

# **HVDN Communicator**

Build Guide (Linux)

Hudson Valley Digital Network

12 February 2020

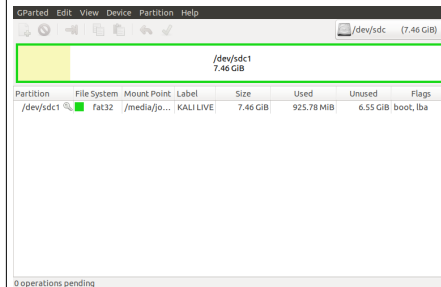
v0.8

# Introduction

This is a guide on how to install HVDN Communicator to a fresh Raspbian Lite image. All reference to “RPI” in this guide are an abbreviation for the Raspberry Pi Zero W hardware. The desktop operating system used in preparing the build is assumed to be a Debian or Debian derivative distribution. Ubuntu 18.04 was used in the preparation of this guide.

## Preparing Media

- From your PC insert an SD Card
- Run **sudo gparted** partitioning software
- From **Gparted** menu select **Devices** and SD card inserted
- Unmount the SD card **Delete** any existing partitions
- Right click on **unallocated**, select **new**, select **FAT32** as file system. Click **add**
- Click **check mark** to apply all operations



- Download Raspbian and Extract from ZIP file
- Install Raspbian on freshly formatted SD card

```
wget --max-redirect=3 https://downloads.raspberrypi.org/raspbian_lite_latest
unzip raspbian_lite_latest
sudo dd bs=4M if=2020-02-05-raspbian-buster-lite.img of=/dev/sdg conv=fsync
```

## Pi Connectivity

### Via connected keyboard, monitor and mouse

- Change to your home directory, run **sync** as sudo, and unmount the SD card
- Remove the SD card

```
cd ~
sudo sync
umount /media/sd-card/root
umount /media/sd-card/boot
```

### Via IP through Wireless Connection

- You can use this method if you wish to access the RPi via an existing WiFi network
- Change to the **boot** directory on the SD card
- Enable **ssh** access by creating a blank file named **ssh**
- Within the boot directory, create a file called **wpa\_supplicant.conf** and edit
- When you have opened the new file, add the configuration at right and save
- Be sure to replace **SSID** with your local wireless network SSID
- Change to your home directory, run **sync** as sudo, and unmount the SD card
- Remove the SD card

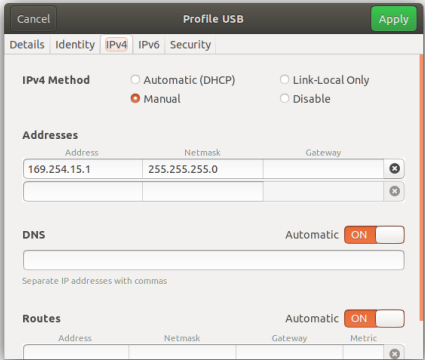
```
cd /media/sd-card/boot
touch ssh
```

```
vi wpa_supplicant.conf
```

```
country=US
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
network={
    ssid="MyWiFiNetwork"
    psk="aVeryStrongPassword"
    key_mgmt=WPA-PSK
}
```

```
cd ~
sudo sync
umount /media/sd-card/root
umount /media/sd-card/boot
```

