Lab 1: Debugging a Raspberry Pi Internet of Things Flask Application

Author	Dave Glover, Microsoft Cloud Developer Advocate
Platforms	Linux, macOS, Windows, Raspbian Buster
Tools	Visual Studio Code Insiders Edition
Hardware	Raspberry Pi 4. 4GB model required for 20 Users. Raspberry Pi Sense HAT, Optional: Raspberry Pi case, active cooling fan
USB3 SSD Drive	To support up to 20 users per Raspberry Pi you need a **fast** USB3 SSD Drive to run Raspbian Buster Linux on. A 120 USB3 SSD drive will be more than sufficient. These are readily available from online stores.
Language	Python
Date	As of August, 2019

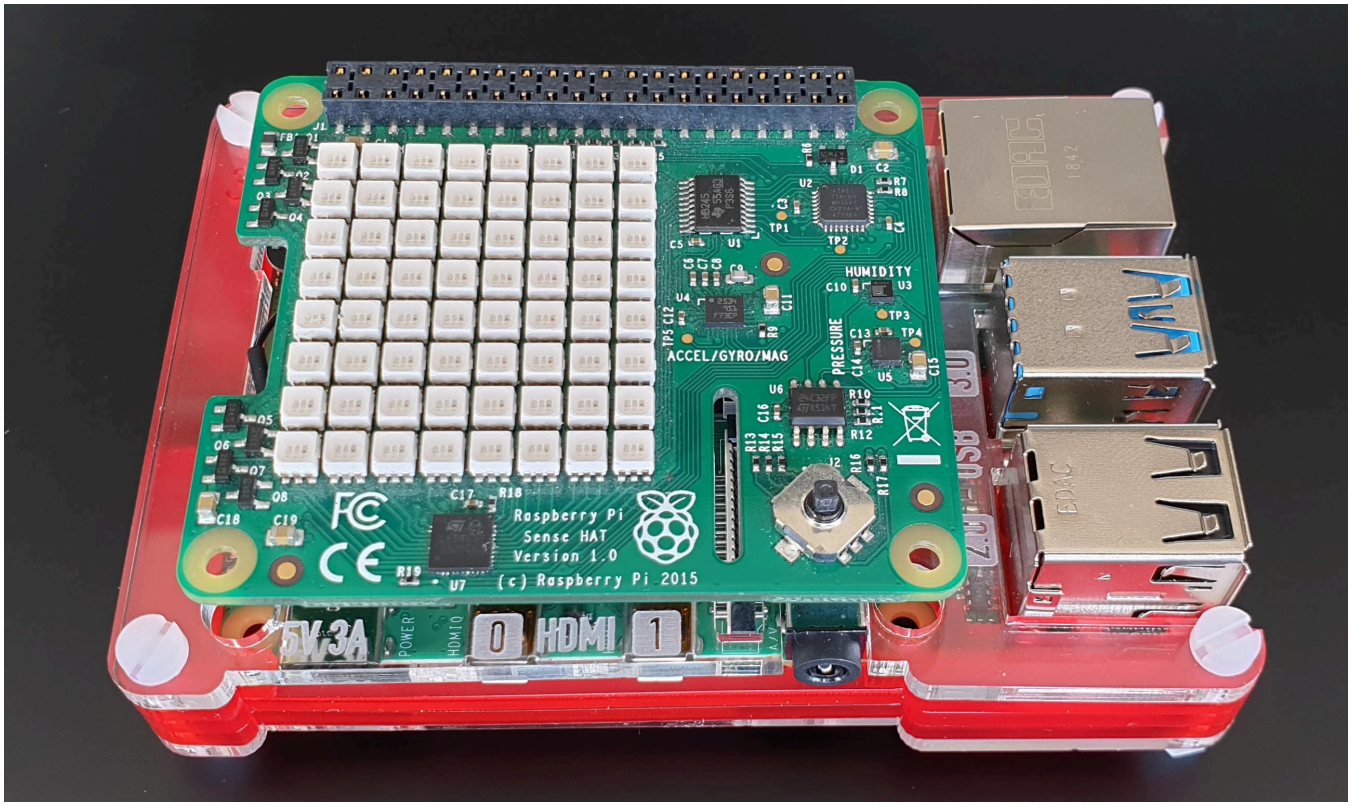
Follow me on Twitter @dglover

PDF Lab Guide

You may find it easier to download and follow the PDF version of the Debugging Raspberry Pi Internet of Things Flask App hands-on lab guide.

