HVDN Communicator

Linux Distro Build Guide

Hudson Valley Digital Network

2 December 2019

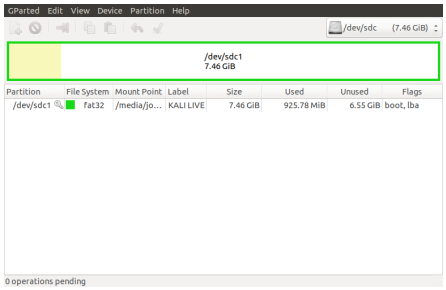
v0.4

Introduction

This is a guide on how to build the distribution image for the HVDN Communicator. 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 use din the preparation of this guide.

Preparing Media

- From your PC mount an 8GB SD Card
- Run **gparted** partitioning software
- From Gparted menu select Device and SD card inserted
- Destroy any existing parititons on the SD card and format as FAT32
- Note only 4GB of the SD Card will be formatted



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

```
wget -O https://downloads.raspberrypi.org/raspbian_lite/images/raspbian_lite-2019-07-12/2019-07-10-raspbian-buster-lite.zip

unzip 2019-07-10-raspbian-buster-lite.zip

sudo dd bs=4M if=2019-07-10-raspbian-buster-lite.img of=/dev/sdg conv=fsync
```

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 USB connection

- You can use this method if Bonjour Services are running on your PC. In this configuration you will be powering the RPi via the USB cable to your PC

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

Via IP through Wireless Connection

- You can use this method you wish to access the RPi via an existing WiFi network.

```
cd /media/sd-card/boot
touch ssh
vi wpa_supplicant.conf
```

- When you have opened the new file **wpa_supplicant.conf** add the configuration at right and save.
- Be sure to replace **SSID** with your local wireless network SSID

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

- Once saves, change directories, **sync** the drive and the unmount it

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

Pi Setup

• Install Adafruit Radio Bonnet and Antenna • Insert SD card • Connect RPI and Power on • Log in with default pi:raspberry	If connnecting via SSH for first time click **yes** to **accept fingerprint**
• Run Configuration tool	`sudo raspi-config`

• Navigate throught each menu making selections as noted then exit tool	`Change User Password to` **`hvdnCOMM$`**
	`Network options, N1 Change hostname to` **`your call + number [1-15]`**
	`Boot Options, B1 Desktop / CLI choose` **`B1 Consol`**`e`
	`Localization, I1 Change Local to` **`en_US.UTF-8,`** `I2 Change Timezone`
	`Interfacing options P2` **`Enable SSH,`** `P4` **`Enable SPI`**`, P5` **`Enable I2C`**
	`Advanced Options, A3 memory Split .` **`Reduce GPU from 64 to 16`**
	`Update`

• Run **sync** as sudo, and **reboot**	`sudo sync` `sudo reboot`

Install Packages, Libraries and Repository

Packages

• Log back into the RPi.
• Ensure you are in the home directory
• Install the following packages
  ◦ pip3 – Python Package Index
  ◦ nginx – Web Server (Future Interface)
  ◦ Git – For cloning repositories
  ◦ Tilde – MS-DOS Style Text editor
	`cd ~`
	`usdo apt-get install python3-pip`
	`sudo apt-get install nginx`
	`sudo apt-get install git`
	`sudo apt-get install tilde`

• Create self-signed SSL certificate for web server	`sudo openssl req -x509 -nodes -days 3650 -newkey rsa:2048 -keyout /etc/ssl/private/hvdn-selfsigned.key -out /etc/ssl`rvices are not running on your PC and/or you wish to access the RPi via a WiFi network. In this configuration you will use an external power source for the RPi`/certs/hvdn-selfsigned.crt`

• Provide answers in **bold** to questions presented
• Press **enter** for EMAIL address**Changing Node ID**
	`Generating a RSA private key`
	`.....+++++`
	`.....+++++`
	`writing new private key to '/etc/ssl/private/hvdn-selfsigned.key'`
	`-----`
	`You are about to be asked to enter information that will be incorporated`
	`into your certificate request.`
	`What you are about to enter is what is called a Distinguished Name or a DN.`
	`There are quite a few fields but you can leave some blank`
	`For some fields there will be a default value,`
	`If you enter '.', the field will be left blank.`
	`-----`
	`Country Name (2 letter code) [AU]:`**`US`**
	`State or Province Name (full name) [Some-State]:`**`New York`**
	`Locality Name (eg, city) []:`**`Hudson Valley`**
	`Organization Name (eg, company) [Internet Widgits Pty Ltd]:`**`Hudson Valley Digital Network`**
	`Organizational Unit Name (eg, section) []:`**`Communicator`**
	`Common Name (e.g. server FQDN or YOUR name) []:`**`www.hvdn.org`**
	`Email Address []:`

• Update web server to use SSL for connections by adding configuration to the default	`sudo vi /etc/nginx/sites-enabled/default`
	
	`Append the following lines:`
	
	`listen 443 ssl default_server;`
	`listen [::]:443 ssl default_server;`
	`ssl on;`
	`ssl_protocols TLSv1 TLSv1.1 TLSv1.2;`
	`ssl_ciphers "HIGH:!aNULL:!MD5 or HIGH:!aNULL:!MD5:!3DES";`
	`ssl_certificate /etc/ssl/certs/hvdn-selfsigned.crt;`
	`ssl_certificate_key /etc/ssl/private/hvdn-selfsigned.key;`

Python Libraries

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

```
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
```

HVDN Repository

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

```
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-comm
```

Create Build Inventory

- Capture a list of currently installed packages and save to a date stamped file

```
sudo apt-cache pkgnames|sort>/home/pi/hvdn-comm_build_20191202
```

Using HVDN Communicator

HVDN Communicator is currently designed to be used on flat local LoRa networks with each node having a unique ID between 1 and 254. 255 is reserved as a broadcast address to all nodes.

HVDN Communicator applications and supporting code are stored in ***/home/pi/hvdn-comm***

Executable applications include;

- **hvdn-comm-lora-broadcast_rf95.py** sends a repeating message to broadcast address (255)
- **hvdn-comm-lora-message_rf95.py** sends a message to another LoRa station
- **hvdn-comm-lora-rx_rf95.py** listens for messages from other LoRa stations

Running each command without arguments will provide you the syntax require to use it. Three files dependend by all applications are;

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

The only one of those three files that is edited for individual use is the **hvdn-comm.ini** file. A first step you need to do before transmitting is change the **node address**. While frequency and transmit power can be edited in that file, we recommend leaving the frequency until you are completely comfortable with using LoRa.

Changing Node ID

- Within the **hvdn-comm.ini** file change Node ID to a number unique on the network between 1 and 254

```
[DEFAULT]
gpio_rfm_cs=1
gpio_rfm_irq = 22
node_address = 1
freqmhz = 911.25
txpwr = 5
```