

PyLab 1 : Raspberry Pi, 调试 Python 物联网应用程序

在 Twitter [@dglover](#) 上关注我



作者	微软云开发者倡导者 Dave Glover
平台	Linux, macOS, Windows, Raspbian Buster
服务	Azure 物联网中心
工具	Visual Studio Code
硬件	Raspberry Pi , Raspberry Pi Sense HAT
语言	Python
日期	2019 年 9 月

PDF 实验指南

您可能会发现下载和遵循“调试 **Raspberry Pi 物联网 Flask App** 动手实验室指南”的 PDF 版本更容易。

- [英语实验室指南](#)
- [简体中文实验室指南](#)

PyLab 内容

- [PyLab 1 : Raspberry Pi, 调试 Python 物联网应用程序](#)
- [PyLab 2 : Raspberry Pi, Azure 物联网中心和 Docker 容器调试](#)

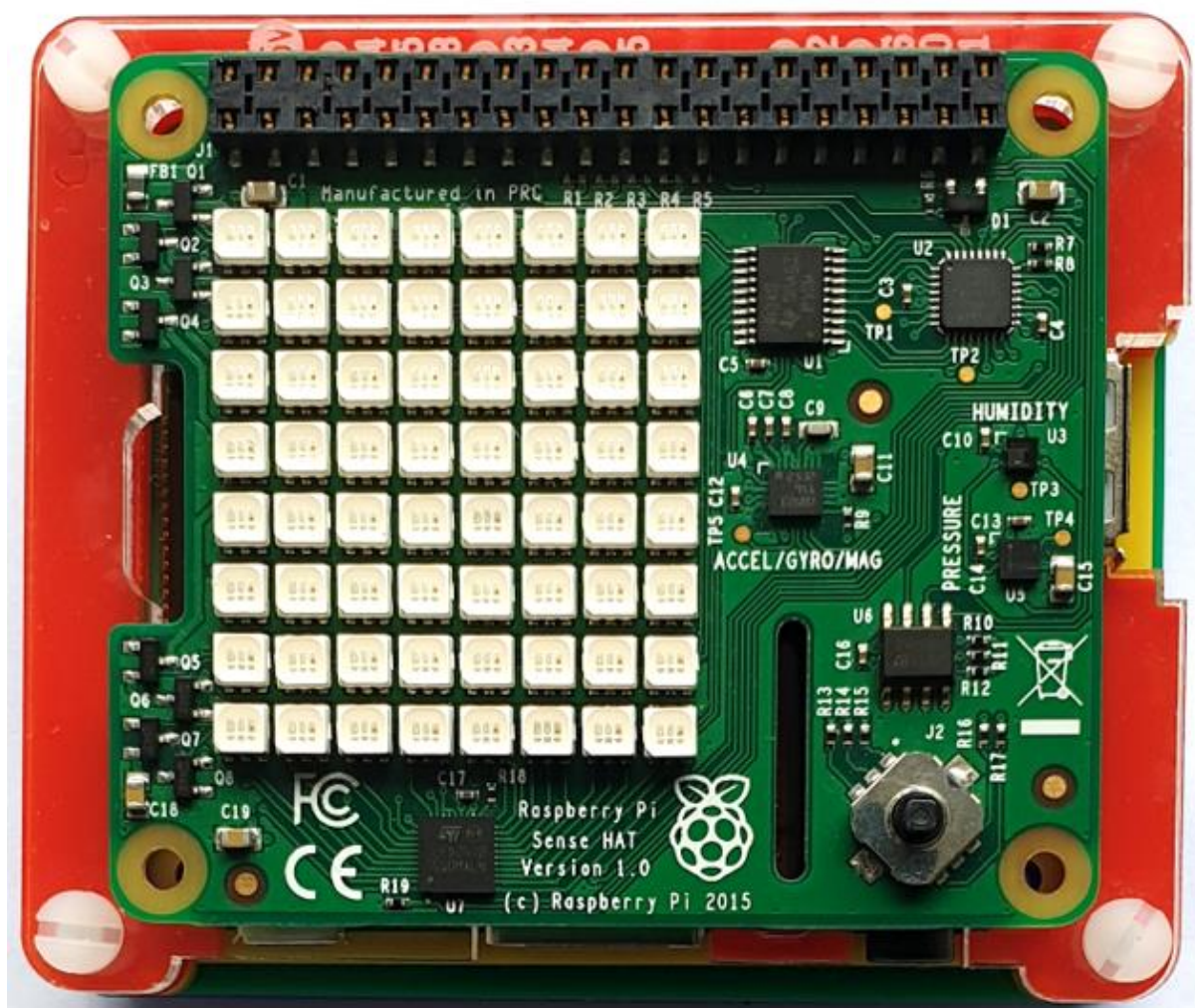
介绍

在本动手实验中，您将学习如何使用 [Visual Studio Code](#) 和 [Remote SSH](#) 扩展在 Raspberry Pi 上创建和调试 Python Web 应用程序。Web 应用程序将从连接到 Raspberry Pi 的传感器读取温度，湿度和气压遥测。

