TNC-Pi Assembly Instructions & Operating Tips

By John Hansen, W2FS

Portions by

John Wiseman, G8BPQ – Linux configuration

Peter Klotzback, AF9FA - YAAC

Document editing support by

Ed Slingland, N2WD

Website

http://tnc-x.com/TNCPi.htm

Contents

Introduction

Thank you for purchasing a TNC-Pi2: TNC-X for Raspberry Pi.

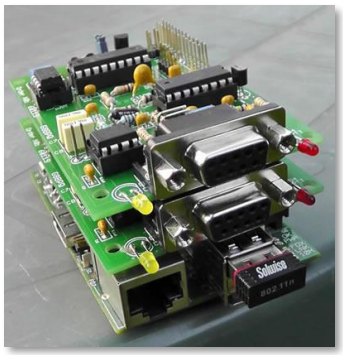


Figure 1: Two TNC-Pi's stacked on one Raspberry Pi

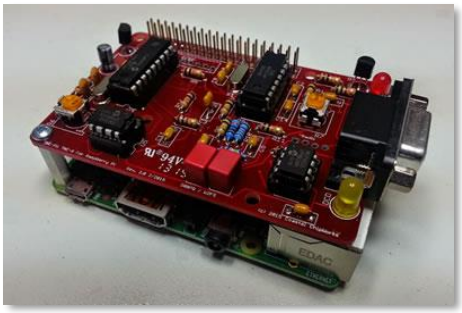


Figure 2: TNC-Pi 2 mounted on a Raspberry Pi 2.

Assembly Instructions

Parts List

Your kit should come with the parts listed in the table below. You can use the two columns of checkboxes to track your progress. As you inventory your parts, put a checkmark in the first column. Once you've installed a part, put a checkmark in the second column.

Inventoried	Installed	Designation	Description	Notes
☐	☐	C1	4.7 uf or 10 µf electrolytic	Polarized
☐	☐	C2, C4, C5, C6, C11, C14, C15, C21	0.1 µf monocap	
☐	☐	C7, C8	0.01 µf 2.5%	Red and yellow or grey; if grey marked 10 nF
☐	☐	C9, C10	18 pf ceramic disk	
☐	☐	C12, C13	22 pf ceramic disk	
☐	☐	C3	100 pf ceramic disk	
☐	☐	R1, R2, R3	100K resistor	Brown, black, yellow
☐	☐	R4, R17, R19,	1K resistor	Brown, black, red
☐	☐	R5, R11, R14, R16, R18	10K resistor	Brown, black, orange
☐	☐	R8	24.9K resistor	Red, yellow, white, red, brown
☐	☐	R9	9.31K resistor	White, orange, brown, brown, brown

☐	☐	R10	18.7K resistor	Brown, grey, purple, red, brown
☐	☐	R6, R7	10K trimmer potentiometer	Orange or Blue
☐	☐	X1	3.57 MHz crystal	Marked A035G87
☐	☐	X2	20.00 MHz crystal	
☐	☐	D4	Red LED (PTT)	Polarized
☐	☐	D5	Yellow LED (DCD)	Polarized
☐	☐	Q1	PN2222 transistor	3 pin, flat side (don't confuse with U1)
☐	☐	U1	MCP1700-33 or MCP1700-30 Regulator	3 pin, flat side
☐	☐	U2	CML MX-614 Modem	16 pin IC
☐	☐	U3	PIC16F1847 microcontroller	18 pin IC
☐	☐	U4	MCP6023 Op Amp	8 pin IC
☐	☐	U5	23K640 Memory or 23K256 Memory	8 pin IC
☐	☐	IC Sockets	For U2, U3, U4, U5	One 16-pin, one 18-pin and two 8-pin sockets
☐	☐		2 x 20 extra long header	
☐	☐		9 Pin D-Sub connector	
☐	☐		Printed Circuit Board	

Assembly

☐ Start by installing the parts that lie flat on the board. This includes:

a. all of the 0.1uf monocaps and

b. all of the resistors except for R6 and R7.

Note that all of the components except for the 40 pin header are installed on the side of the board with the silk screen.

☐ Next install the two crystals.
Ensure that the 20 MHz crystal is the one nearest the 18 pin IC.