Introduction

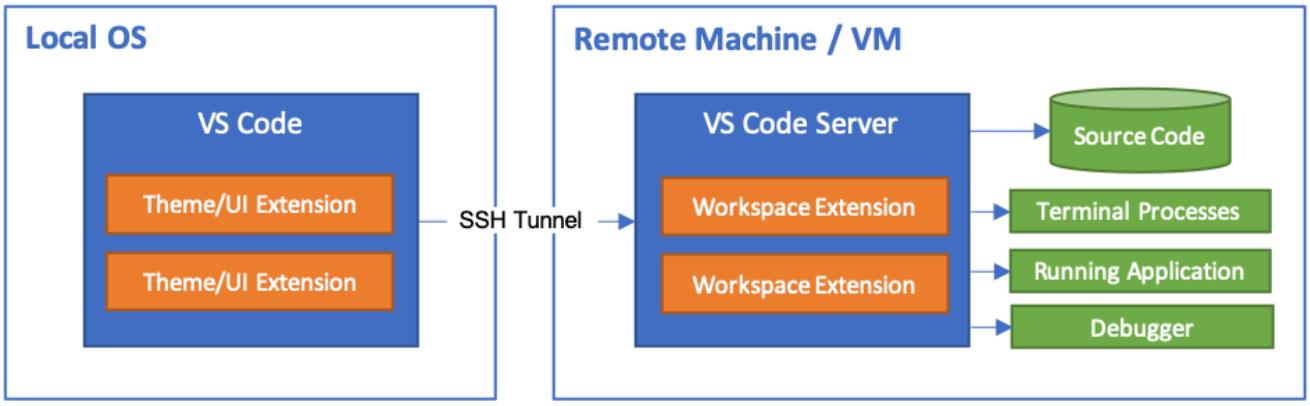
In this hands-on lab, you will learn how to create and debug a Python web application on a Raspberry Pi with Visual Studio Code and the Remote SSH extension. The web app will read the temperature, humidity, and air pressure telemetry from a sensor connected to the Raspberry Pi.



Remote Development using SSH

The Visual Studio Code Remote - SSH extension allows you to open a remote folder on any remote machine, virtual machine, or container with a running SSH server and take full advantage of Visual Studio Code's feature set. Once connected to a server, you can interact with files and folders anywhere on the remote filesystem.

No source code needs to be on your local machine to gain these benefits since the extension runs commands and other extensions directly on the remote machine.



Software Installation

This hands-on lab uses Visual Studio Code. Visual Studio Code is a code editor and is one of the most popular **Open Source** projects on GitHub. It runs on Linux, macOS, and Windows.

1. Visual Studio Code Insiders Edition
   As at August 2019, **Visual Studio Code Insiders Edition** is required as it has early support for Raspberry Pi and Remote Development over SSH.
2. Remote - SSH Visual Studio Code Extension

For information on contributing or submitting issues see the Visual Studio GitHub Repository. Visual Studio Code documentation is also Open Source, and you can contribute or submit issues from the Visual Studio Documentation GitHub Repository.

Raspberry Pi Hardware

If you are attending a workshop, then you can use a shared network-connected Raspberry Pi. You can also use your own network-connected Raspberry Pi for this hands-on lab.

You will need the following information from the lab instructor.

1. The **Network IP Address** of the Raspberry Pi
2. Your assigned **login name** and **password**.

SSH Authentication with private/public keys

Setting up a public/private key pair for SSH authentication is a secure and fast way to authenticate from your computer to the Raspberry Pi. This is needed for this hands-on lab.

SSH for Linux and macOS

From a Linux or macOS **Terminal Console** run the following commands:

1. Create your key. This is typically a one-time operation. **Take the default options**.

   ```
   ssh-keygen -t rsa -b 4096 -f ~/.ssh/id_rsa_python_lab
   ```

2. Copy the public key to the Raspberry Pi.

   ```
   ssh-copy-id -i ~/.ssh/id_rsa_python_lab <login@Raspberry IP Address>
   ```

   For example:

   ```
   ssh-copy-id -i ~/.ssh/id_rsa_python_lab dev99@192.168.1.200
   ```

3. Test the SSH Authentication Key

   ```
   ssh -i ~/.ssh/id_rsa_python_lab <login@Raspberry IP Address>
   ```

   A new SSH session will start. You should now be connected to the Raspberry Pi **without** being prompted for the password.
4. Close the SSH session. In the SSH terminal, type exit, followed by ENTER.