PyLab 设置

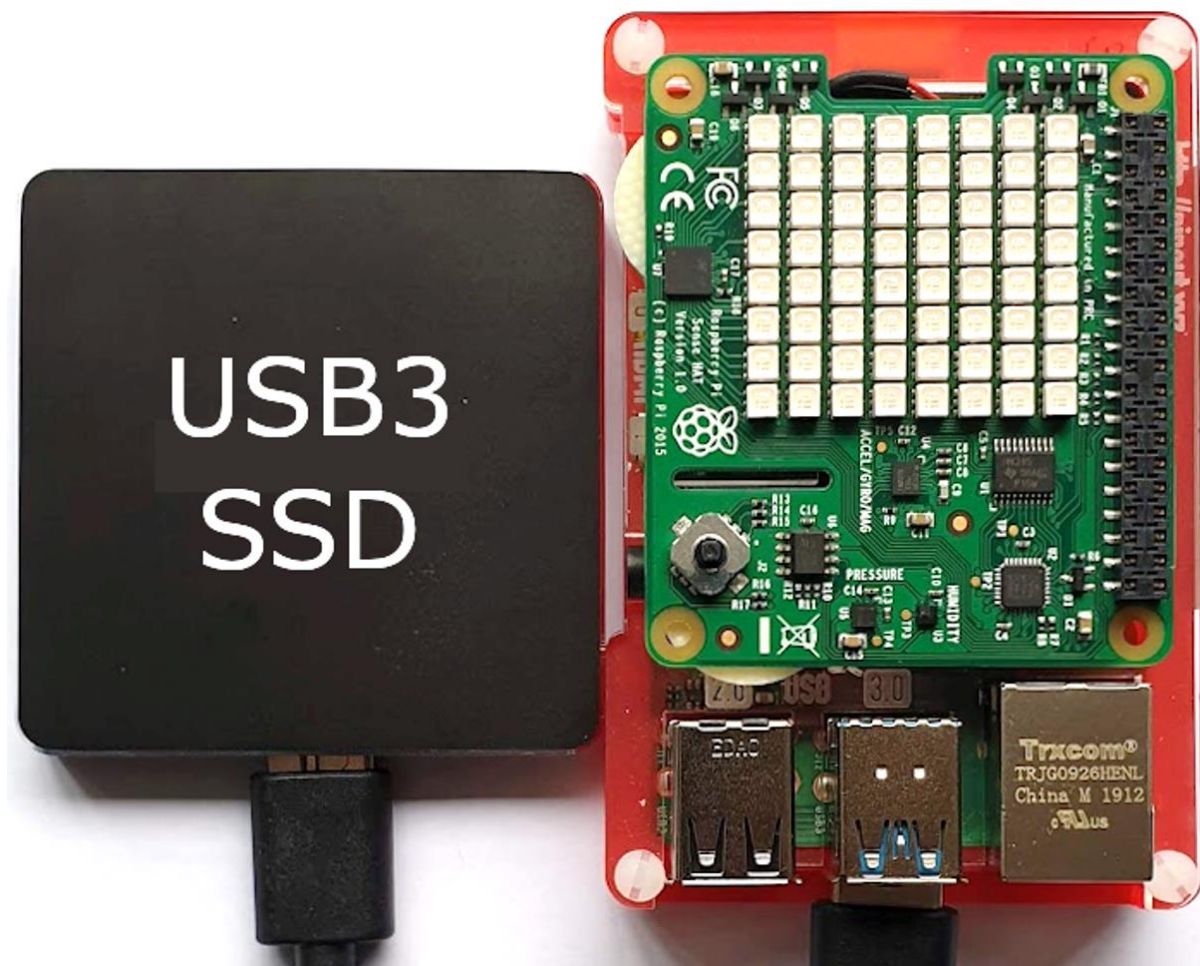
- [单用户设置](#)

此自动设置安装所需的库，Docker，并构建实验室 docker 镜像。



- [多用户设置](#)

多用户设置允许每个 Raspberry Pi 4 4GB 最多 20 个用户/学生。需要 USB3 SSD 驱动器来支持此数量用户的磁盘 IO 要求。安装脚本安装实验室内容和 Docker。构建实验室 Docker 镜像，并设置所有用户。



软件安装



这个动手实验室使用 Visual Studio Code。Visual Studio Code 是一个代码编辑器，是 [GitHub 上](#) 最受欢迎的**开源**项目之一。它可以在 Linux，macOS 和 Windows 上运行。

安装 Visual Studio Code

1. 安装 [Visual Studio Code](#)

Visual Studio Code 扩展

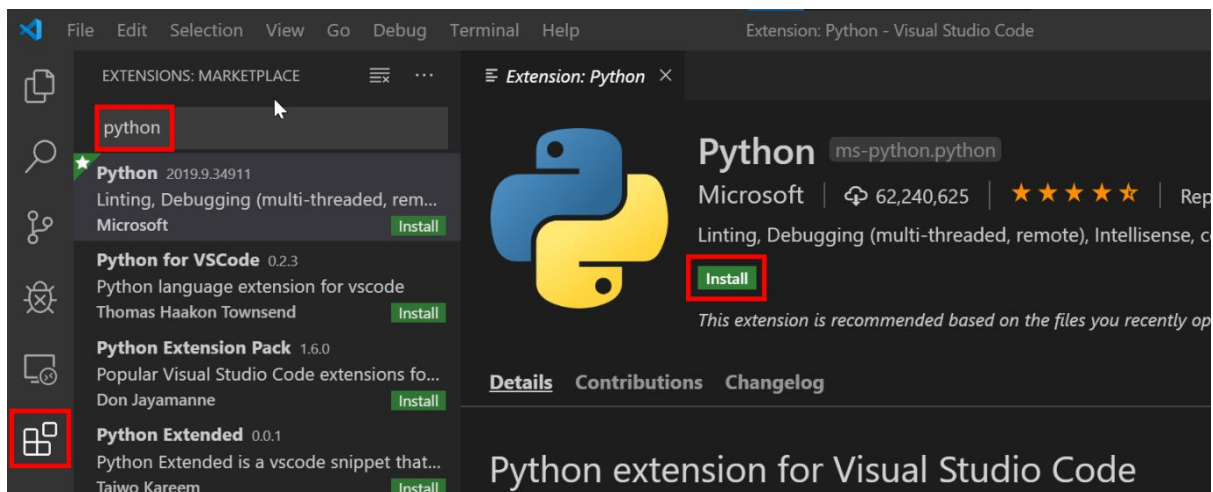
Visual Studio Code 包含开箱即用的功能只是一个开始。VS Code 扩展允许您为安装添加语言，调试器和工具，以支持您的开发工作流程。

浏览扩展程序

您可以从 Visual Studio Code 中搜索和安装扩展。从 Visual Studio Code 主菜单中打开 Extensions 视图，选择 **View > Extensions** 或单击 Visual Studio Code 一侧活动栏中的 Extensions 图标。



这将显示 [VS Code Marketplace](#) 上最受欢迎的 VS Code 扩展的列表。



安装 Python 和 RemoteSSH 扩展

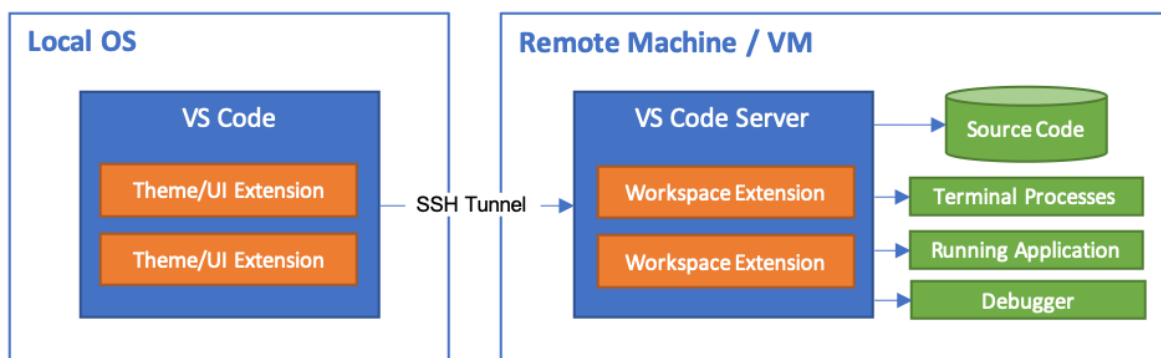
搜索并安装 Microsoft 发布的以下两个 Visual Studio Code 扩展。

