



Python Hands-on Labs Set Up

OpenWRT and the Linksys WRT 1900 ACS Router

Raspberry Pi Sense HAT

Raspbian Buster Headless/Lite will not boot with the Raspberry Pi Sense HAT Attached.

Run the following to fix the issue:

```
# Patch for Raspberry Pi Sense Hat on Buster Headless/lite
sudo sed -i 's/#hdmi_force_hotplug=1/hdmi_force_hotplug=1/g' /boot/config.txt && \
sudo sed -i 's/#hdmi_group=1/hdmi_group=2/g' /boot/config.txt && \
sudo sed -i 's/#hdmi_mode=1/hdmi_mode=4/g' /boot/config.txt && \
```

Visual Studio Code Remote Extensions

1. Start VS Code Insiders
2. Install Remote SSH Extension
3. Start Remote SSH session as user Pi
4. Install Python extension on Remote SSH

Update Raspberry Pi

```
sudo apt update && sudo apt upgrade -y && sudo reboot
```

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 <pi@Raspberry IP Address>
```

3. Test the SSH Authentication Key

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

SSH for Windows 10 (1809+) Users with PowerShell

1. Create an SSH Key

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

2. Copy SSH Key to Raspberry Pi

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

3. Test the SSH Authentication Key

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

Change Raspberry Pi Default Password for pi

```
passwd
```

Booting from a USB 3 Flash or SSD Drive

1. Plug in your USB 3 drive, then list your drives. If you only plugged in one USB drive then it's highly likely your drive will be /dev/sda.

```
sudo fdisk -l
```

2. Delete existing partitions and create a new primary partition on the USB drive.

```
sudo fdisk /dev/sda
```

fdisk commands

- p = print partitions
- d = delete a partition
- n = new partition - create a primary partition
- w = write the partition information to disk

3. Format the newly created partition

```
sudo mkfs.ext4 /dev/sda1
```

4. Create a mount point, mount the USB 3 drive, copy the Operating System files to the USB drive, and amend the cmdline.txt to enable booting from the USB 3 drive

```
sudo mkdir /media/usbdrive && \
sudo mount /dev/sda1 /media/usbdrive && \
sudo rsync -avx / /media/usbdrive && \
sudo sed -i '$s/$/ root=\dev\sda1 rootfstype=ext4 rootwait/' /boot/cmdline.txt
```

5. Reboot the Raspberry Pi

```
sudo reboot
```

Install the Fan SHIM Software

Check out the [Getting Started with Fan SHIM](#) article. In summary, install git and pip3 support, clone the Fan SHIM GitHub repo, install the dependencies, and then set up the automatic temperature monitor service that turns the fan on as required.

```
sudo apt install -y git sudo python3-pip && \
git clone https://github.com/pimoroni/fanshim-python && \
cd fanshim-python && \
sudo ./install.sh && \
cd examples && \
sudo ./install-service.sh --on-threshold 65 --off-threshold 55 --delay 2
```

Install Core Libraries

```
sudo apt install -y git python3-pip nmap bmon libatlas-base-dev libopenjp2-7 libtiff5 && \
sudo pip3 install --upgrade pip && \
sudo -H pip3 install numpy pillow requests pandas matplotlib flask jupyter autopep8 pylint

# Install Docker
# Links valid as of August 2019
# https://download.docker.com/linux/debian/dists/buster/pool/stable/armhf

wget https://download.docker.com/linux/debian/dists/buster/pool/stable/armhf/containerd.io
wget https://download.docker.com/linux/debian/dists/buster/pool/stable/armhf/docker-ce-cli
wget https://download.docker.com/linux/debian/dists/buster/pool/stable/armhf/docker-ce_19.

sudo dpkg -i containerd.io* && \
sudo dpkg -i docker-ce-cli* && \
sudo dpkg -i docker-ce_* && \
sudo usermod -aG docker $USER && \

sudo groupadd i2c && \
sudo chown :i2c /dev/i2c-1 && \
sudo chmod g+rw /dev/i2c-1 && \

sudo sed -i 's/CONF_SWAPSIZE=100/CONF_SWAPSIZE=2048/g' /etc/dphys-swapfile && \
sudo /etc/init.d/dphys-swapfile stop && \
sudo /etc/init.d/dphys-swapfile start && \

# Patch for Raspberry Pi Sense Hat on Buster Headless/lite
sudo sed -i 's/#hdmi_force_hotplug=1/hdmi_force_hotplug=1/g' /boot/config.txt && \
sudo sed -i 's/#hdmi_group=1/hdmi_group=2/g' /boot/config.txt && \
sudo sed -i 's/#hdmi_mode=1/hdmi_mode=4/g' /boot/config.txt && \

# Enable I2C
sudo sed -i 's/#dtparam=i2c_arm=on/dtparam=i2c_arm=on/g' /boot/config.txt && \

sudo reboot
```

Create Users

```
for i in {01..25}
do
    sudo useradd -p $(openssl passwd -1 raspberry) dev$i -G i2c,users,docker -m
    echo "dev$i ALL=(ALL) NOPASSWD: ALL" | sudo tee -a /etc/sudoers.d/010_pi-nopasswd
    echo "export LAB_PORT=$(shuf -i 5000-8000 -n 1)" | sudo tee -a /home/dev$i/.bashrc
    echo 'export LAB_HOST=$(hostname -I | cut -d" " -f 1)' | sudo tee -a /home/dev$i/.bashrc
done
```

Deploy Lab Content to all users

```
sudo rm -r -f ~/github && \
git clone --depth 1 https://github.com/gloveboxes/PyCon-Hands-on-Lab.git ~/github && \
# find .vscode-server-insiders -type f -name *.lock -exec rm {} \; && \

for i in {01..25}
do
    echo "Set up lab content for user dev$i"
    sudo rm -r -f /home/dev$i/.vscode-server-insiders
    sudo rm -r -f /home/dev$i/github
    sudo cp -r /home/pi/.vscode-server-insiders /home/dev$i/.vscode-server-insiders
    sudo cp -r /home/pi/github /home/dev$i/github
    sudo chown -R dev$i:dev$i /home/dev$i
done
```

Build Lab Docker Images

```
sudo systemctl start docker && \  
cd ~/github/Lab2-docker-debug && \  
docker build -t glovebox:latest . && \  
cd  
  
cd ~/github/Lab4-telemetry-service && \  
docker build -t lab-telemetry-service:latest . && \  
cd  
  
docker run -d \  
-p 8080:8080 \  
--privileged \  
--restart always \  
--device /dev/i2c-1 \  
--name pi-sense-hat \  
lab-telemetry-service:latest && \  

```

Clean Up Lab

Delete all devNN users and remove files and reset nopasswd

```
for i in {01..25}  
do  
    sudo deluser dev$i  
done  
  
echo "pi ALL=(ALL) NOPASSWD: ALL" | sudo tee /etc/sudoers.d/010_pi-nopasswd && \  
sudo rm -r /home/dev*  
  
# clean up docker images  
sudo systemctl start docker  
docker rm $(docker ps -a -q)  
docker rmi $(docker images -q) -f  
sudo systemctl stop docker
```

Useful Commands

Raspberry Pi CPU Temperature

```
watch vcgencmd measure_temp
```

Holding back VS Code Insiders Updates

[How to prevent updating of a specific package?](#)

apt

Hold a package:

```
sudo apt-mark hold code-insiders
```

Remove the hold:

```
sudo apt-mark unhold code-insiders
```

Show all packages on hold:

```
sudo apt-mark showhold
```