SSH for Windows 10 (1809+) Users with PowerShell

1. Start PowerShell as Administrator and install OpenSSH.Client

```
    Add-WindowsCapability -Online -Name OpenSSH.Client
```

2. **Exit** PowerShell
3. Restart PowerShell (**NOT** as Administrator)
4. Create an SSH Key

```
ssh-keygen -t rsa -f $env:userprofile\.ssh\id_rsa_python_lab
```

5. Copy SSH Key to Raspberry Pi

```
cat $env:userprofile\.ssh\id_rsa_python_lab.pub | ssh `
<login@Raspberry IP Address> `
"mkdir -p ~/.ssh; cat >> ~/.ssh/authorized_keys"
```

6. Test the SSH Authentication Key

```
ssh -i $env:userprofile\.ssh\id_rsa_python_lab <login@Raspberry IP Address>
```

   A new SSH session will start. You should now be connected to the Raspberry Pi **without** being prompted for the password.
7. Close the SSH session. In the SSH terminal, type exit, followed by ENTER.

SSH for earlier versions of Windows

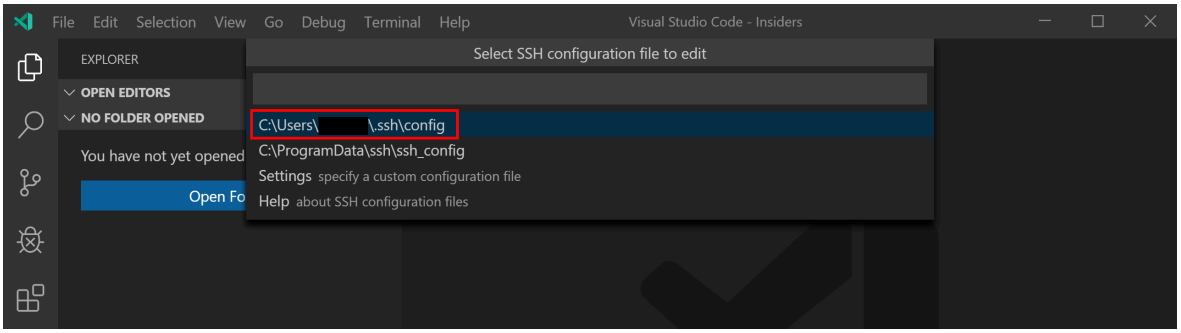
SSH for earlier versions of Windows

Trouble Shooting SSH Client Installation

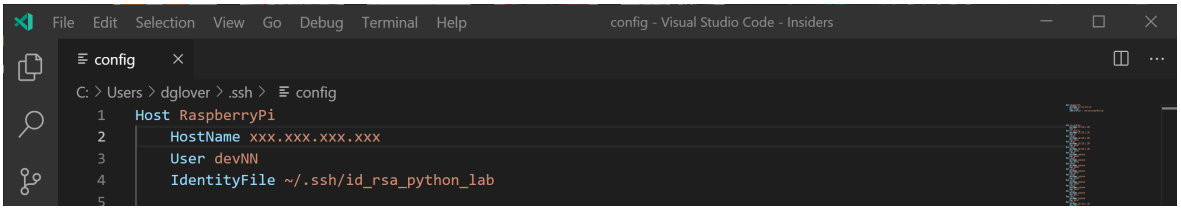
- Remote Development using SSH
- Installing a supported SSH client

Configure Visual Studio Code Remote SSH Development

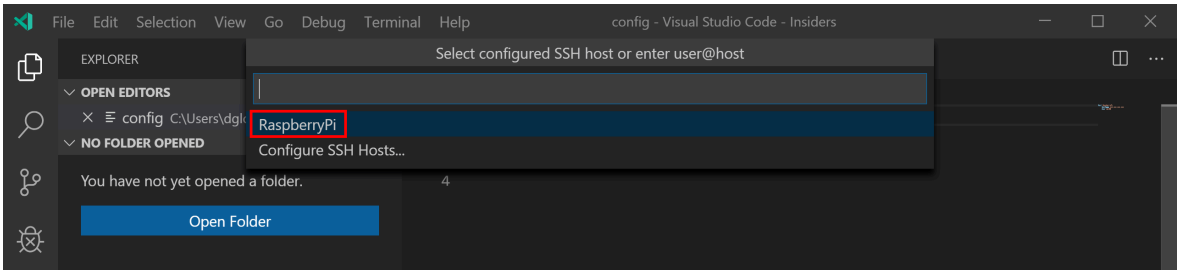
1. Start Visual Studio Code Insiders Edition
2. Press F1 to open the Command Palette, type **ssh config** and select **Remote-SSH: Open Configuration**
3. Select the user .ssh config file



4. Set the SSH connection configuration as follows:
    - **Host**: Set to **RaspberryPi**
    - **HostName**: The Raspberry Pi **IP Address**
    - **User**: Your **login name**
    - **IdentityFile**: Set to **~/.ssh/id_rsa_python_lab**.
    - Save these changes (Ctrl+S).