1. Python

2. [Remote - SSH](#)

RemoteSSH 开发

Visual Studio Code Remote - SSH 扩展允许您打开任何远程计算机，虚拟机或容器，并充分利用 Visual Studio Code。



Raspberry Pi 硬件

如果您正在参加研讨会，那么您可以使用共享的网络连接 Raspberry Pi。您也可以使用自己的网络连接 Raspberry Pi 进行动手实验。

您将需要实验室指导员提供以下信息。

1. Raspberry Pi 的网络 IP 地址
2. 您指定的登录名和密码。

使用私钥/公钥进行 SSH 身份验证



为 [SSH](#) 身份验证设置公钥/私钥对是一种安全，快速的方法，可以从您的计算机登录到 Raspberry Pi。建议用于此动手实验。

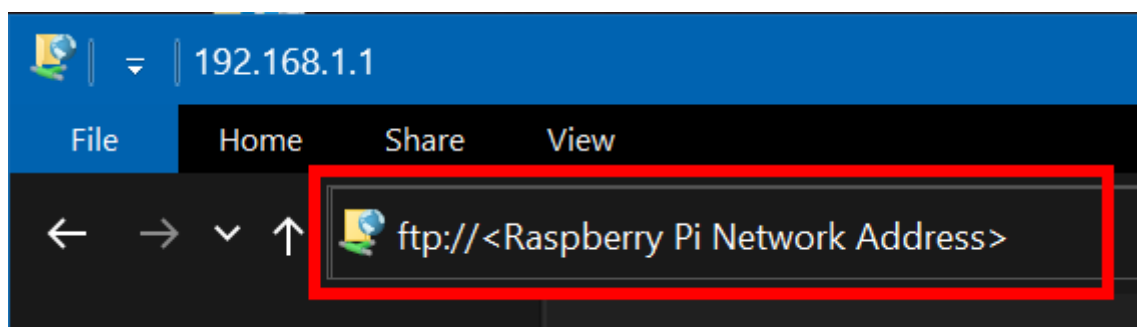
SSH 为 Windows 用户设置

SSH 实用程序将指导您完成成为 Visual Studio Code 和 Raspberry Pi 设置安全 SSH 通道的过程。

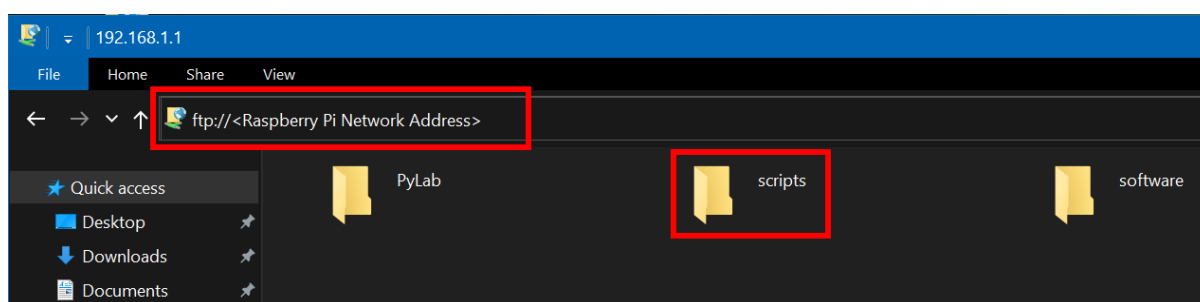
系统将提示您：

- Raspberry Pi 网络 IP 地址，
- Raspberry Pi 登录名和密码

1. 从 Windows 文件资源管理器中，打开 `ftp://<Raspberry Pi 地址>`



2. 将脚本目录复制到桌面



3. 打开复制到桌面的脚本文件夹

4. 双击 `windows-setup-ssh.cmd`

Linux 和 macOS 用户设置的 SSH 设置

SSH 实用程序将指导您完成成为 Visual Studio Code 和 Raspberry Pi 设置安全 SSH 通道的过程

系统将提示您：

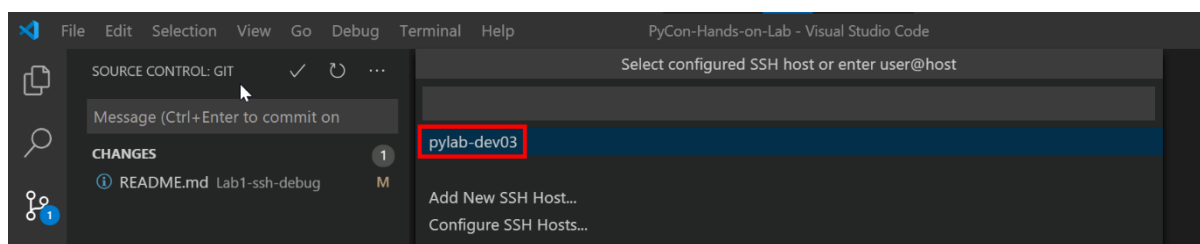
- Raspberry Pi 网络 IP 地址,
- Raspberry Pi 登录名和密码

1. 打开终端窗口
2. 复制并粘贴以下命令，然后按 **ENTER** 键

```
read -p "Enter the Raspberry Pi Address: " pyurl && \  
curl ftp://$pyurl/scripts/ssh-setup.sh | bash
```

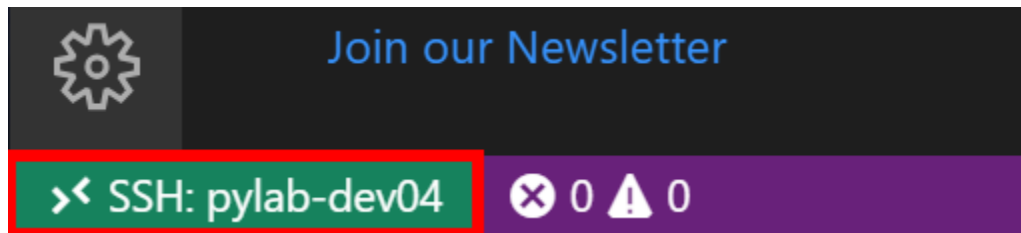
启动 RemoteSSH 连接

1. 启动 Visual Studio Code
2. 按 **F1** 打开命令选项板，键入 **ssh connect** 并选择 **Remote-SSH : Connect to Host**
3. 选择 **pylab-devnn** 配置



