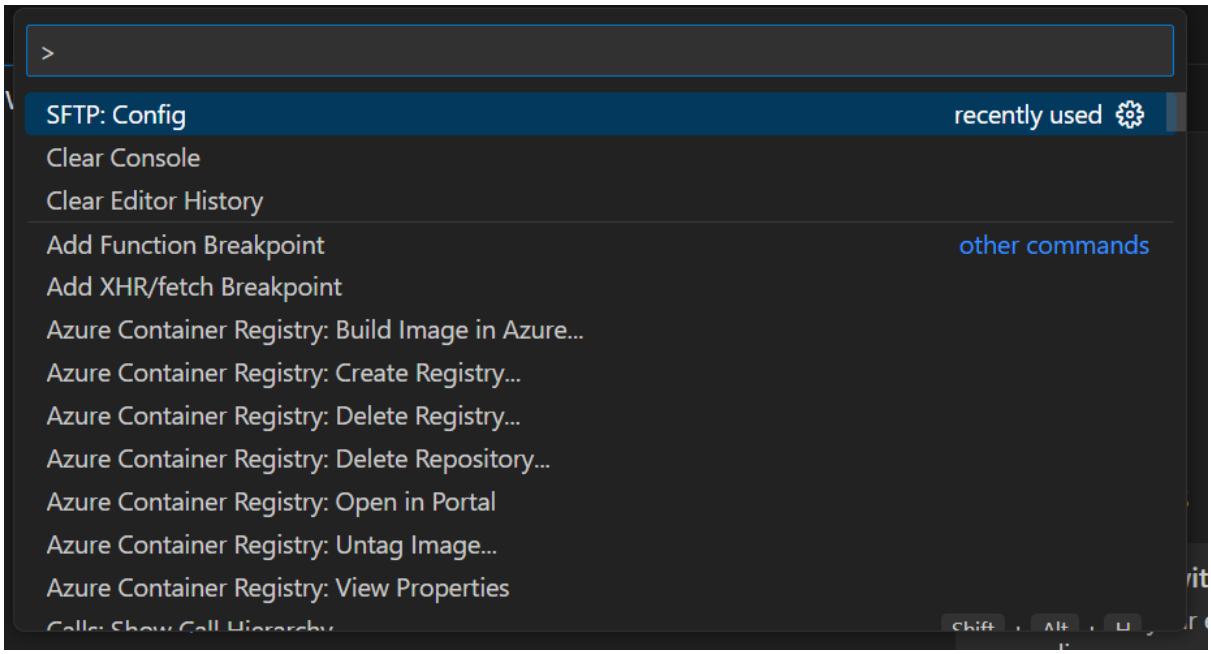

Figure 16 – Opening SFTP File

6.6. Configure the settings as shown below (Figure 17) and save (keyboard shortcut : ctrl-s). Edit the “name”, “host”, “username” and “remotePath” as shown below. You can change the remotePath to the path in the Raspberry Pi where you want to upload your project code. Do note that the folder path that you set must be present in the Raspberry Pi.