5. Press **F1** to open the Command Palette, type **ssh connect** and select **Remote-SSH: Connect to Host**
6. Select the host **RaspberryPi** configuration



It will take a moment to connect to the Raspberry Pi.

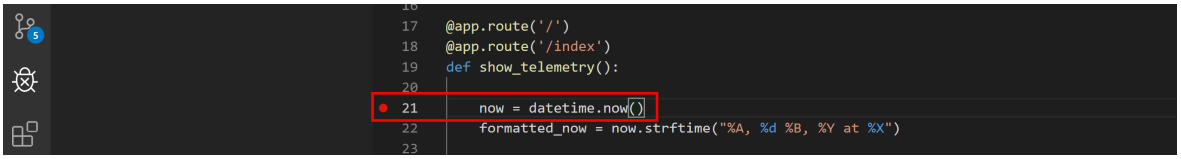
Open the Lab 1 SSH Debug Project

From **Visual Studio Code**, select **File** from the main menu, then **Open Folder**. Navigate to and open the **github/Lab1-ssh-debug** folder.

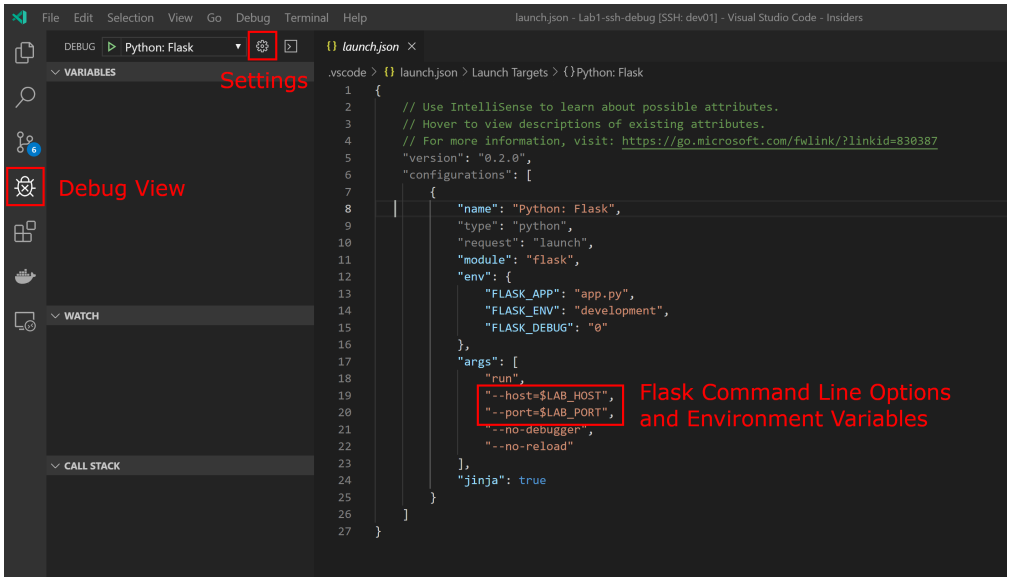
1. From Visual Studio Code: File -> Open Folder
2. Navigate to **github/Lab1-ssh-debug** directory
3. Open the **app.py** file and review the contents
4. Set a breakpoint at the first line of code in the **show_telemetry** function (**now =**

datetime.now()) by doing any one of the following:

- With the cursor on that line, press F9, or,
- With the cursor on that line, select the Debug > Toggle Breakpoint menu command, or, click directly in the margin to the left of the line number (a faded red dot appears when hovering there). The breakpoint appears as a red dot in the left margin:

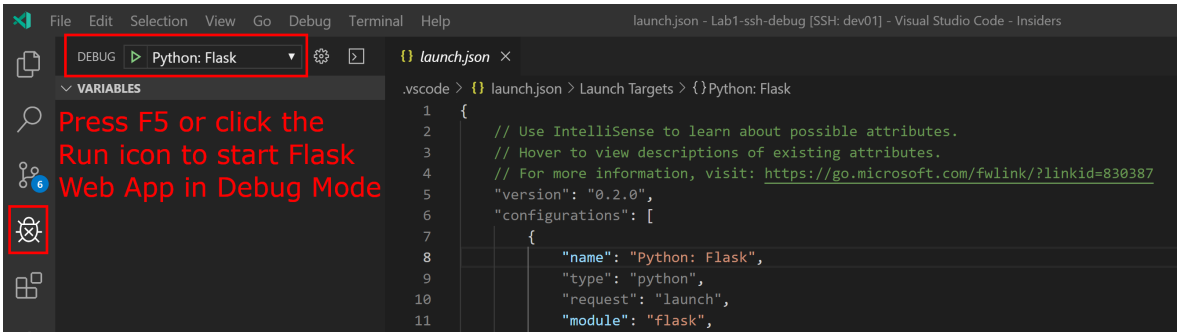


5. Review the **debug** options.
    1. Switch to Debug view in Visual Studio Code (using the left-side activity bar).

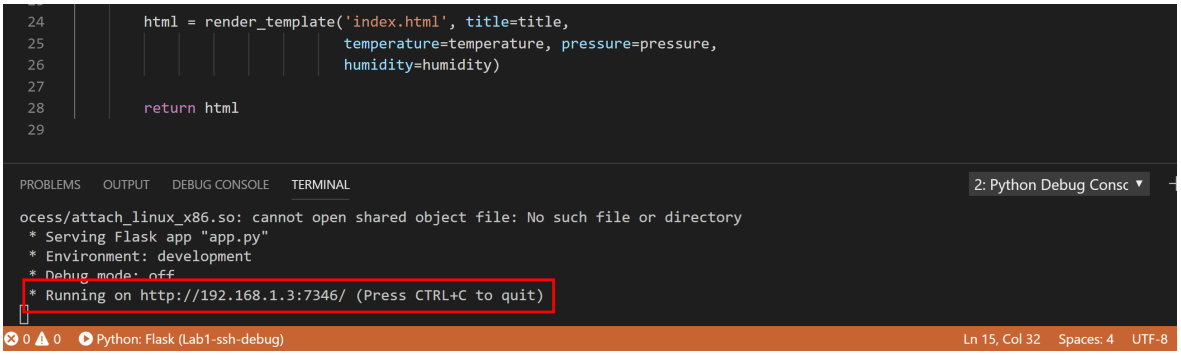


    2. Click the **Settings** button which will open the **launch.json** file.
    3. The **launch.json** file defines how the Flask app will start, and what Flask Command Line parameters to pass at startup.

       There are two environment variables used in the launch.json file. These are **LAB_HOST** (which is the IP Address of the Raspberry Pi), and **LAB_PORT** (a random TCP/IP Port number between 5000 and 8000). These environment variables are set by the .bashrc script which runs when you connect to the Raspberry Pi with Visual Studio Remote SSH.
6. Press F5 (or click the Run icon) to launch the **Python: Flask** debug configuration. This will start the Web Application on the Raspberry Pi in debug mode.



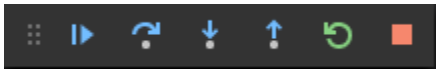
7. Ctrl+click the Flask Web link in the Visual Studio Terminal Window. This will launch your desktop Web Browser.



8. Next switch back to Visual Studio Code. The code execution has stopped at the breakpoint you set.

Debug actions

Once a debug session starts, the **Debug toolbar** will appear at the top of the editor window.



A debugging toolbar (shown above) will appear in Visual Studio Code. It has the following options:

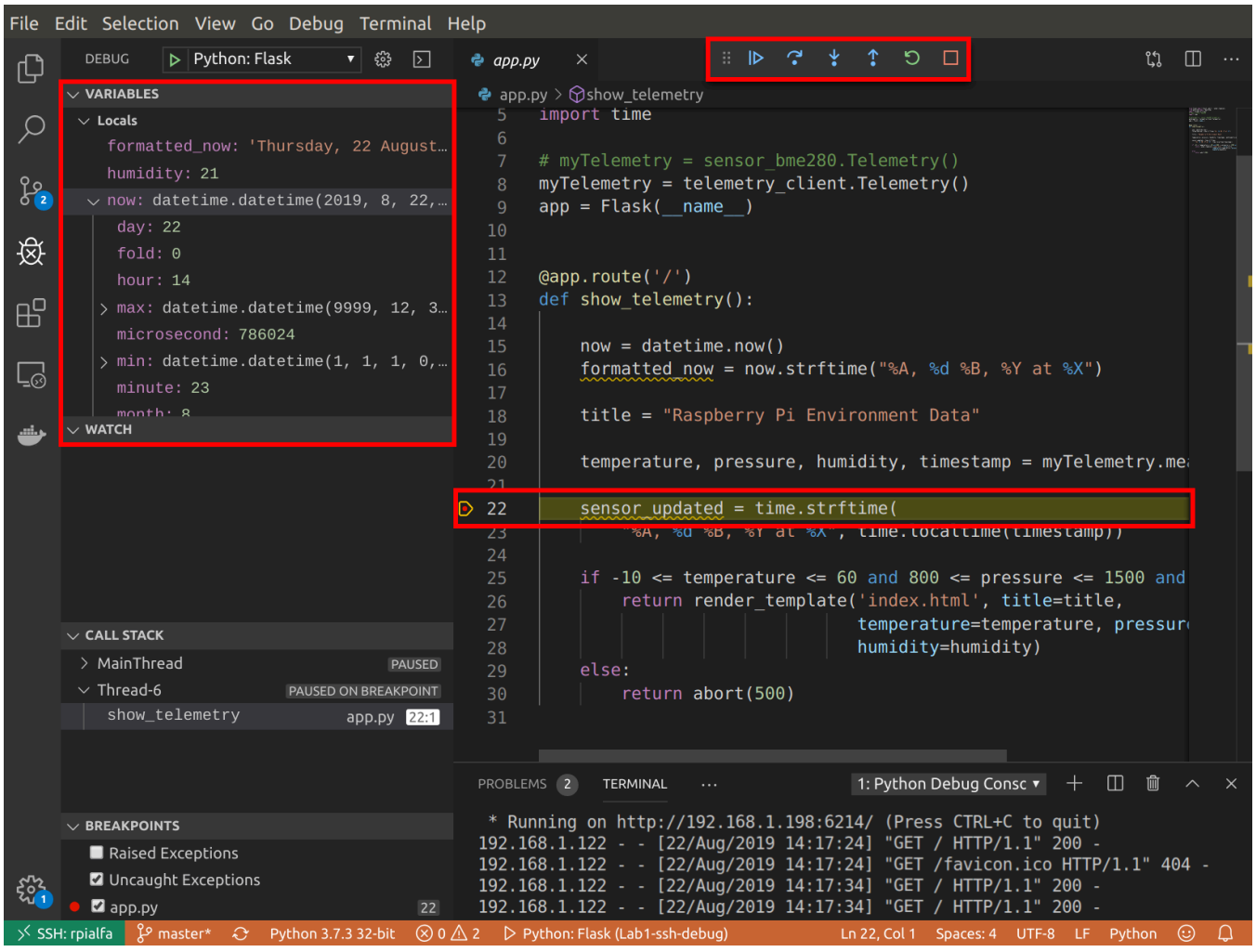
1. Pause (or Continue, F5),
2. Step Over (F10)
3. Step Into (F11),
4. Step Out (Shift+F11),
5. Restart (Ctrl+Shift+F5),
6. and Stop (Shift+F5).

Using the Debugger

Next, we are going to **Step Into** (F11) the code using the debugging toolbox. Observe the debugger steps into the **Telemetry** class and calls the **show_telemetry** method.

Variable Explorer

1. As you step through the **show_telemetry** method you will notice that Python variables are displayed in the **Variables Window**.



2. Right mouse click a variable, and you will discover you can change, copy, or watch variables. Try to change the value of a variable. **Hint**, double click on a variable value.
3. Press F5 to resume the Flask App, then switch back to your web browser and you will see the temperature, humidity, and pressure Sensor data displayed on the web

page.

Raspberry Pi Environment Data

Telemetry	Value
Temperature	31.7 C
Humidity	17 %
Pressure	995 hPa

4. Press the **Refresh** button on your web browser. The Flask app will again stop at the breakpoint in Visual Studio Code.

Experiment with Debugger Options

Things to try:

1. Review the Visual Studio Code Python Tutorial
2. Review the Python Flask tutorial
3. Review the Visual Studio Code Debugging Tutorial