4. 检查 RemoteSSH 是否已连接。

连接需要一些时间，然后 Visual Studio Code 左下角的 SSH Status 将更改为 **>< SSH : pylab-devnn**。devnn 是你的 Raspberry Pi 登录名。

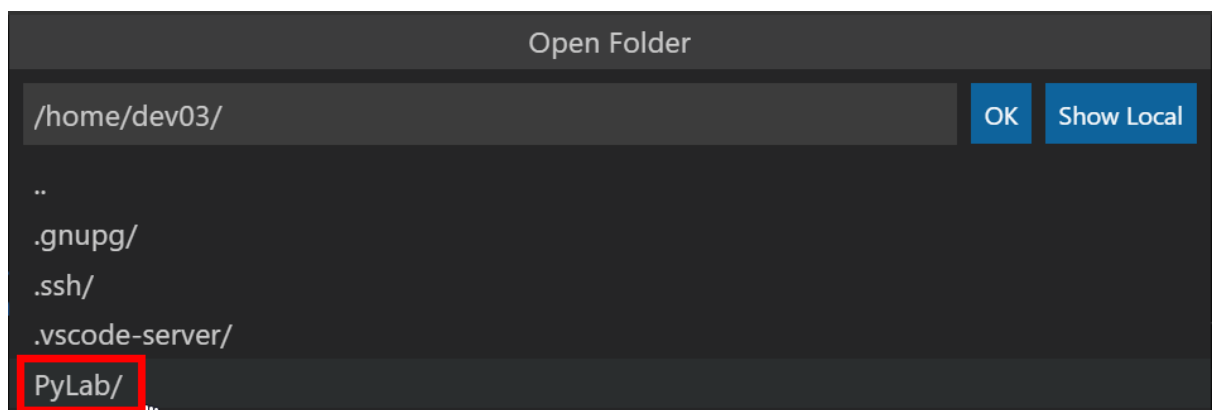


打开 PyLab 1 Python 调试项目

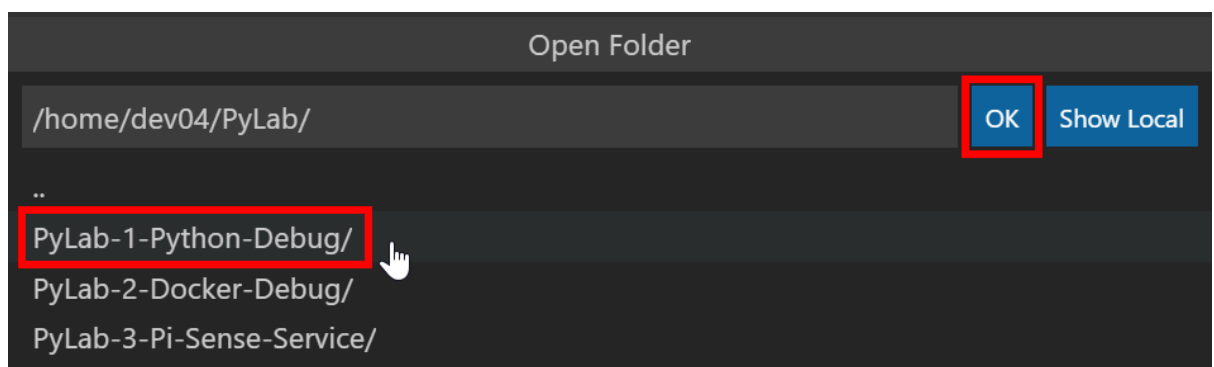
Python Flask Web 应用程序

在本实验中，我们将启动并调试 [Flask](#) 应用程序，该应用程序读取连接到 Raspberry Pi 的传感器。Flask 是一个流行的 Python Web 框架，功能强大，但对初学者也很容易。