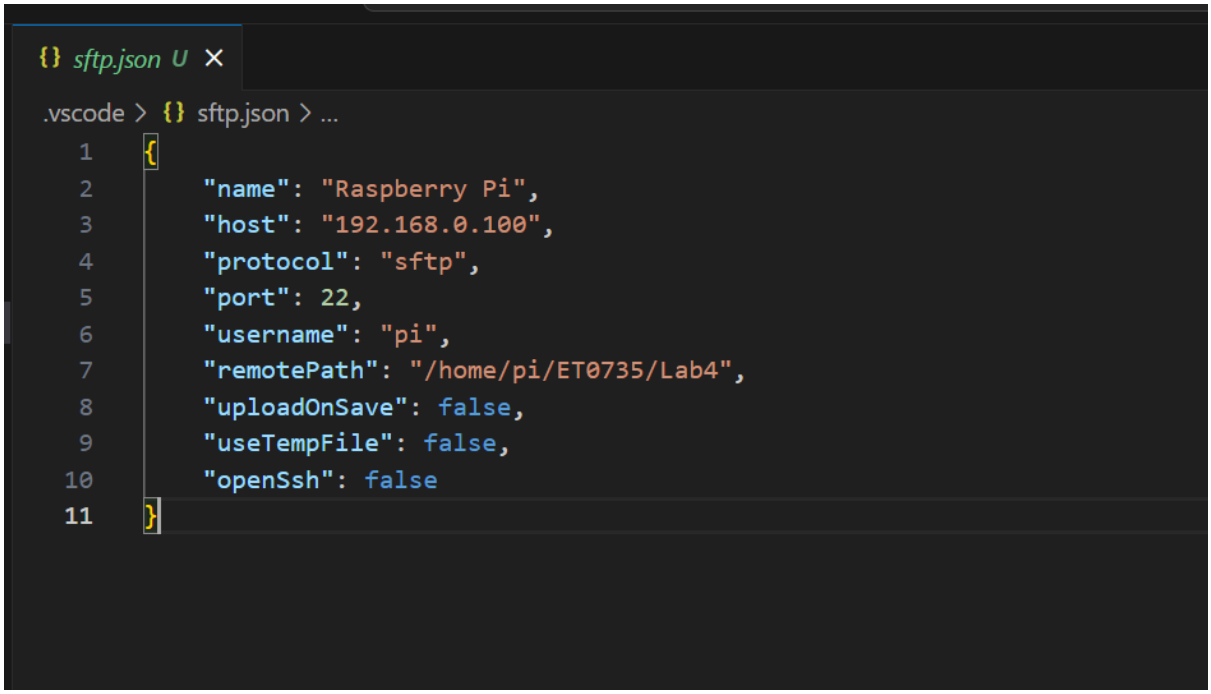

Figure 17 – SFTP Configuration

6.7. Click on the “SFTP” Icon on the side panel.

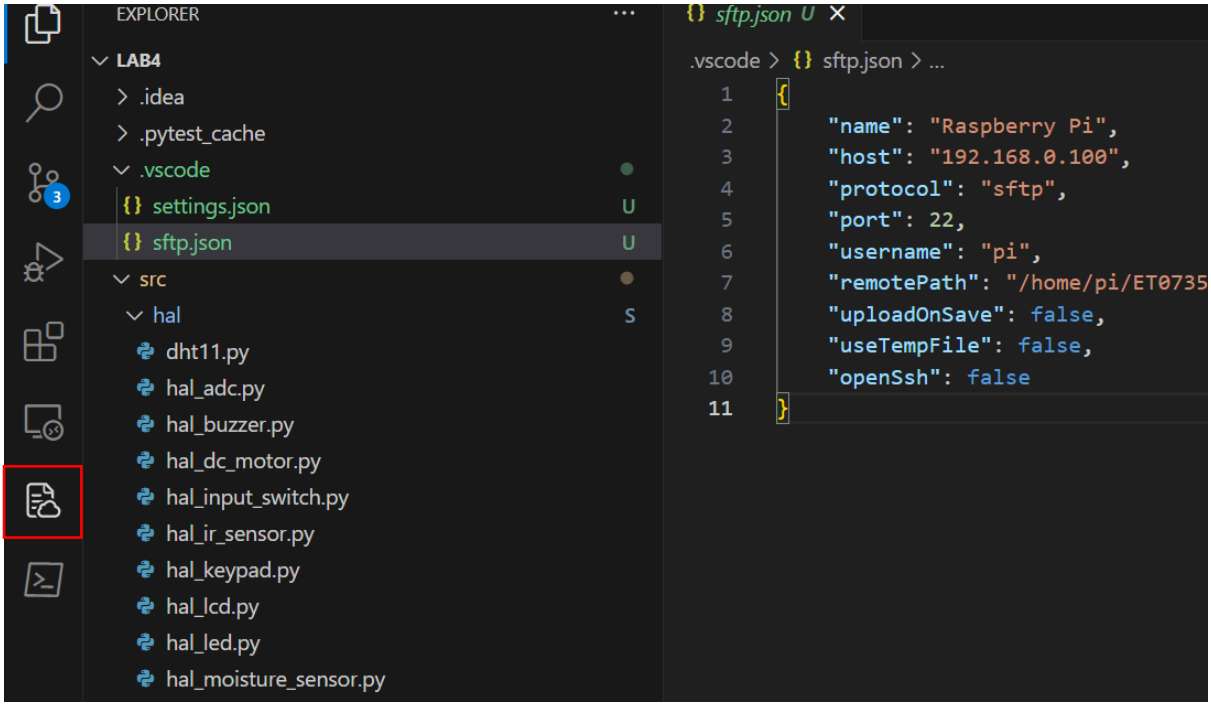

Figure 18 – SFTP Icon

6.8. In the SFTP Explorer, you should see “Raspberry Pi”, the name that was set in your SFTP Configuration file previously. If you do not see it, click on the reload icon on the top right of the SFTP Explorer.

6.9. Click on the “Raspberry Pi” folder to connect to it. Enter the password “password” for the Raspberry Pi board and press enter.

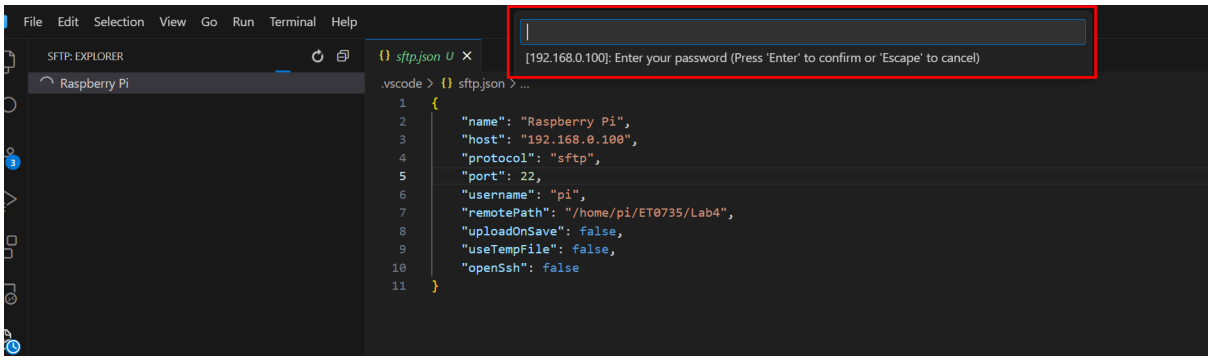

Figure 19 – SFTP Connection

6.10. Once connected, head back to the project folder by pressing on “Explorer” icon on the side panel. Select the folder that you want to upload to the Raspberry Pi. In this case the “src” folder is selected. Right click on “src” and click on “Upload Folder” to upload it to the Raspberry Pi (Figure 20).