### Via IP through USB connection

<ul style="list-style-type: none"> <li>In this configuration you will be powering the RPi via the USB cable to your PC. Be sure to use the correct USB cable</li> <li>On your Linux host you need to add a USB Network Interface and a network with the Pi on the USB interface. Give it a static address (ie 169.254.15.1 )</li> </ul>	
<ul style="list-style-type: none"> <li>Change to the <b>boot</b> directory on the SD card</li> <li>Edit the <b>config.txt</b> file</li> </ul>	<pre>cd /media/sd-card/boot vi config.txt</pre> <p>Append the following line:</p> <pre>dtoverlay=dwc2</pre> <p>Then save the file.</p>
<ul style="list-style-type: none"> <li>While in the boot directory, edit the <b>cmdline.txt</b> file, replace a line, then save file.</li> </ul>	<pre>vi cmdline.txt</pre> <p>Replace with the following all as one continuous line:</p> <pre>dwc_otg.lpm_enable=0 console=serial0,115200 console=tty1 root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline fsck.repair=yes rootwait modules-load=dwc2,g_ether quiet init=/usr/lib/raspi-config/init_resize.sh</pre> <p>Than save the file.</p>
<ul style="list-style-type: none"> <li>Remaining in the boot directory, enable <b>ssh</b> access by creating a blank file named <b>ssh</b></li> </ul>	<pre>touch ssh</pre>
<ul style="list-style-type: none"> <li>Change to the <b>rootfs</b> directory</li> <li>Edit the <b>interfaces</b> file</li> </ul>	<pre>cd /media/sd-card/rootfs/etc/network vi interfaces</pre> <p>Append the following lines:</p> <pre>allow-hotplug usb0 iface usb0 inet static     address 169.254.15.2     netmask 255.255.255.0     network 169.254.15.0     broadcast 169.254.15.255     gateway 169.254.15.1</pre> <p>Then save the file</p>
<ul style="list-style-type: none"> <li>Change to your home directory, run <b>sync</b> as sudo, and unmount the SD card</li> <li>Remove the SD card</li> </ul>	<pre>cd ~ sudo sync umount /media/sd-card/root umount /media/sd-card/b</pre>

# Initial Pi Setup

<ul style="list-style-type: none"><li>• Install Adafruit Radio Bonnet and Antenna</li><li>• Insert SD card</li><li>• Connect RPI and Power on</li><li>• Log in with default pi:raspberrypi</li></ul>	<p>If accessing via IP over USB, connect PC USB port to RPi USB Data port (next to mini-HDMI port)</p> <p>ssh -l pi &lt;yourpi&gt;</p> <p>If connecting via SSH for first time click <b>yes</b> to <b>accept fingerprint</b></p>
<ul style="list-style-type: none"><li>• Run Configuration tool</li></ul>	<pre>sudo raspi-config</pre>
<ul style="list-style-type: none"><li>• Navigate through each menu making selections as noted then exit tool</li></ul>	<ol style="list-style-type: none"><li>1. Change User Password to <b>&lt;something&gt;</b></li><li>2. Network options<ul style="list-style-type: none"><li>N1 Change hostname to <b>your call + number [1-15]</b> <b>yourcall-5</b></li></ul></li><li>3. Boot Options<ul style="list-style-type: none"><li>B1 Desktop/CLI choose <b>B1 Console</b></li></ul></li><li>4. Localization<ul style="list-style-type: none"><li>I1 Change Local to <b>en_US.UTF-8</b></li><li>I2 Change Timezone</li></ul></li><li>5. Interfacing options<ul style="list-style-type: none"><li>P2 <b>Enable SSH</b></li><li>P4 <b>Enable SPI</b></li><li>P5 <b>Enable I2C</b></li></ul></li><li>7. Advanced Options<ul style="list-style-type: none"><li>A1 Expand filesystem</li><li>A3 memory Split <b>Reduce GPU from 64 to 16</b></li></ul></li><li>8. Update</li></ol>
<ul style="list-style-type: none"><li>• Run <b>sync</b> as sudo, and <b>reboot</b></li></ul>	<pre>sudo sync sudo reboot</pre>
<ul style="list-style-type: none"><li>• Log back in to the Pi and run update and upgrade</li></ul>	<pre>sudo apt update sudo apt upgrade</pre>

# HAS Violet Install

Copy Report and Run Install	
<ul style="list-style-type: none"><li>• Ensure you are in the home directory</li><li>• Clone the Github HAS Violet repo locally</li><li>• Go into the build directory and run the install shell script</li></ul>	<pre>cd ~ git clone https://github.com/hudsonvalleydigitalnetwork/hasviolet.git cd /home/pi/hvdn-repo/hasviolet/build ../hvdn_hasviolet_install.sh</pre>
<ul style="list-style-type: none"><li>• Installation is complete. Apps are run from <b>/home/pi/hvdn</b></li></ul>	<p>Installed Directories include</p> <ul style="list-style-type: none"><li>◦ <b>/home/pi/hvdn</b> where the programs and their config files are to be run from</li><li>◦ <b>/home/pi/hvdn-repo</b> is local repo of hvdn apps are</li><li>◦ <b>/home/pi/hvdn-repo/hasviolet</b> is local repo of HAS Violet</li></ul>