1. 从 Visual Studio Code 主菜单：**文件 > 打开文件夹**
2. 选择 **PyLab** 目录

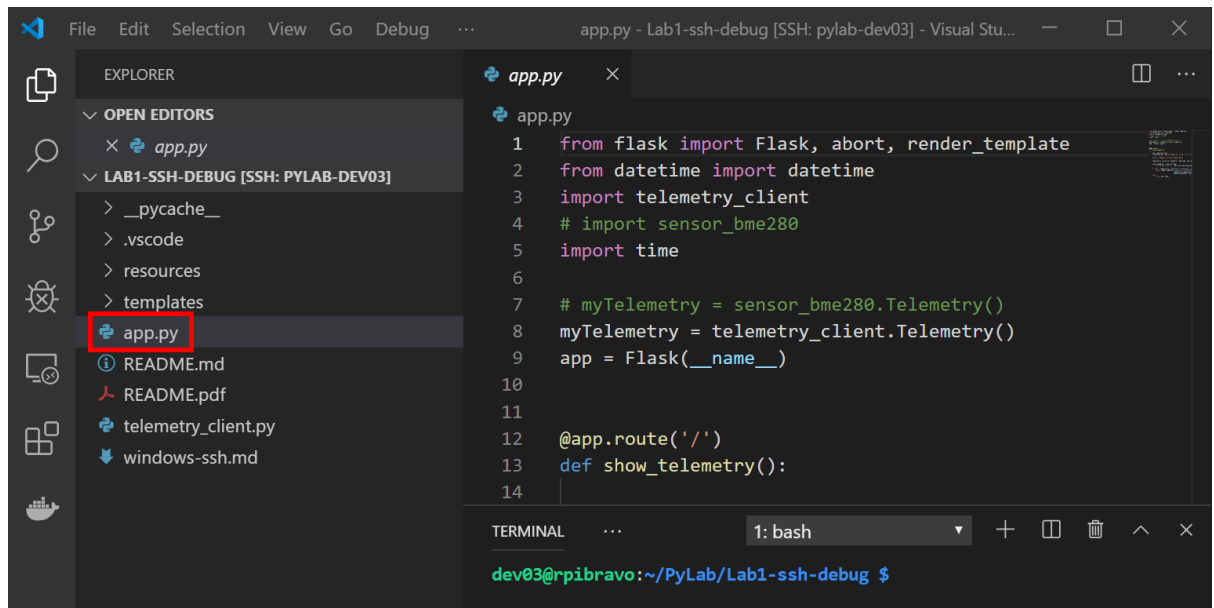


3. 接下来选择 **PyLab-1-Python-Debug** 目录



4. 单击“**确定**”以打开目录

5. 在资源管理器栏中，打开 `app.py` 文件并查看内容



花一点时间来查看 Python Flask Web 应用程序。

`app.py`

```
from flask import Flask, abort, render_template
from datetime import datetime
import telemetry_client
# import sensor_bme280
import time

# myTelemetry = sensor_bme280.Telemetry()
myTelemetry = telemetry_client.Telemetry()
app = Flask(__name__)

@app.route('/')
def show_telemetry():

    now = datetime.now()
    formatted_now = now.strftime("%A, %d %B, %Y at %X")

    title = "Raspberry Pi Environment Data"

    temperature, pressure, humidity, timestamp, cpu_temperature = myTelemetry.measure()

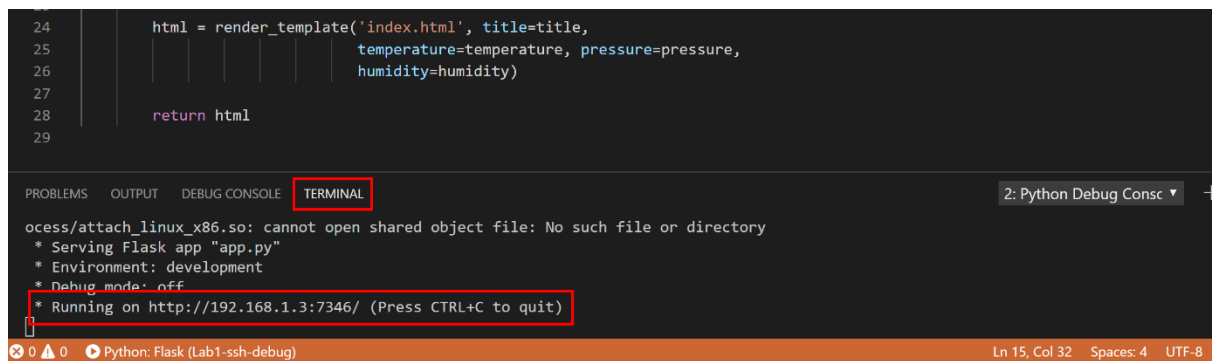
    sensor_updated = time.strftime(
        "%A, %d %B, %Y at %X", time.localtime(timestamp))

    if -40 <= temperature <= 60 and 0 <= pressure <= 1500 and 0 <= humidity <= 100:
        return render_template('index.html', title=title,
                                temperature=temperature, pressure=pressure,
```

```
        humidity=humidity, cputemperature=cpu_temperature)
else:
    return abort(500)
```

启动 Python Flask App

1. 按 **F5** 启动 Python Flask 应用程序。
2. 在 Visual Studio Remote 终端窗口中，单击在 **http : // ...** 链接。

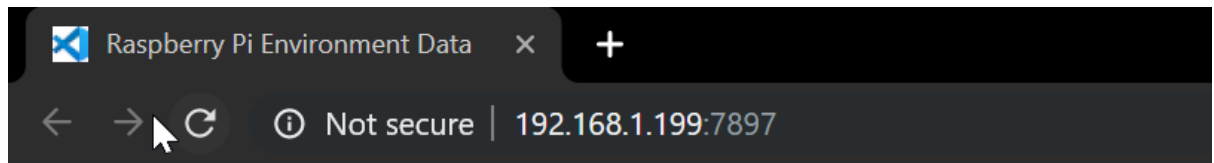


The screenshot shows the Visual Studio Remote interface. The top panel displays a Python code snippet for a Flask application. The bottom panel shows the 'TERMINAL' window with the following output:

```
process/attach_linux_x86.so: cannot open shared object file: No such file or directory
* Serving Flask app "app.py"
* Environment: development
* Debug mode: off
* Running on http://192.168.1.3:7346/ (Press CTRL+C to quit)
```

The 'Running on http://192.168.1.3:7346/' line is highlighted with a red box. The status bar at the bottom indicates 'Python: Flask (Lab1-ssh-debug)' and 'Ln 15, Col 32 Spaces: 4 UTF-8'.

3. 这将启动您的桌面 Web 浏览器。
 - Flask 应用程序将读取连接 Raspberry Pi 的传感器的温度，气压，湿度，并在 Web 浏览器中显示结果。



Raspberry Pi Environment Data

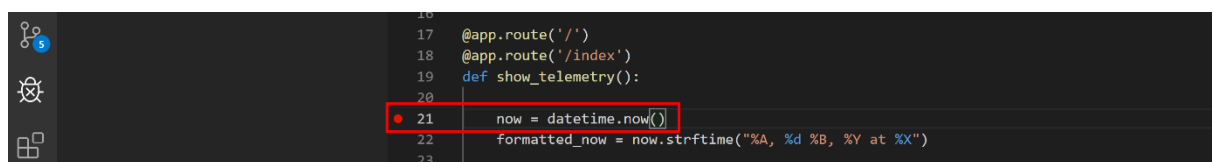
Raspberry Pi Sense HAT



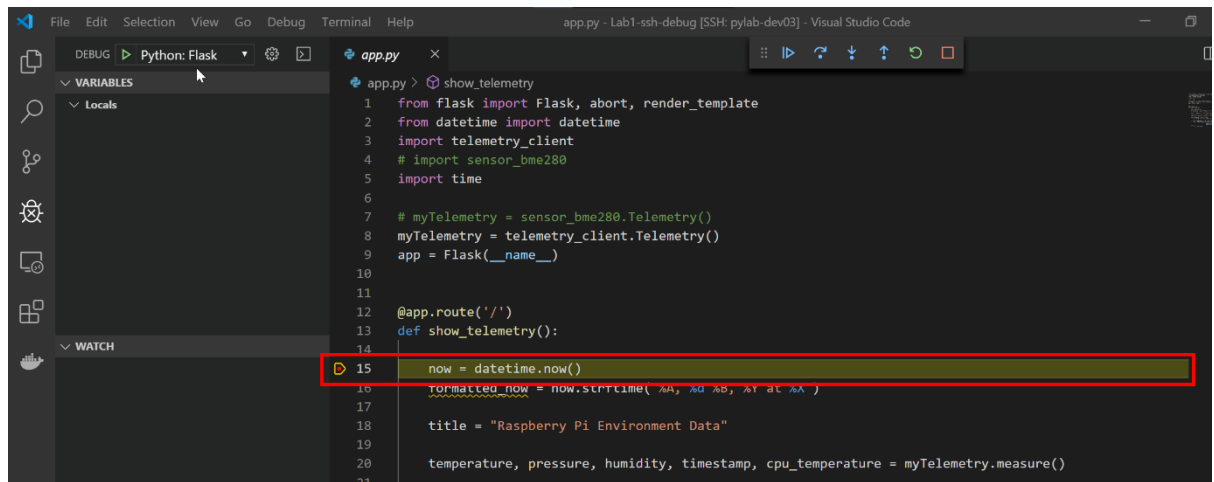
Telemetry	Value
Temperature	36.3 C
Humidity	22 %
Pressure	1020 hPa
CPU Temperature	57.9 C