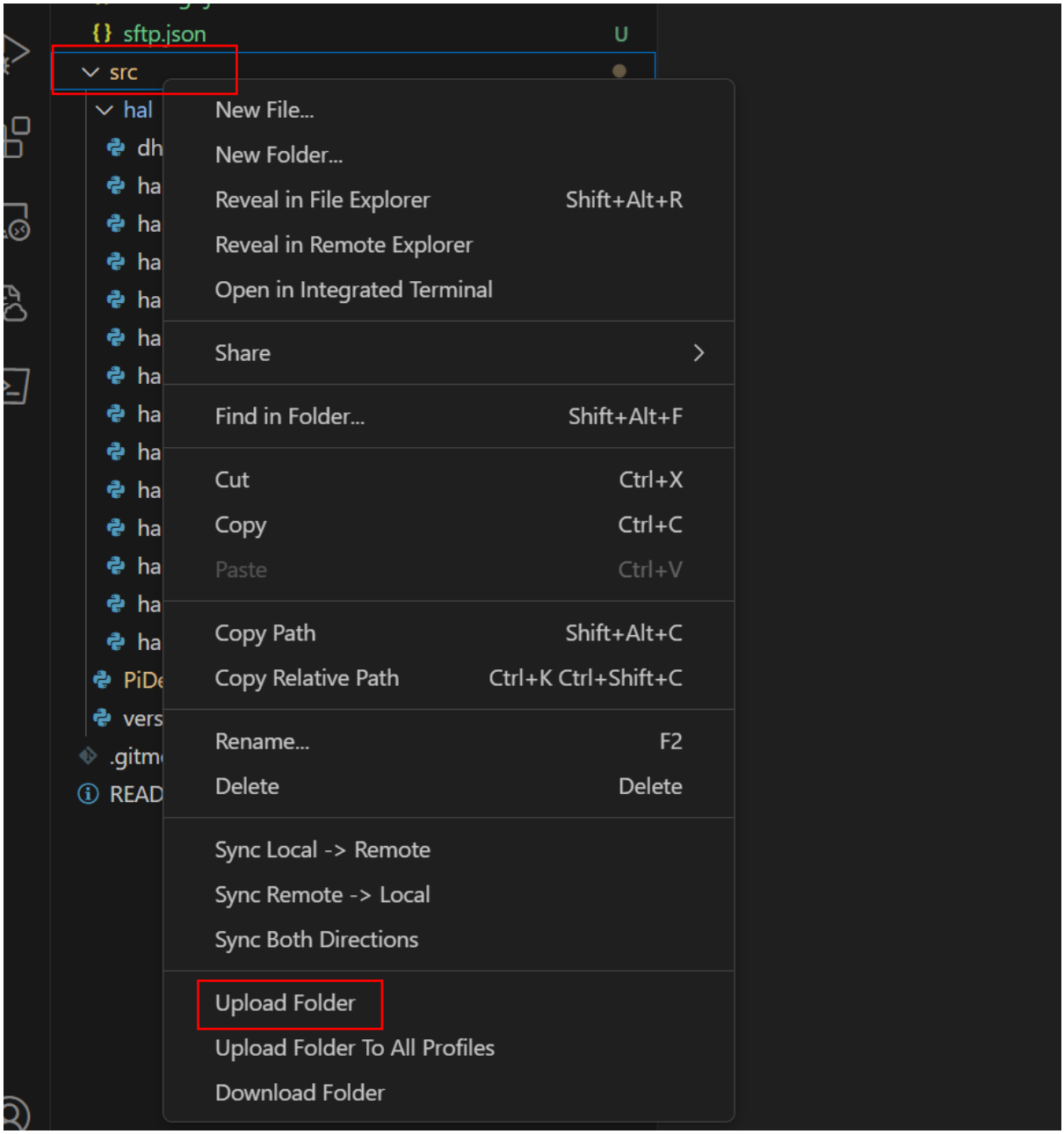

Figure 20 – Uploading folder to Raspberry Pi

6.11. Return to the SFTP Explorer by clicking on the “SFTP” icon on the side panel. Click on the reload icon. The “src” folder should now be visible under the “Raspberry Pi” folder. The folder has been uploaded from the laptop to Raspberry Pi. You can now access the Raspberry Pi from VNC Viewer to run the Python files.

7 Running Python Files directly from VS Code

You have learned how to upload files from your laptop to Raspberry Pi and you can run the python codes in your Raspberry Pi through the VNC interface. However, there is a way to run the files straight from VS Code without the use of VNC Viewer.

7.1. Given that you have installed Remote – SSH extension in VS Code, there will be a "Remote Explorer" icon on the side panel. Click on it and click on the settings icon.