☐ Next install the IC sockets.
Ensure that the notch on the socket lines up with the notch on the IC outline on the PC board.

Do not plug the chips into the sockets at this point.

☐ Next install the rest of the capacitors. With C1, make sure that the longer lead is placed in the hole marked with a +.

☐ Next install the two potentiometers (R6 and R7).

☐ Install the transistor. Ensure you are installing the transistor rather than the voltage regulator… they look a lot alike. Ensure that the flat side of the transistor lies up with the flat side of the part outline on the PC board.

☐ Now install the voltage regulator. It goes in the three holes above C1. Install it so that the flat side of the regulator faces away from U3.

☐ Now install the 9 pin D-Sub connector. Ensure that you push it all the way in so that it is flush against the board. In addition to soldering the pins, you'll gain mechanical stability by soldering the pins that go into the large round holes on the sides of the connector. It is a good idea to clip off the 2 pins as they come through these holes to minimize the chance of the pins shorting on the Pi's USB connector. You might also want to put a piece of insulating tape on top of the USB connector just to be on the safe side.

Note: The D-Sub connector is optional. You can install the Radio header to connect your radio to the TNC-Pi instead. See the section, Connecting the Radio, for more information.

☐ Next install the two LEDs. Ensure the shorter leads on the LEDs go through the holes closest to the flat side of the LED outline.

If you like, The LEDs can be installed with bent legs so the LEDs point toward the front of the board. This makes them easier to see when TNC-Pis are stacked one on top of the other.

☐ Solder in the 2 x 20 pin header. This part is somewhat tricky. It is the only part that is installed through the bottom of the board.

a. Lower the TNC-Pi board onto the connector so that the body of the connector is on the BOTTOM of the TNC-Pi board.

b. Now you'll need to solder the board about a millimeter from all the way down in order to prevent the board from bumping into the USB connector on the Pi. It is not necessary to solder all of the pins. You should at least solder the 4 pins in the corners (for stability) and the first five pins on each row (pins 1 – 10).

These pins will allow you to stack a second TNC-Pi on top of the first one if you choose to do so.

☐ I've included 2 4-40 screws and a 5/8" spacer. You can use this to provide some additional mechanical stability by putting it between the hole in the Pi and the hole in the TNC-Pi. If you are stacking 2 or more TNC's you might also find it preferable to use a spacer that is a male to female, rather than female to female. These male to female spacers are available on the TNC-Pi website.

☐ Leave the board connected to the Raspberry Pi and power up the Pi. Check the voltage between pin 5 (negative) and pin 14 (positive) on U3. It should read about 3.3 volts. With the notch at the top of the chip, these pins are the ones half way down the left side (negative) and right side (positive) of the chip.

If the voltage check is not successful, find and fix the fault before proceeding.