使用断点进行调试

1. 切换回 Visual Studio Code 并确保 **app.py** 文件已打开。
2. 将光标放在现在读取的行 **= datetime.now ()**
3. 按 **F9** 设置断点。线上将出现一个红点，表示已设置断点。



4. 切换回 **Web 浏览器**并单击“**刷新**”。由于调试器已在您设置的断点处停止，因此网页**不会响应**。
5. 切换回 **Visual StudioCode**。您将看到代码已在**断点**处停止运行。



调试器工具栏选项

调试会话启动时，“调试”工具栏将显示在编辑器窗口的顶部。

调试工具栏（如下所示）将出现在 Visual Studio Code 中。它有以下选项：

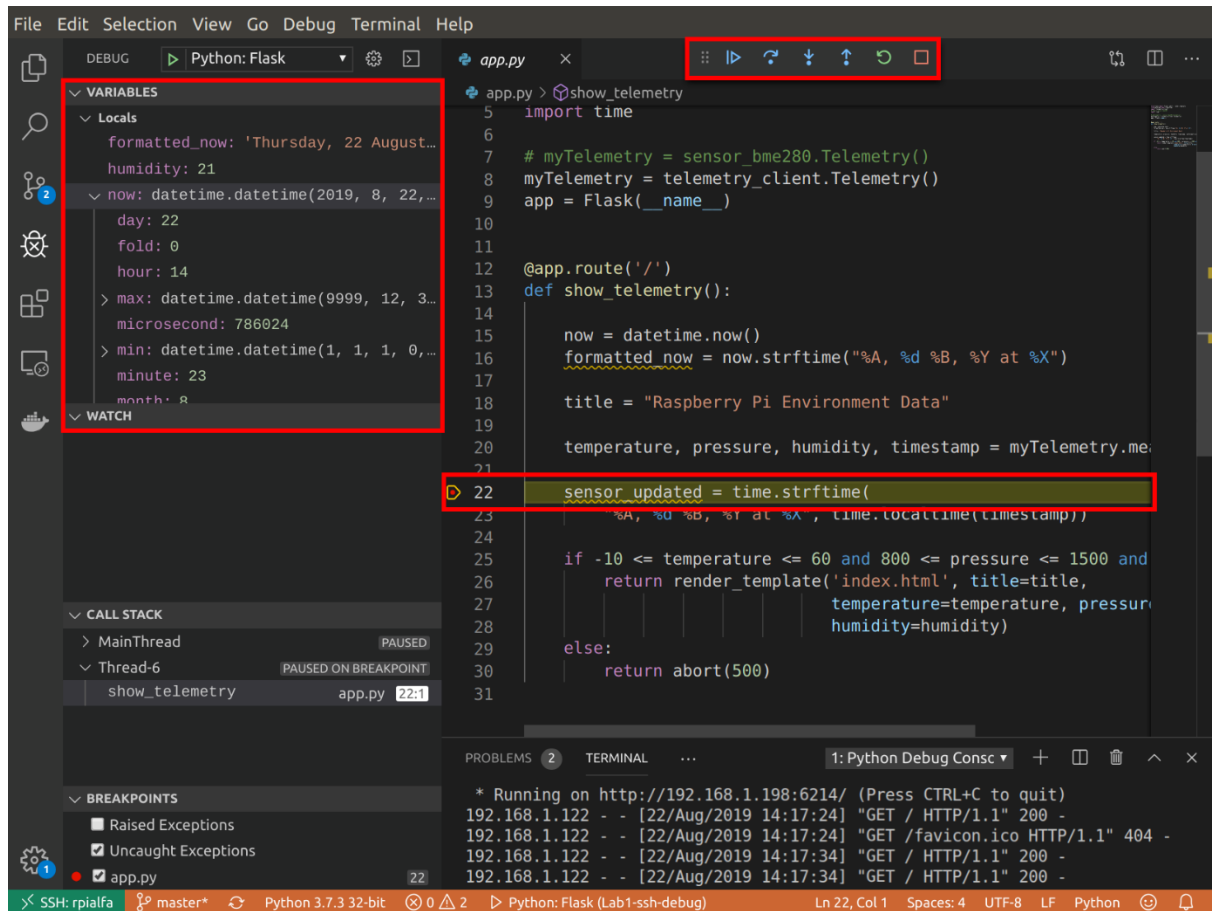


1. 暂停（或继续，F5），
2. 跳过（F10）
3. 步入（F11），
4. 走出去（Shift + F11），
5. 重启（Ctrl + Shift + F5），
6. 并停止（Shift + F5）。

开始调试

1. 按（F10）或单击调试工具栏上的“跳过”逐步执行代码。
2. 重复按 F10，直到到达读取的行，`if -40 <= temperature <= 60 and 0 <= pressure <= 1500 and 0 <= humidity <= 100`
3. 您会注意到 Python 变量显示在变量窗口中。

如果看不到变量窗口，请单击活动栏中的“调试”。



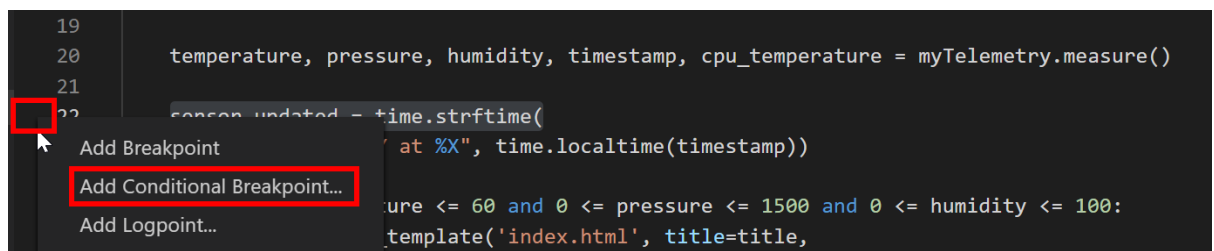
4. 尝试将**温度**变量更改为 **50**。提示，**右键**单击温度变量并选择“**设置值**”，或双击**温度**变量。
5. 按 **F5** 恢复 Flask 应用程序，然后**切换回 Web 浏览器**，您将看到网页上显示的温度，湿度和压力传感器数据。