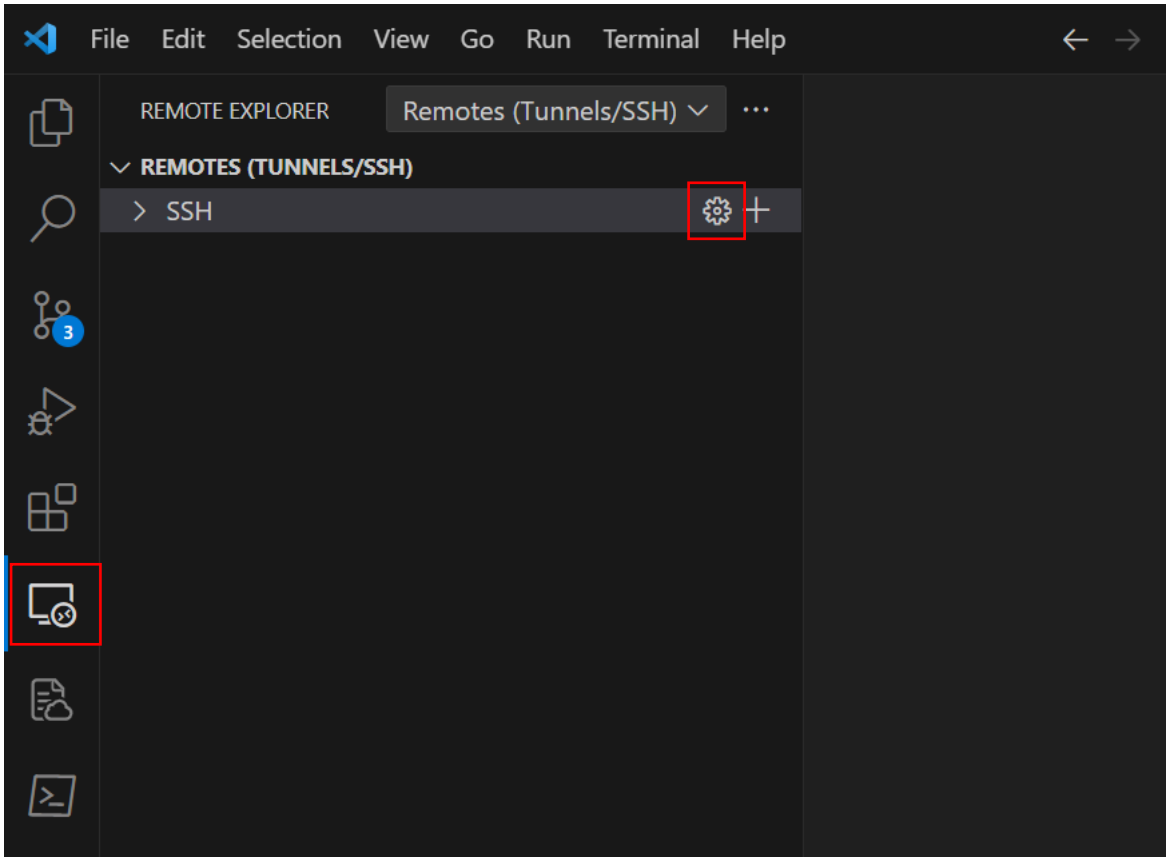

Figure 21 – Remote Explorer

7.2. Select the \.ssh\config file.

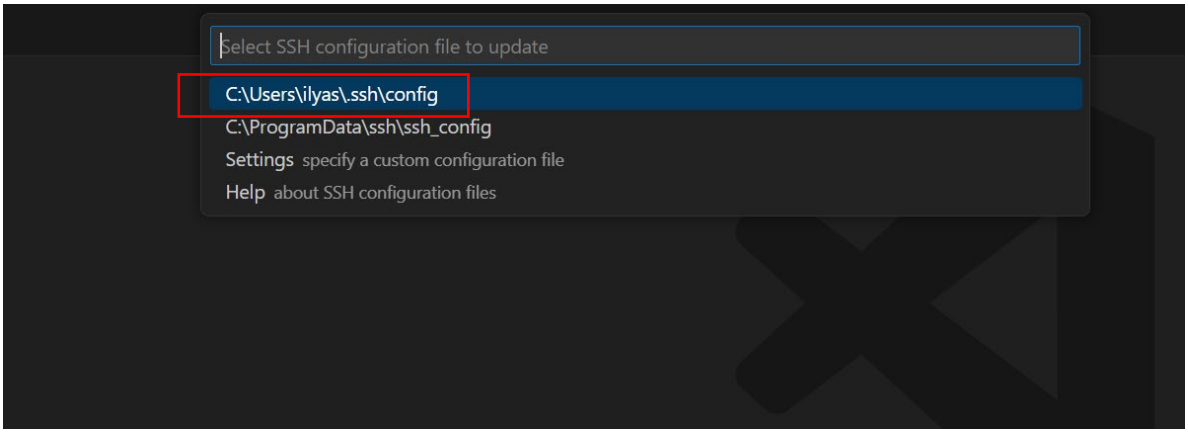

Figure 22 – VS Code Deployment Configuration

7.3. Edit the config file and add the Raspberry Pi with the correct IP Address and User, as shown below (Figure 23) and save.