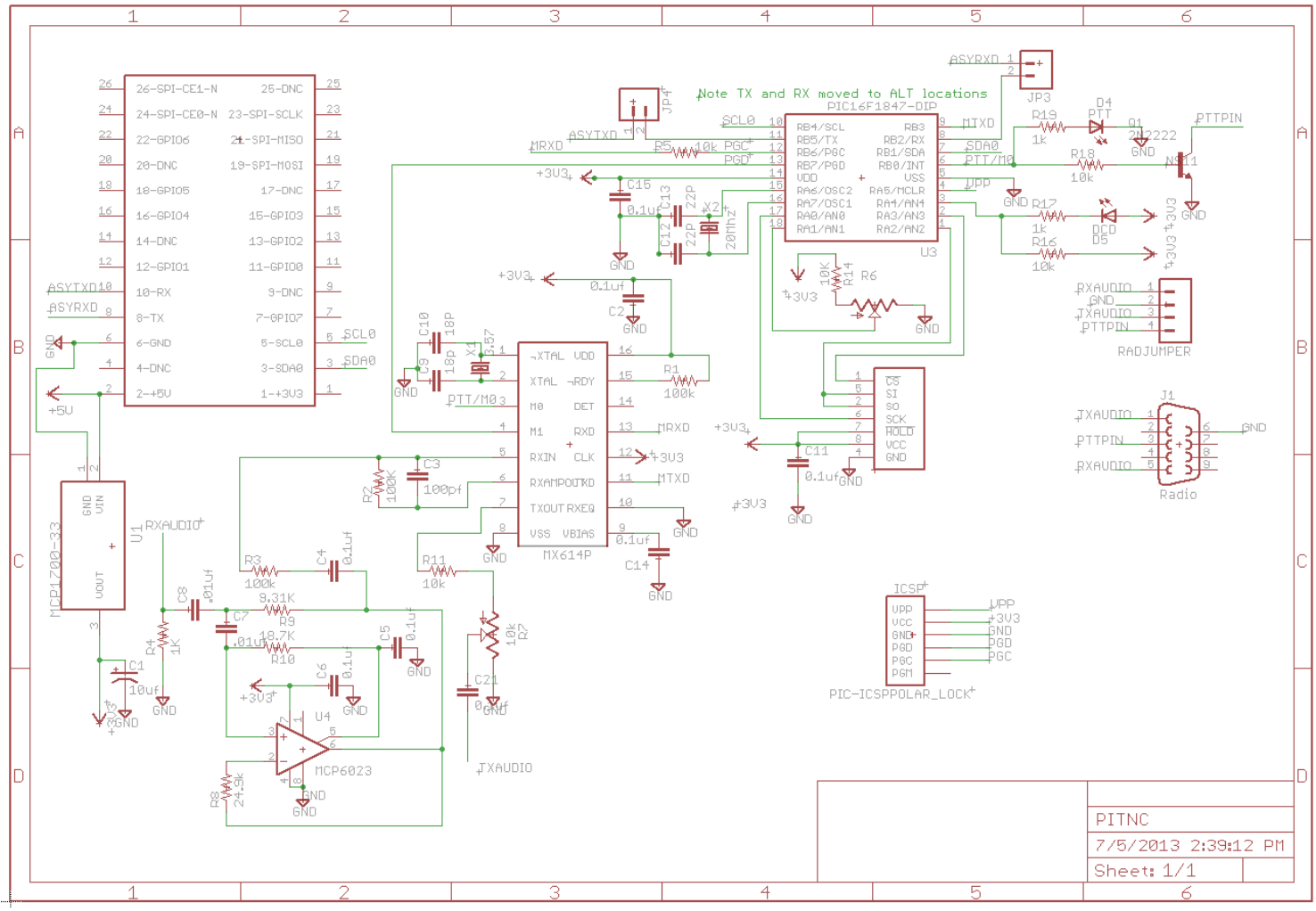
☐ Power down the Pi and remove the TNC-Pi board from the Pi, then install the 4 ICs. You may wish to bend the pins of the four ICs slightly inward to facilitate inserting them into their sockets.

Ensure the notch on the top of each chip lines up with the notch printed on the PC board. Also make absolutely certain that you do not mix up the two 8 pin chips and plug them into the wrong sockets.

Note: Nothing will be installed at the 6 pins marked **ICSP**.

