

HVDN Communicator

Build Guide (Linux)

Hudson Valley Digital Network

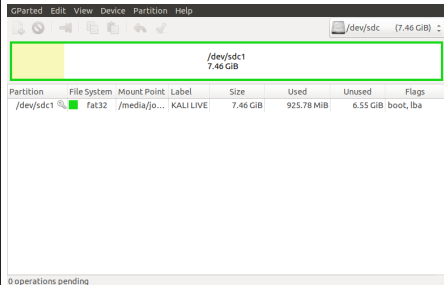
12 February 2020

v0.8

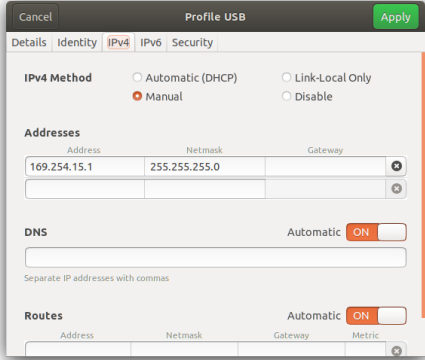
Introduction

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

<ul style="list-style-type: none">• 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	 <table><tr><th>Partition</th><th>File System</th><th>Mount Point</th><th>Label</th><th>Size</th><th>Used</th><th>Unused</th><th>Flags</th></tr><tr><td>/dev/sdc1</td><td>Fat32</td><td>/media/zero</td><td>KALI LIVE</td><td>7.46 GiB</td><td>925.78 MiB</td><td>6.53 GiB</td><td>boot, lba</td></tr></table> <p>0 operations pending</p>	Partition	File System	Mount Point	Label	Size	Used	Unused	Flags	/dev/sdc1	Fat32	/media/zero	KALI LIVE	7.46 GiB	925.78 MiB	6.53 GiB	boot, lba
Partition	File System	Mount Point	Label	Size	Used	Unused	Flags										
/dev/sdc1	Fat32	/media/zero	KALI LIVE	7.46 GiB	925.78 MiB	6.53 GiB	boot, lba										
<ul style="list-style-type: none">• Download Raspbian and Extract from ZIP file• Install Raspbian on freshly formatted SD card	<pre>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</pre>																

Pi Connectivity

Via connected keyboard, monitor and mouse	
<ul style="list-style-type: none"> Change to your home directory, run sync as sudo, and unmount the SD card Remove the SD card 	<pre>cd ~ sudo sync umount /media/sd-card/root umount /media/sd-card/boot</pre>
Via IP through Wireless Connection	
<ul style="list-style-type: none"> 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 	<pre>cd /media/sd-card/boot touch ssh</pre>
<ul style="list-style-type: none"> Within the boot directory, create a file called wpa_supplicant.conf and edit 	<pre>vi wpa_supplicant.conf</pre>
<ul style="list-style-type: none"> 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 	<pre>country=US ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev update_config=1 network={ ssid="MyWiFiNetwork" psk="aVeryStrongPassword" key_mgmt=WPA-PSK }</pre>
<ul style="list-style-type: none"> Change to your home directory, run sync as sudo, and unmount the SD card Remove the SD card 	<pre>cd ~ sudo sync umount /media/sd-card/root umount /media/sd-card/boot</pre>
Via IP through USB connection	
<ul style="list-style-type: none"> In this configuration you will be powering the RPi via the USB cable to your PC. Be sure to use the correct USB cable 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) 	
<ul style="list-style-type: none"> Change to the boot directory on the SD card Edit the config.txt file 	<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"> While in the boot directory, edit the cmdline.txt file, replace a line, then save file. 	<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"> Remaining in the boot directory, enable ssh access by creating a blank file named ssh 	<pre>touch ssh</pre>
<ul style="list-style-type: none"> Change to the rootfs directory Edit the interfaces file 	<pre>cd /media/sd-card/rootfs/etc/network vi interfaces Append the following lines: 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 Then save the file</pre>
<ul style="list-style-type: none"> Change to your home directory, run sync as sudo, and unmount the SD card Remove the SD card 	<pre>cd ~ sudo sync umount /media/sd-card/root umount /media/sd-card/b</pre>

Initial Pi Setup

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

HAS Violet Install

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

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

The build script (**hvdn_hasviolet_install.sh**) automates the following steps.

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 -R /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
```