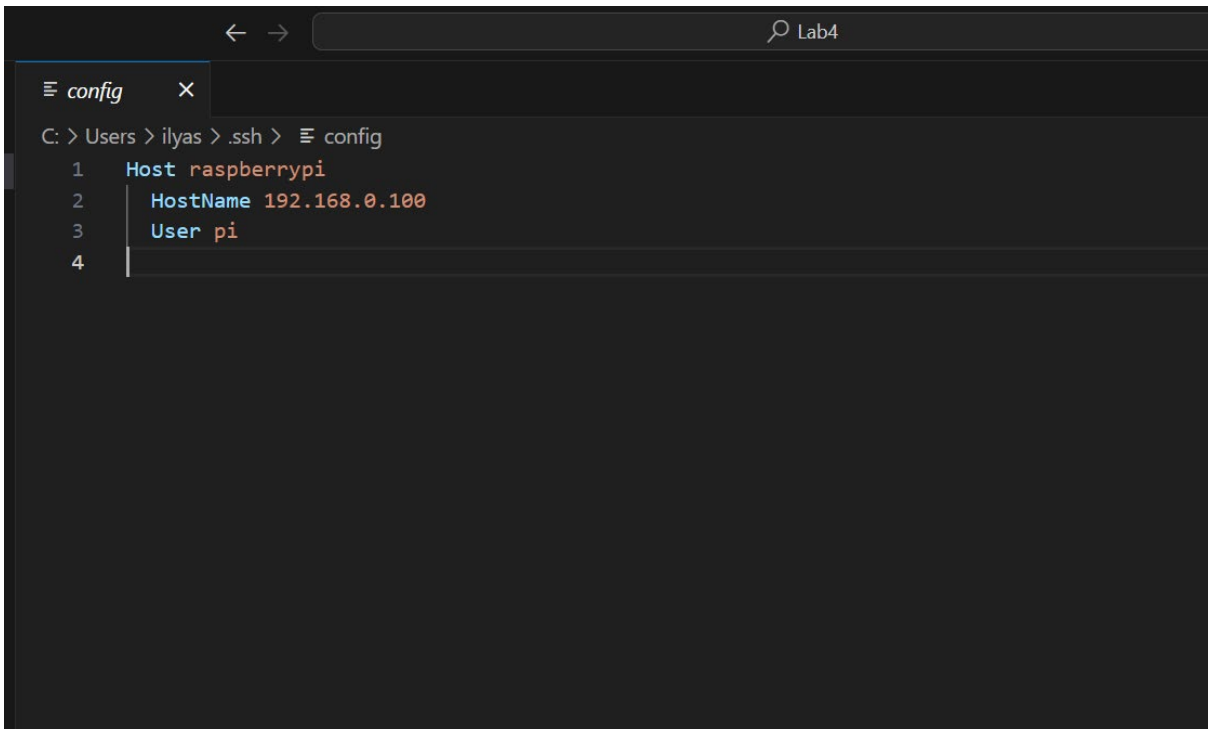

Figure 23 – Configuration Settings for Raspberry Pi

7.4. Head back to "Remote Explorer" and expand the SSH remote. The raspberrypi SSH config should be displayed. You have two choices to open the SSH session: you can either click on the arrow icon to open the SSH session in the current window, or you can click on the folder icon to open the SSH session in a new window (more commonly used).

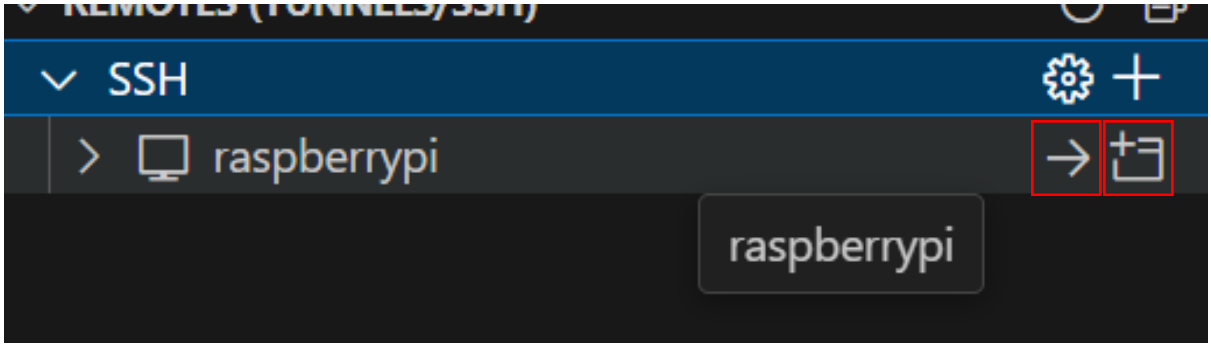

Figure 24 – Remote SSH Tunnel

7.5. If the raspberrypi config is correct, you will be prompted to choose the platform of the remote host. Choose Linux. You will then be prompted to enter the password for the Raspberry Pi. Enter the password when requested. If it is your first time connecting to the Raspberry Pi, the VS Code Server will be downloaded. This may take a while.

7.6. Once connected, you should see the SSH connection being successful with the blue icon showing up in the bottom left corner.