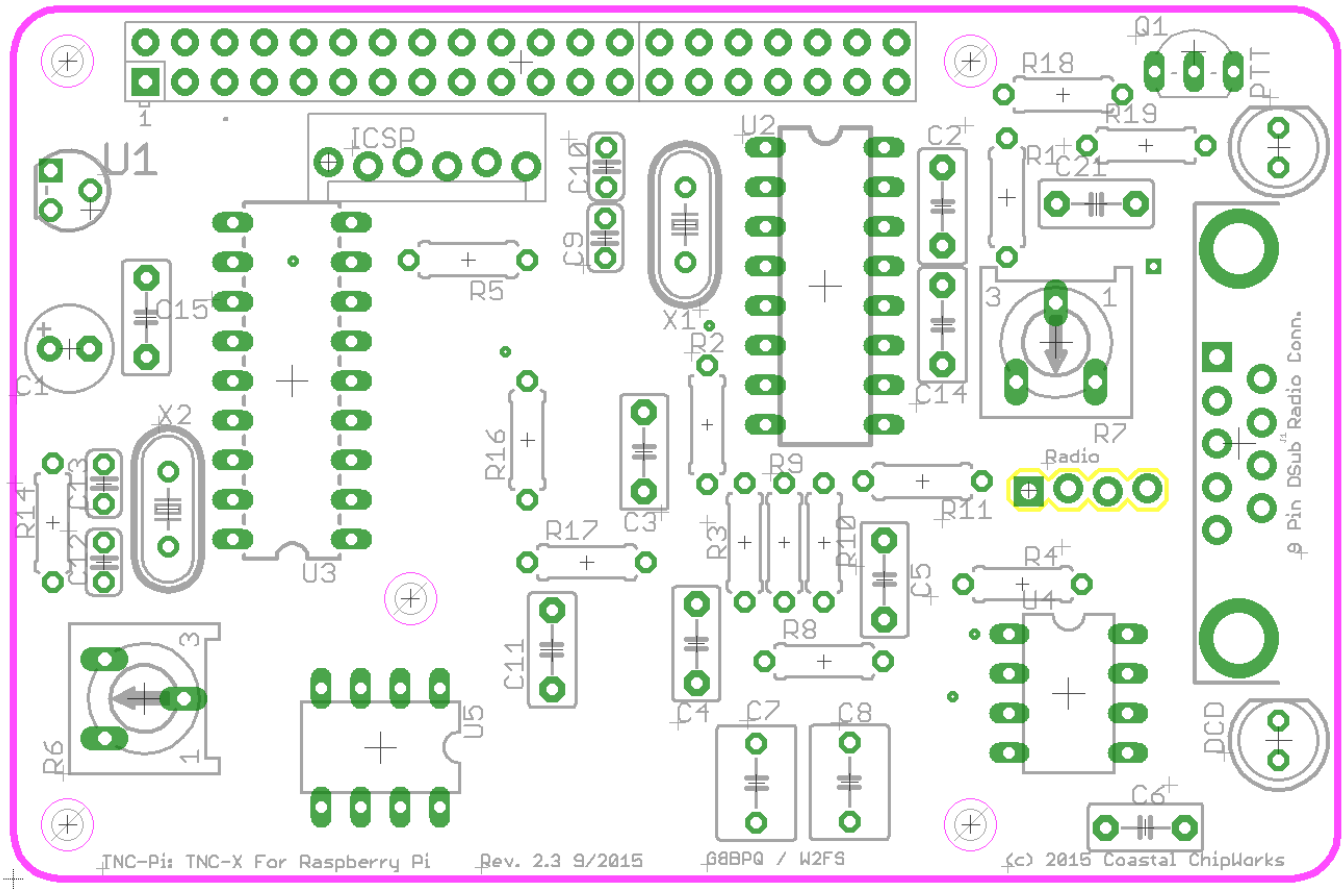
Congratulations, you're done assembling the TNC-Pi.

Schematic Diagram



(see next page for parts layout diagram)

Parts Layout



Installing and Configuring the Raspberry Pi OS

The following instructions were based on a downloaded image of the "Stretch" version of the operating system. I did this by downloading Raspbian, not NOOBs.

To install the Raspian Operating System on your Raspberry Pi, you'll need to obtain a micro SD card and install the OS image on it. I've done this by downloading the Raspian image to my Windows computer (it is also possible to do this with a Mac) and then copy the image onto the micro SD card. You can obtain an up to date image file here:

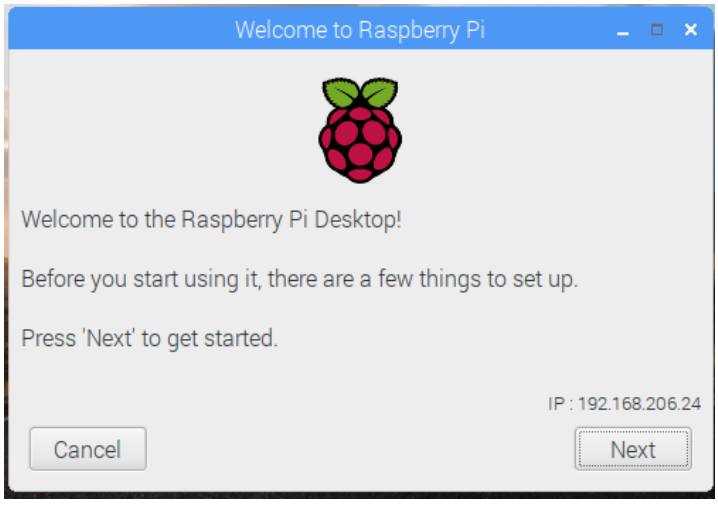
https://www.raspberrypi.org/downloads/raspbian/

