

02. nRF24 on Raspberry Pi

Captain's Log

Pi user is: pi

SSH login is: ssh [pi@192.168.1.118](https://github.com/nRF24/RF24)

password: pi9012

Key Reference:

What I finally discovered on 3/25 is the below site is the key resource for the nRF24:

nRF24 / RF24 github repo @ <https://github.com/nRF24/RF24>

And then the documentation that goes with the above github repo:

<https://rf24.readthedocs.io/en/latest/>

Introduction --

After much research and confusion I have decided to try to get an nRF24 chip working on one of my old Raspberry Pi's. This is a P1 (2011). I have seen some references to doing that as I was researching the ATTiny approaches. Since I got so confused and boloxed up trying to start out with the ATTiny 84 I figured this might be an easier way to get started. Also, I figure that with the richer environment of the Pi I'd be able to use it ongoing to monitor the nRF24 emissions as I work through experiments and issues with the ATTiny platform. And, finally, at the end of the day I expect to use a Pi as the data-catching server in my house that will catch the data from the garden and then post it to the web site on the internet.

So - here's my log of trying to get this to work.

4/4/2022 --

"First Light!," so-to-speak. Today I wired up the RPi to the little nRF24 carrier board I made; booted up the Pi; opened the SSH terminal window into the Pi; and fired up the scanner 'example' program. Amazingly enough, it is working - showing data on some of the channels - which I assume are my home's WiFi channels in the 2.4 GHz band. I wired it up using the reference as noted in the Key Reference section above. That is:

RF24 | Pi pin | Function

1 25 GND

2 17 VCC

3	15	CE
4	24	CSN
5	23	SCK
6	19	MOSI
7	21	MISO
8		(unused)

See file: Dropbox/projects/MoistureSensor/nRF24-with-Rpi/FirstLight_04-04-2022.txt to see the terminal output from the running scanner program.

So now I think I should make a change or two to the scanner.cpp program to see if I can wrap my head around the radio code and how to control it.

3/25/2022 (b) --

The below, 3/25/2022 (a) activities, created a mass of confusion in my mind. Esp as I was unable to compile the example receiver program. It couldn't find any rf24 libraries. And I couldn't work out what was going on. Plus the comments section seemed to imply there were any number of errors and faults with repo/code at that 'Key Reference' site I was using.

So...more googling about. I came across this reference; which gives me confidence because the dates of last updates and pull requests are current, not 8 years old like so many others. So now I'm going down the path as this github repo:

RF24/docs/linux_install.md @ https://github.com/nRF24/RF24/blob/master/docs/linux_install.md

5:54pm: OK, so I am following the step-by-step instructions at the above './linux_install.md' page. The Pi is now doing all sorts of downloading and install stuff. I took all the defaults for the various options it presented to me. It installed with no errors. Here is what it presented me in the end:

```
Complete! To run the example:
cd ~/rf24libs/RF24/examples_linux/build
sudo ./gettingstarted
Would you like to install the unified python wrapper package (pyrf24) [y/N]?

*** Installer Complete ***
See http://tmrh20.github.io for documentation
See http://tmrh20.blogspot.com for info

Listing repositories in ~/rf24libs
RF24
```

For the complete transcript of the terminal output, see: /home/jroc/Dropbox/projects/MoistureSensor/nRF24-with-Rpi/RF24_LibInstall_Log.txt

I had to do a /home/pi/rf24libs/RF24/examples_linux/build/scanner command to find the compiled scanner executable. (I could see from the install.sh terminal output that it did compile it.) So I did find it at /home/pi/rf24libs/RF24/examples_linux/build/scanner. So then I executed:

```
cd /home/pi/rf24libs/RF24/examples_linux/build/scanner
./scanner
```

And 'scanner' is running!! Finally I feel I'm getting somewhere.

OK, time to wire the nRF24 module to the Pi. Note that the install readme page states this:

Edit the gettingstarted example, to set your pin configuration

```
nano gettingstarted.cpp
make
sudo ./gettingstarted
```

3/25/2022 (a) --

Today's Key Reference: **Connecting an nRF24L01+ to Raspberry Pi** @ <https://www.hackster.io/wirekraken/connecting-an-nrf24l01-to-raspberry-pi-9c0a57>

Ok, interesting to study the git repo from the above reference. So apparently much / most? of the code is actually using the same source as the Arduino and ATtiny examples I have been looking at. The nRF24 library contains pragma defs to tell the source what architecture you are compiling for. So this 'wirekraken' guy took the original nRF24 source, set the defs for 'Linux,' and then built some examples to run on the pi.

Importantly, he also created the 'scanner' executable program. Which, apparently, is compiled to run on the Pi. So, in principle, it can simply be downloaded and run.

But I am going to git clone the entire repo, per his sequence of steps in his instructable. Tho first I have to install git on the pi.

1. First make sure the pi is all up to date. Execute:

```
sudo apt full-upgrade
```

This took quite awhile. About 30 minutes I'd say.