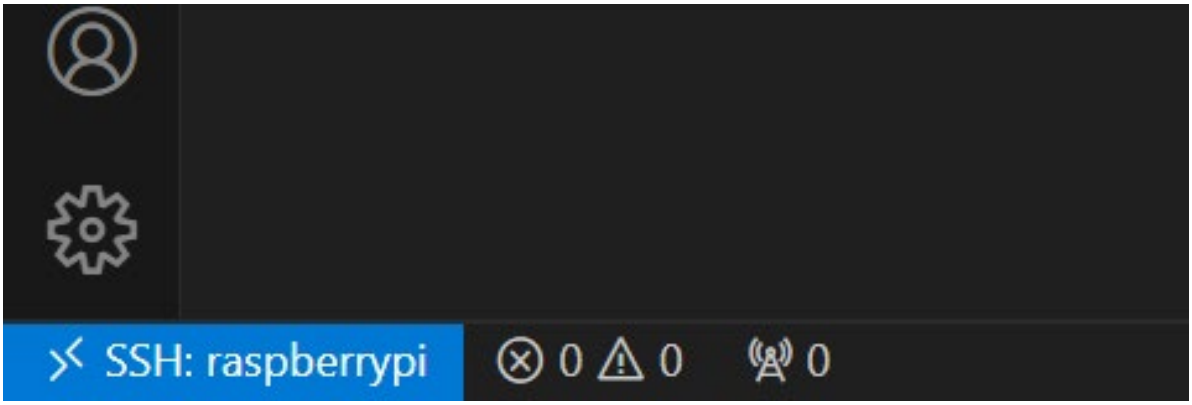

Figure 25 – SSH Connection Successful

7.7. Click on the “Explorer” and select “Open Folder”.

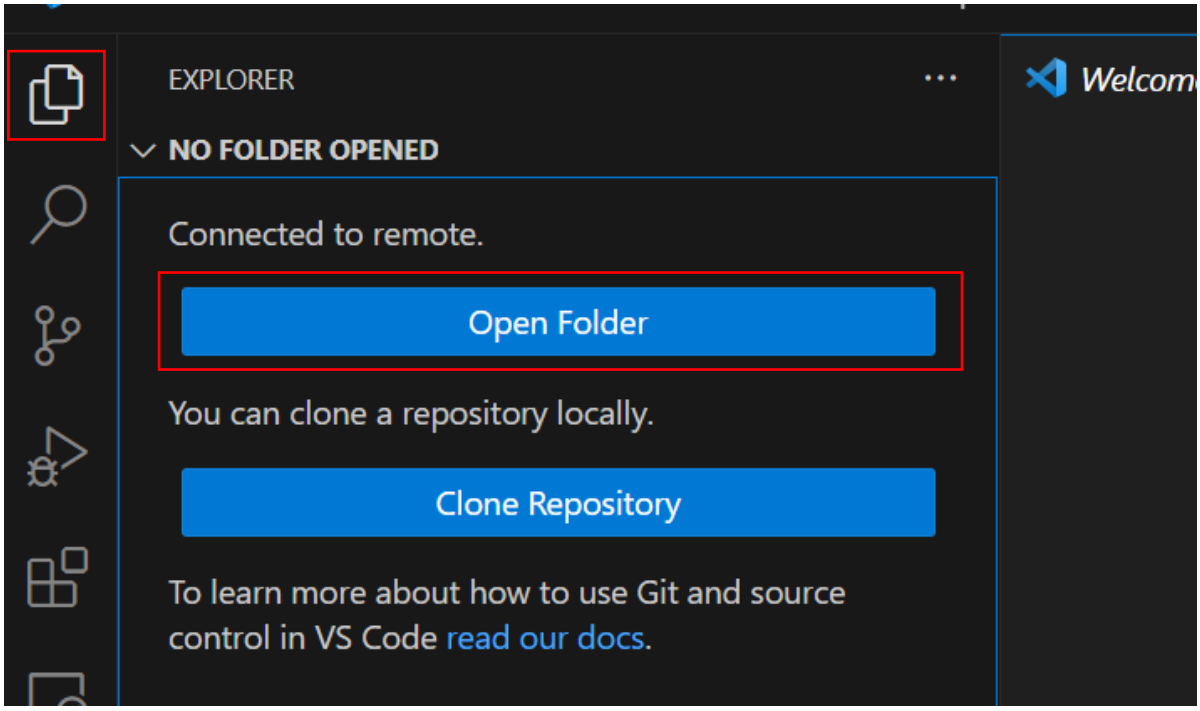

Figure 26 – Open RPI Folder through VS Code

7.8. Choose the file path of your project folder in the prompt and press “OK”. You will be prompted to enter the password.

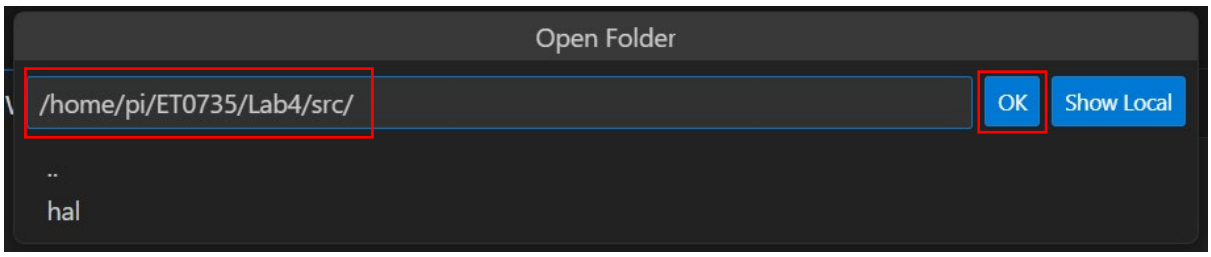

Figure 27 – Select File Path of Project

7.9. The project folder in the Raspberry Pi will be shown.

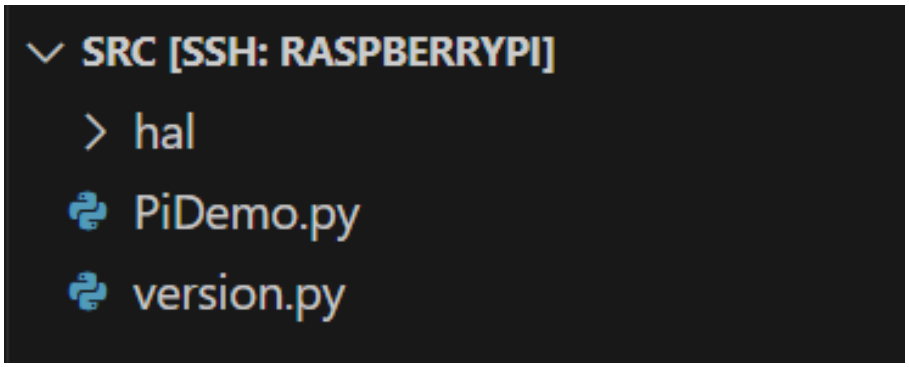

Figure 28 – Raspberry Pi Project Folder

7.10. Click on “Terminal” and select “New Terminal” to open a terminal for the Raspberry Pi. A Raspberry Pi terminal will be opened in VS Code.