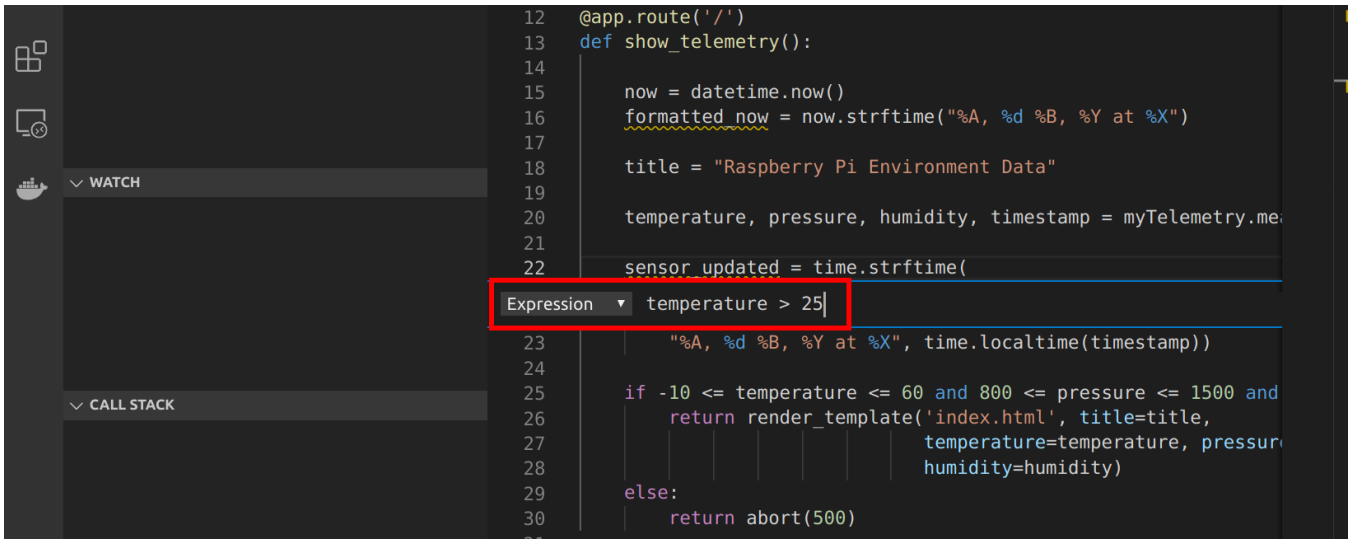
Experiment with the Variable Window

Try to change the value of a variable from the Visual Studio Code **Variable Window**.

Set a Conditional Breakpoint

Try setting a **conditional** breakpoint

Right mouse click directly in the margin to the left of the line number 22, select **Add Conditional Breakpoint...**. The breakpoint appears as a red dot with an equals sign in the middle:



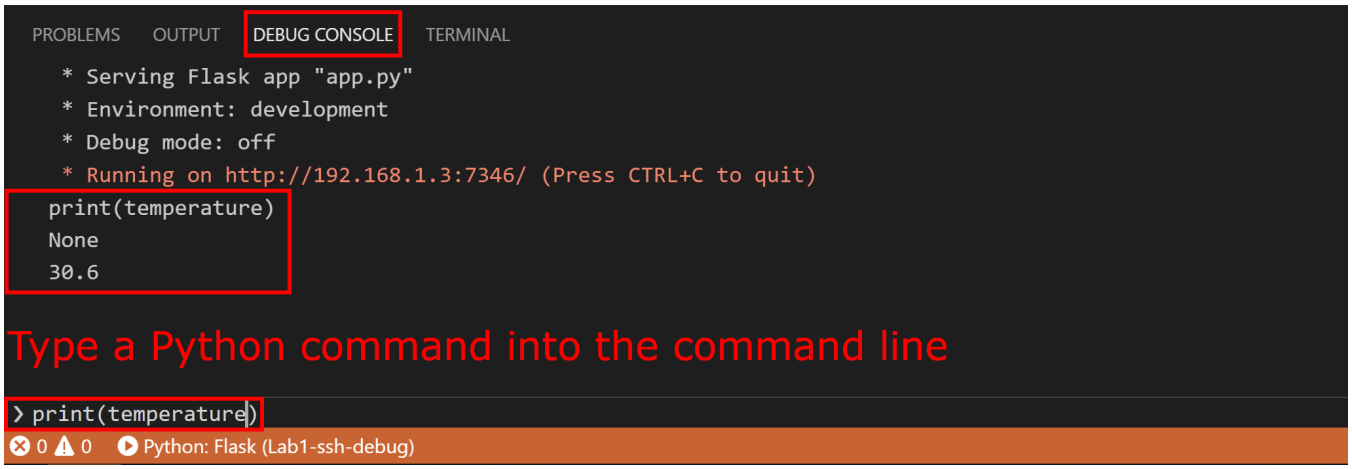
Try the Debug Console

Try the Visual Studio Code **Debug Console**. This will give you access to the Python REPL (Read, Evaluate, Print Loop), try printing or setting variables, importing libraries, etc.

```
print(temperature)

temperature = 24

import random
random.randrange(100, 1000)
```