使用条件断点进行调试

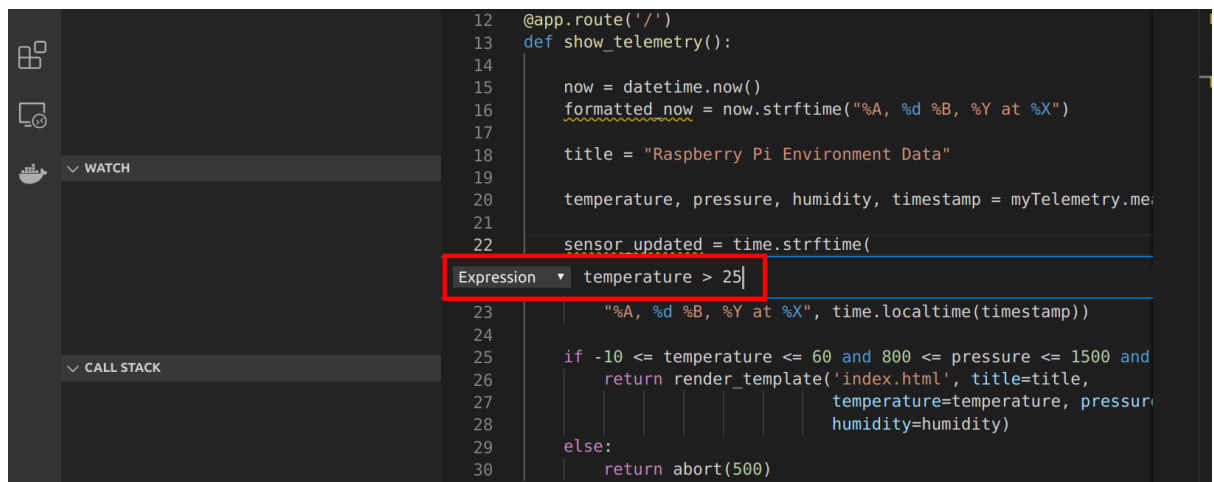
尝试设置**条件断点**

1. 清除现有断点。从主菜单中选择 **Debug > Remove all breakpoints**。

2. 确保打开 **app.py** 文件。
3. 右键单击第 22 行左侧的边距。



4. 选择**添加条件断点...**
5. 将条件设置为 `temperature > 25`，然后按 **ENTER**



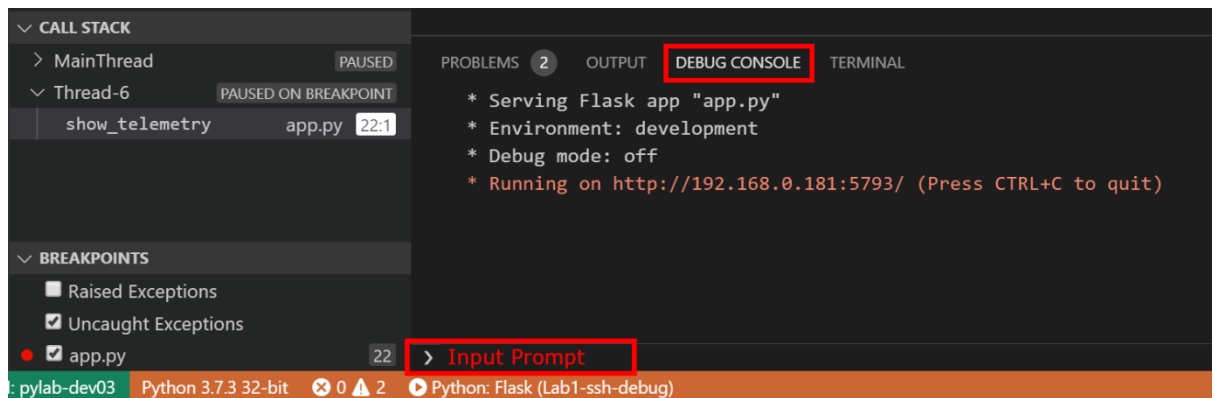
断点显示为红点，中间带有等号

6. 切换回 **Web 浏览器**并单击“**刷新**”。由于调试器已在您设置的断点处停止，因此网页**不会响应**。
7. 切换回 **Visual Studio Code**，您将看到调试器已在**条件断点**处停止。
8. 按 **F5** 继续运行代码
9. 切换回 **Web 浏览器**以查看该页面。

交互式调试控制台

Visual StudioCode 调试控制台将允许您访问 [Python REPL](#)（读取，评估，打印循环）。

1. 切换回 **Web** 浏览器并单击“刷新”。由于调试器已停止 Python 代码，因此网页不会响应。
2. 切换回 **Visual StudioCode**
3. 代码将在您先前设置的条件断点处停止。
4. 选择 Visual Studio 调试控制台窗口。



5. 在输入提示>中键入以下 Python 代码

6. `print(temperature)`

7. 按 **Enter** 键以执行您键入的 Python 代码。
8. 尝试从输入提示符运行以下 Python 代码段。

```
temperature = 24
import random
random.randrange(100, 1000)
```