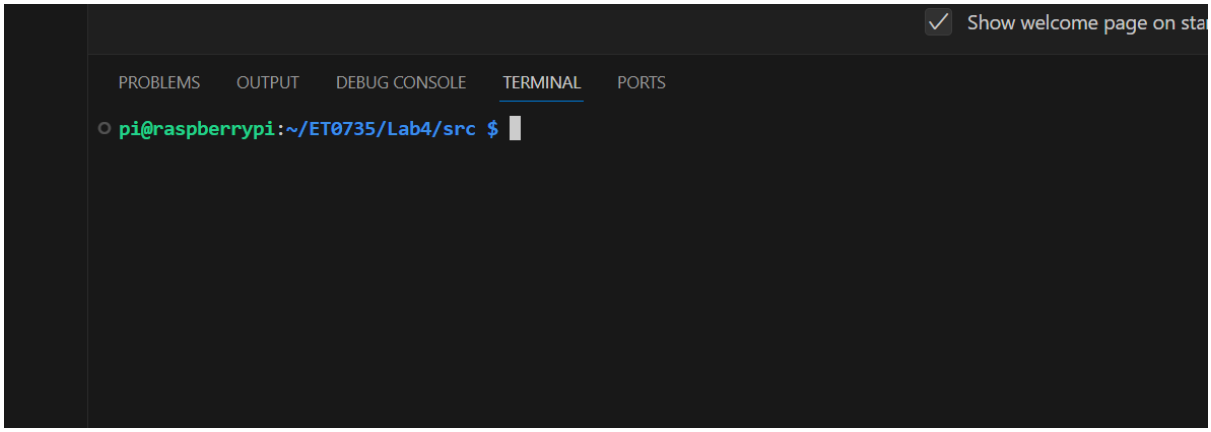

Figure 29 – Open Raspberry Pi Terminal

7.11. You can run the Python files from the terminal now. You should be able to remotely run your code from VS Code.

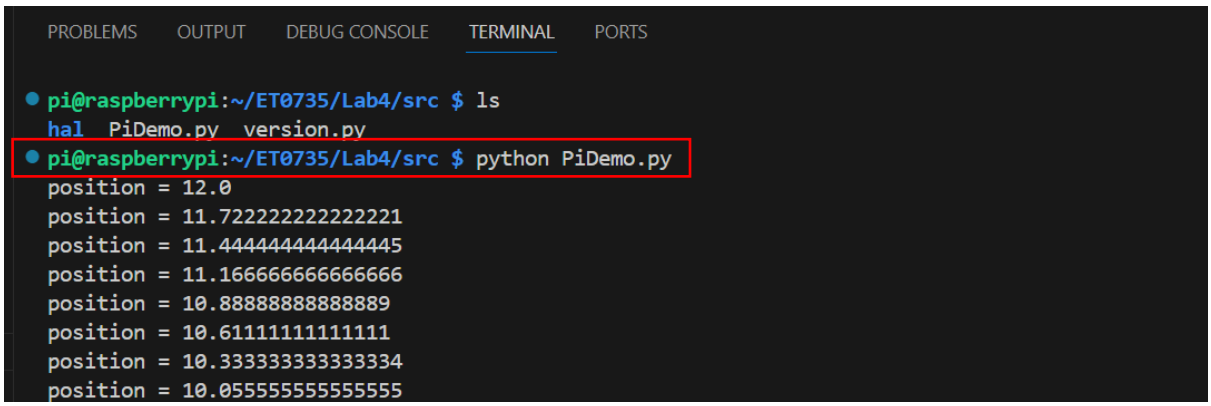

Figure 30 – Running Python Files within VS Code Terminal

7.12. Alternatively, you can run the Python files without using the terminal. Click on PiDemo.py and select "Run Without Debugging" under Run.

7.13. There is a caveat with this method however, as you will be required to install a Python Debugger in the Remote SSH Client (Figure 31). This installation may take longer compared to using the terminal directly.