Then: `sudo reboot`

2. ~~Make a directory to clone into and CD into. Execute:~~

```
mkdir nrf1  
cd nrf1
```

3. Now install git. Execute:

```
sudo apt install git
```

Reference @ <https://linuxize.com/post/how-to-install-git-on-raspberry-pi/>

4. Clone the repo. Execute:

```
git clone https://github.com/wirekraken/Raspberry-Pi-NRF24.git xxx
```

Turns out this command created a new subdirectory, called Raspberry-Pi-NRF24. So I shouldn't have done #2 above. Or better, I should specified which directory I wanted to use so that I could have had my intended simple to type directory name.

5. Attempt to run the scanner program. Execute:

```
sudo ./scanner
```

For some reason I don't understand, you do have use that "./" syntax in front of the program for it to be executed.

Running it, with no radio chip yet wired in, I get the same error many others have reported. Namely:
./scanner: error while loading shared libraries: librf24.so.1: cannot open shared object file: No such file or directory . If I am reading the comments about this correctly, you do have to have radio wired in to not get that error.

For hardware connections - Key Reference: web page with the pi's pinout reference, which I'll need in order to hook up the nRF24 chip to the pi. See "Raspberry gPlo" @ <https://learn.sparkfun.com/tutorials/raspberry-gpio/gpio-pinout>

And this is also useful: @ <https://docs.google.com/document/d/1aZ2xJYndN81k93rzOXoLlpljogRsWb3JwEnGnfbqTXE/edit>

3/24/2022 --

The attempt to get my pi up and running on 3/22 didn't work. First, trying to use an adapter to connect the screen to one of my big Dell screens didn't work. The Dell screen never saw a 'signal.' Going downstairs and connecting to the TV via HDMI did work to show the raspberry pi OS's desktop; but I was never able to get WiFi working. After *much* mucking about I found an article about running the pi OS 'headless.' There is a way to do that such that you can use the graphical UI on the pi from a window on a seperate PC, using a special app on your PC for this - so-called 'VNC.' However, I elected to reburn my SD card with the 'Lite' version of pi OS - this is the Linux OS without any desktop. With this version you SSH into the pi from any terminal running on any PC in your network.

Also of note - in making that 2nd SD card image I found that the raspberry pi 'Imager' app was available in the fedora repository, so I installed that app onto the white laptop and used that to burn the SD card. This is much easier than the manual method I used on 3/22. Definitely go this way in future.

To create the SD card with the OS on it and SSH into it; and/or run the UI remotely: "**How to setup your Raspberry Pi, Headless!**" @ <https://www.hackster.io/435738/how-to-setup-your-raspberry-pi-headless-8a905f>

To get the WiFi dongle to work on my Pi, the post @ <https://raspberrypi.stackexchange.com/questions/112062/raspbian-buster-lite-couldnt-communicate-with-wpa-supPLICANT> finally led me to a solution. The dongle I am using - "AirLink 101 Wireless N 150" is an older dongle, so it doesn't get automatically configured to run at boot up. The above linked to article itself points to another article that contained the answer. The answer being that you have to force the dhcpd daemon to accept the dongle as a WiFi interface device:

It seems that **dhcpcd** does not acknowledge that it is a WiFi adapter, so we have to force it. Just add an interface section into **/etc/dhcpcd.conf**:

```
interface wlan0
env ifwireless=1
env wpa_supplicant_driver=nl80211,wext
```

The driver specification is required, otherwise wpa_supplicant will die with the error message

```
nl80211: Driver does not support authentication/association or connect commands
nl80211: deinit ifname=wlan0 disabled_11b_rates=0
wlan0: Failed to initialize driver interface
```

After a reboot everything worked as expected, which is confirmed by the wpa_supplicant process running:

```
wpa_supplicant -B -c/etc/wpa_supplicant/wpa_supplicant.conf -iwlan0 -Dnl80211,wext
```

Notice that the configuration file used is **/etc/wpa_supplicant/wpa_supplicant.conf**, but if one named **wpa_supplicant-wlan0.conf** exists, it will be preferred.

So I edited the file **/etc/dhcpcd.conf** (while SSH'd in via ethernet cable) to add those above 3 lines.

Edited with command: `sudo nano /etc/dhcpcd.conf`

After doing this I could then get into the WiFi settings panel from the `raspi-config` command. I did that and put in the SSID: `cosmosnet` and the passkey and saved those.

Then after a reboot the WiFi worked fine.

OK - so now I figure I am ready to try to install the nRF24 utilities referenced in the post @

3/22/2022 --

Downloaded, and wrote to the SD card, the base Raspberry Pi OS. This one:

Raspberry Pi OS with desktop
Release date: January 28th 2022
System: 32-bit
Kernel version: 5.10
Debian version: 11 (bullseye)
Size: 1,246MB

Downloaded from: <https://www.raspberrypi.com/software/operating-systems/>

I used the instructions in my Evernote note "Raspberry Pi OS Install" to burn the image to the SD card.

I found a web page with the p1's pinout reference, which I'll need in order to hook up the nRF24 chip to the pi. See "Raspberry gPIo" @ <https://learn.sparkfun.com/tutorials/raspberry-gpio/gpio-pinout>