9. 按 **F5** 继续执行 Python 代码。
10. 切换回您的 Web 浏览器以查看更新的页面。

实验室挑战

实验室挑战 1：更新 Flask 模板

1. 更新 **templates** 文件夹中的 Flask **index.html** 模板以显示当前日期和时间。

2. 重新运行 Flask 应用程序。

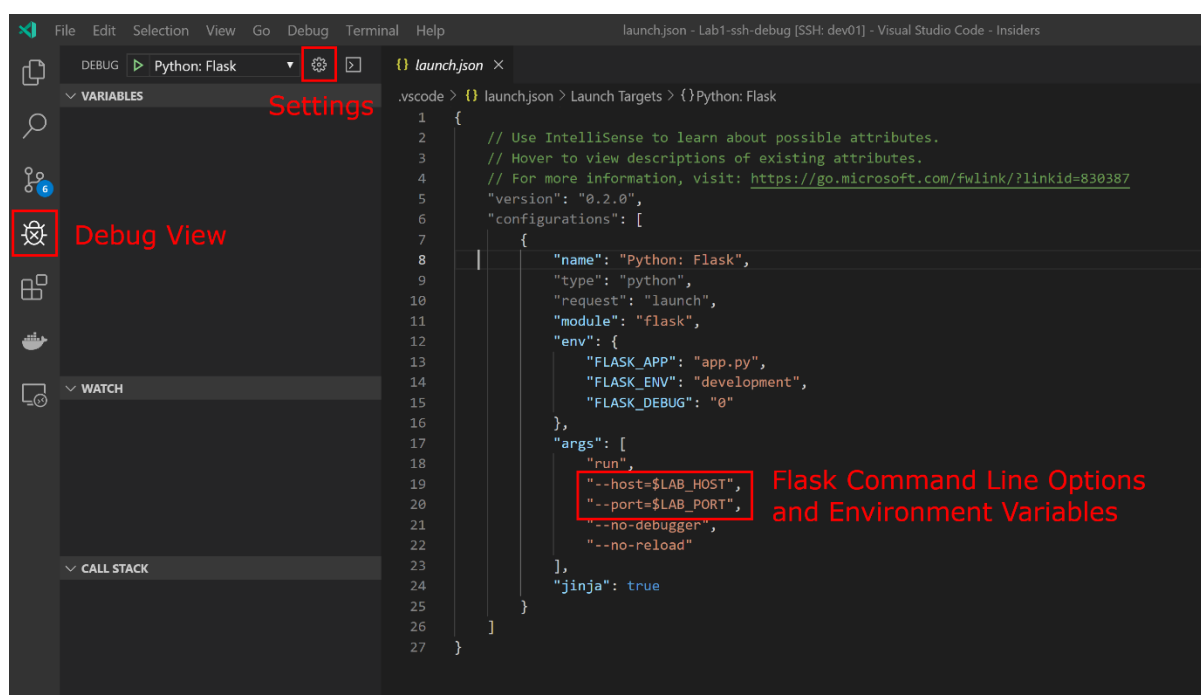
实验室挑战 2：使用调试器选项进行实验

要尝试的事情：

1. 查看 [Visual Studio Code Python 教程](#)
2. 查看 [Python Flask 教程](#)
3. 查看 [Visual StudioCode 调试教程](#)

查看调试启动设置

1. 在 Visual Studio Code 中切换到 Debug 视图（使用左侧活动栏）。



2. 单击“设置”按钮，将打开 launch.json 文件。
3. 该 launch.json 文件定义了 Flask App 将如何开始，启动时传递什么 Flask 命令行参数。

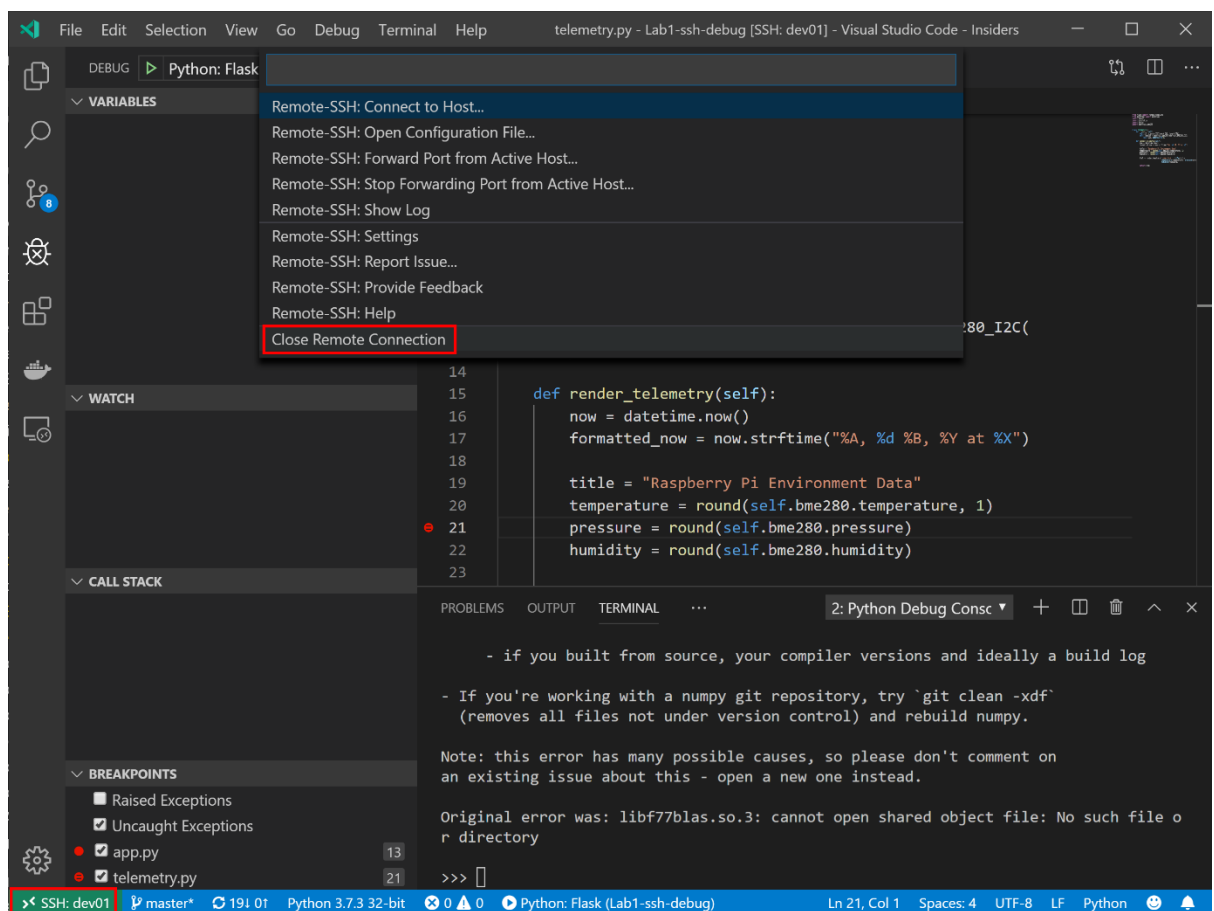
launch.json 文件中使用了两个环境变量。这些是 **LAB_HOST**（它是 Raspberry Pi 的 IP 地址）和 **LAB_PORT**（5000 到 8000 之间的随机 TCP / IP

端口号)。这些环境变量由.bashrc 脚本设置，该脚本在使用 Visual StudioRemoteSSH 连接到 Raspberry Pi 时运行。

关闭 Remote SSH 会话

从 Visual Studio Code，关闭远程连接。

1. 单击**左下角的 Remote SSH 按钮**，然后从下拉列表中选择 **Close Remote Connection**。



成品



参考

- [Visual Studio Code](#)
- Python
- Raspberry Pi
- [Flask](#)
- SSH 客户端安装排错[使用 SSH 进行远程开发](#)
- [安装支持的 SSH 客户端](#)

[PyLab-1-Debugging-a-Python-Internet-of-Things-Application](#) 由 [Dave Glover](#) 维护。