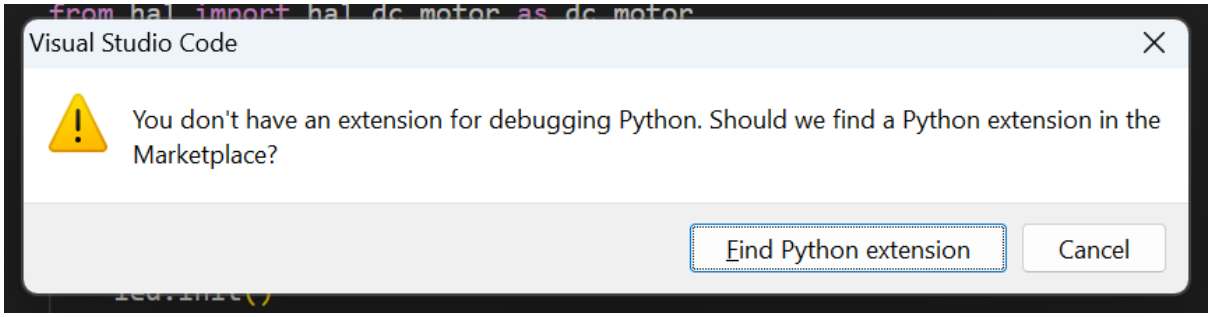

Figure 31 – Python Extension Required

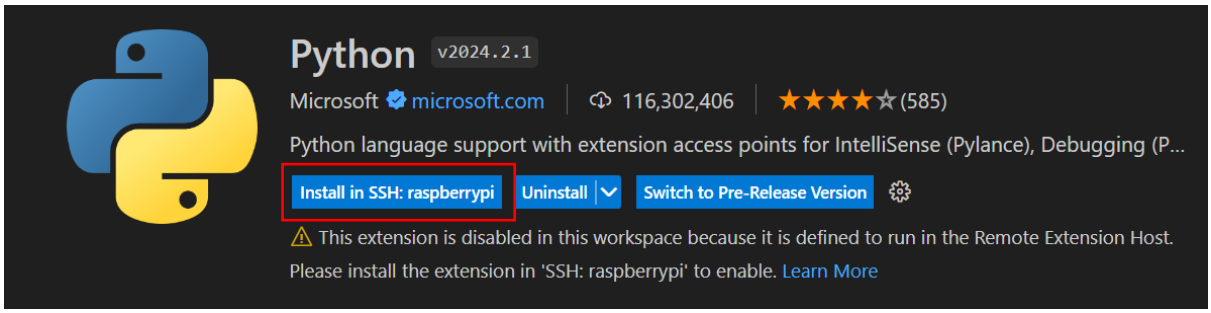

Figure 32 – Installing Python Debugger in SSH

7.14. Finally, after running the the main Python File "PiDemo.py", check that the following Raspberry Pi peripherals are activated:

- LCD shows the message "DevOps for AIoT" and the Software version number "v1.01"
- Buzzer beeps
- LED blinks
- Servo Motor rotates
- DC motor is switched on for 2 second

Exercise 2

Now that we have setup VS Code to deploy and execute Python application remotely on the Raspberry Pi, we can now start to implement some basic Python applications on the Raspberry Pi.

In this exercise, we will implement a simple Python application to control the blinking rate of the LED on the Raspberry Pi development board.

For additional documentation on the Raspberry Pi Development board hardware pin assignments, HAL sample code, etc, please refer to the documentation at the Github page below,

https://github.com/ET0735-DevOps-AIoT/raspberry-pi-dev-board/wiki

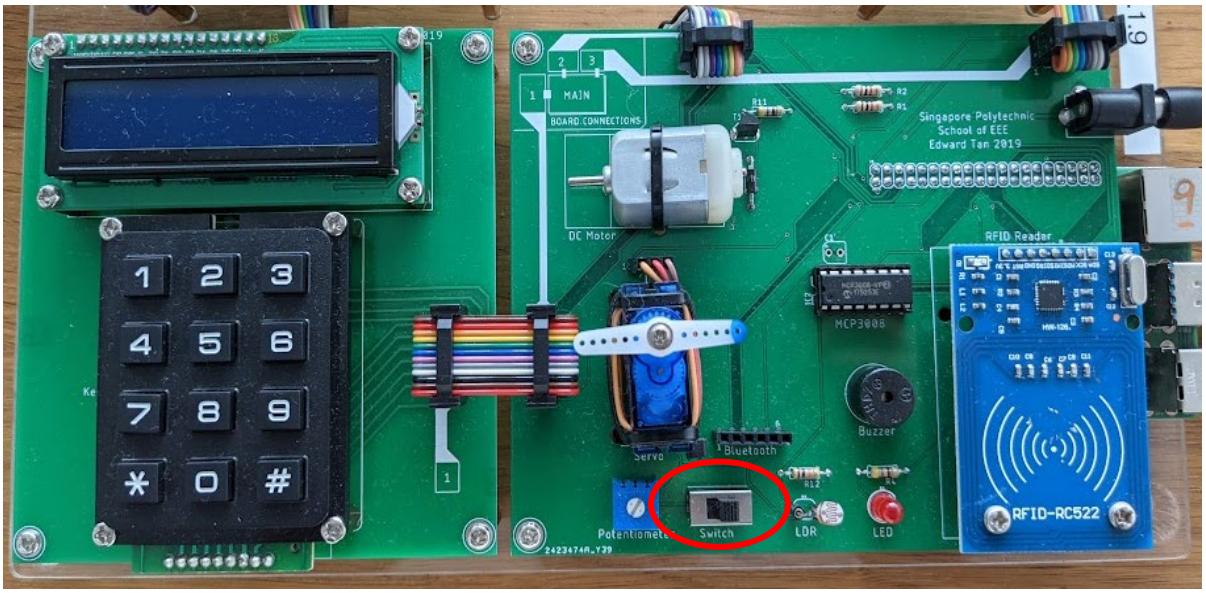

Figure 26

(a) In the existing VS Code project for Lab 4, create a new Python file "PiLedTest.py". Under the src directory by right-click the src folder in the VS Code project directory, and select "python file", enter the file name "PiLedTest.py" and press <ENTER>

(b) Using the HAL Python file "hal_input_switch.py", implement Python code that will blink the LED at a frequency of 5 Hz if the switch highlighted in Figure 26 above is moved to the **left position**.

If the switch is changed to the **right position**, then the LED should be turned OFF.

(c) Update your implementation to increase the blinking rate of the LED to 10Hz if the switch is changed to the **right position**. The LED should continue to blink at 5 Hz if the switch is in the **left position**.

(d) Extend your implementation such that when the LED is changed to the **right position** the LED for will blink at 10 Hz for 5 seconds, after which the LED should be turned OFF.

Exercise 3

To complete the lab exercises, create a new Github repository for Lab 4.

(a) Add commit and push all changes to Github

(b) Create a new Github release "v1.1" in your Lab 4 Github repository with the changes pushed to Github for Exercise 2