# Using HVDN Communicator

HVDN Communicator is data only currently designed to be used on local LoRa networks. It is installed in ***/home/pi/hvdn-comm***

HVDN Communicator is built with Python. Applications include;

- **hvdn\_lora-beacon.py** sends a repeating broadcast message
- **hvdn\_lora-chat.py** is a half-duplex messaging app
- **hvdn\_lora-tx.py** sends a message to another LoRa station
- **hvdn\_lora-rx.py** listens for messages from other LoRa stations

Three files dependend by all applications are;

- **hvdn-comm.ini** is a configuration file
- **rf95.py** is a Python Library for the HOPE RFM95 modules on the Raspberry Radio Bonnet
- **font5x8.bin** used by the OLED on the Adafruit Radio Bonnet

## hvdn\_lora-beacon.py

Beacon a LoRa message

Usage: hvdn\_lora-beacon.py -c COUNT -t DELAY "message"

### OPTIONS

- c Number of times to repeat MESSAGE
- t Number of seconds before repeat MESSAGE
- MESSAGE is message to be send within double quotes

## hvdn\_lora-chat.py

Half-duplex LoRa messaging app

Usage: ./hvdn\_lora-chat [-r] [-s]

### OPTIONS

- h, --help show this help message and exit
- r, --raw\_data Receive raw data
- s, --signal Signal Strength

- Starts and loops in Listening Mode
- CTRL-Z to send a message, CTRL-C to exit program
- When in send mode
  - Recipient is node id (255 = broadcast address)
  - Message is whatever message followed by enter
  - Message is sent, return to listening mode

## hvdn\_lora-tx.py

Send a LoRa message

Usage: hvdn\_lora-tx.py -d DESTINATION "message"

### OPTIONS

-d Destination ID

MESSAGE is message to be send within double quotes

## hvdn\_lora-rx.py

Listens for messages from other LoRa stations

Usage: ./hvdn\_lora-rx.py -r -s

### OPTIONS

-h, --help show this help message and exit

-r, --raw\_data Receive raw data

-s, --signal Signal Strength

# HAS Violet Manual Install

A build script has been provided on the GitHub HAS Violet repo in the build directory (**hvdn\_hasviolet\_install.sh**) that automates the following sections.

## Install Raspbian Packages

- Log back into the RPi.
- Ensure you are in the home directory
- Install the following packages
  - pip3 – Python Package Index
  - Git – For cloning repositories

```
cd ~
sudo apt-get install python3-pip
sudo apt-get install git
```

## Install Python Libraries

- Install the following Python libraries
  - Python Image Library
  - APRS and APRSlib
  - Adafruit Radio Bonnet Libraries

```
sudo apt-get install python3-pil
sudo pip3 install aprs
sudo pip3 install aprslib
sudo pip3 install adafruit-circuitpython-rfm69
sudo pip3 install adafruit-circuitpython-rfm9x
sudo pip3 install adafruit-circuitpython-ssd1306
sudo pip3 install adafruit-circuitpython-framebuf
```

## Install HVDN Repository

- Ensure you are in the home directory
- Make two new directories called **hvdn** and **hvdn-repo**
- Change directory to hvdn-repo and clone the HASViolet repo from Github
- Copy the HASviolet stable directory to **hvdn**

```
cd ~
mkdir hvdn-comm
mkdir hvdn-repo
cd hvdn-repo
git clone https://github.com/hudsonvalleydigitalnetwork/hasviolet.git
cp /home/pi/hvdn-repo/hasviolet/stable /home/pi/hvdn
```

- Installation is complete. Apps are run from **/home/pi/hvdn**

Installed Directories include

- **/home/pi/hvdn** where the programs and their config files are to be run from
- **/home/pi/hvdn-repo** is local repo of hvdn apps are
- **/home/pi/hvdn-repo/hasviolet** is local repo of HAS Violet