Update the Flask Template

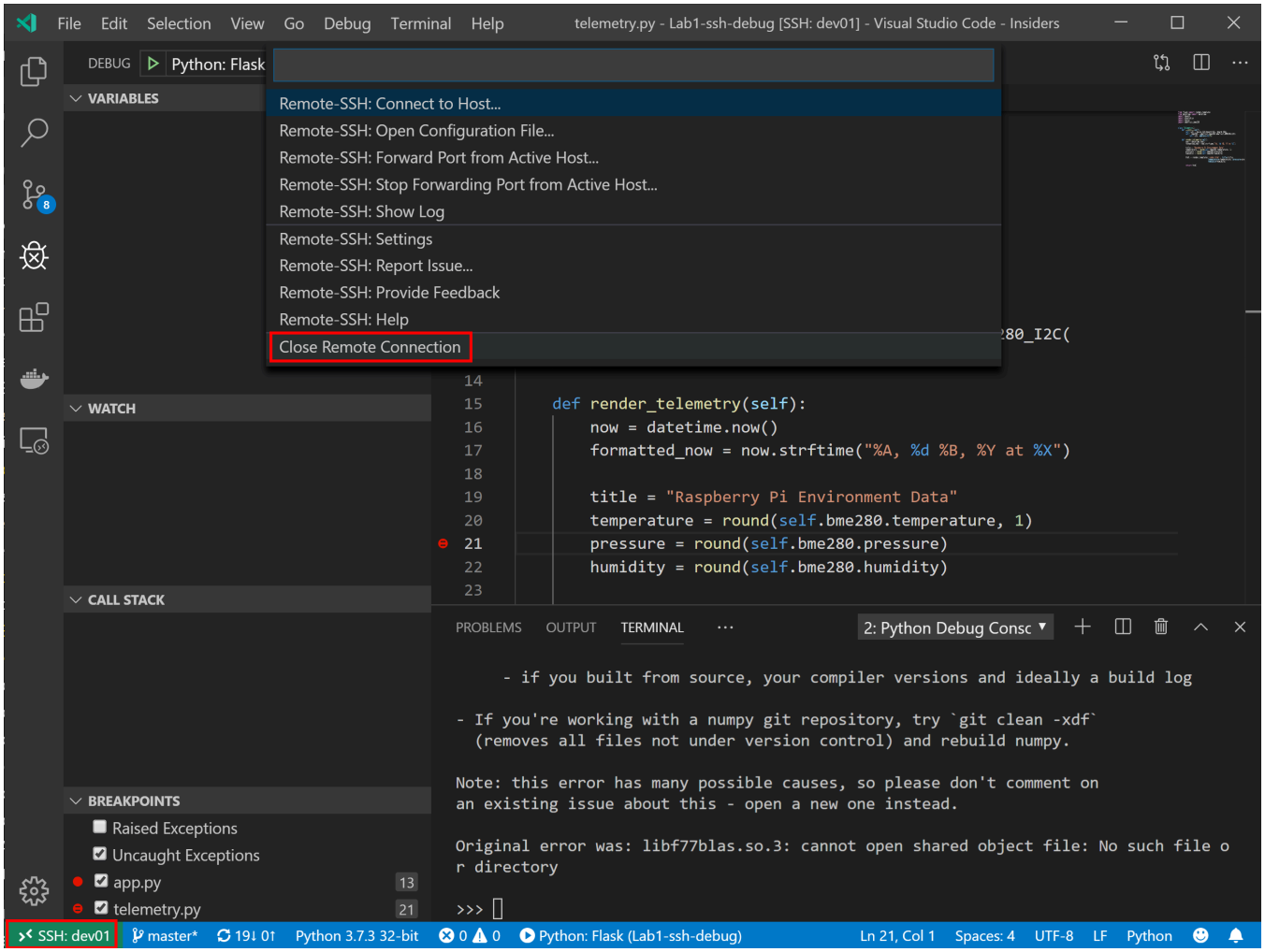
1. Update the Flask **index.html** template to display the current date and time.

2. Rerun the Flask app.

Closing the Visual Studio Code Remote SSH

From Visual Studio Code, **Close Remote Connection**.

1. Click the **Remote SSH** button in the bottom left-hand corner and select **Close Remote Connection** from the dropdown list.



Finished

finished

References

- Visual Studio Code
- Python
- Raspberry Pi
- Flask