Be sure and get the zip file version that says "Raspbian Buster with desktop and recommended software." This is a large file and it will take a while to download. There are several good programs for copying the ISO image to your SD card. I use the Win32DiskImager program which you can find here:

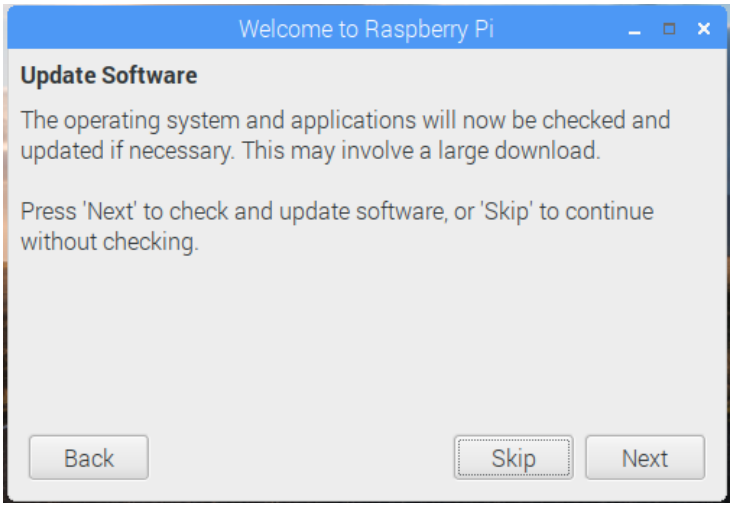
https://sourceforge.net/projects/win32diskimager/

Make absolutely sure you are copying the image (.iso) file onto your SD card, not onto your hard drive. This program will overwrite whatever is on the drive you specify!

Place the SD card into the Pi and boot it up. As soon as it boots the first time this box pops up.



This is a wizard to configure your Pi to your location. It will ask about language, network, etc. Work through the menus choosing the configuration for your location.



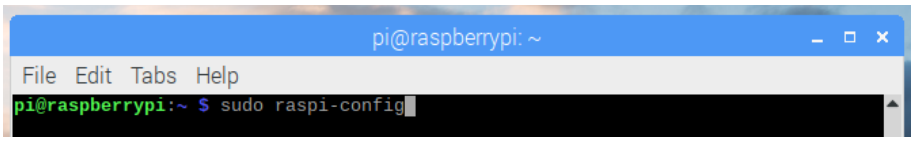
When you get to this screen click on "Next". Now is the time to walk away and grab a snack and the beverage of your choice. The updates usually take a while.

To use the serial port for the TNC-Pi, we'll need to change the Pi's configuration so it will support this. To do this you will have to open a command line window by clicking on the terminal button (the black box at the top toward the left of the screen).

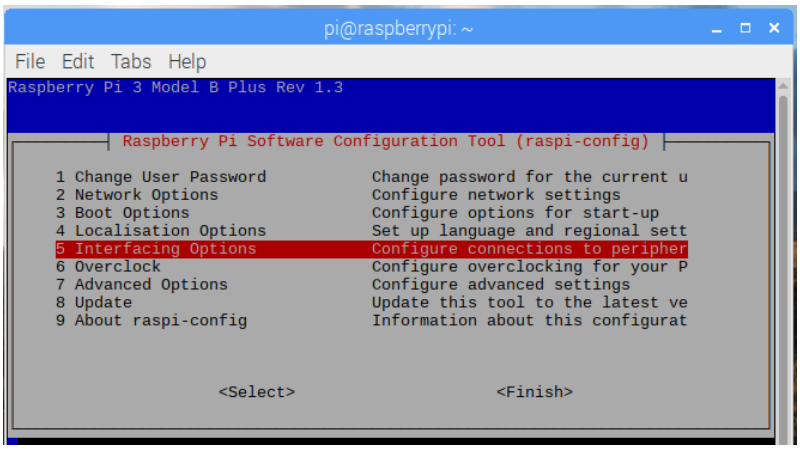
Type in:

sudo raspi-config

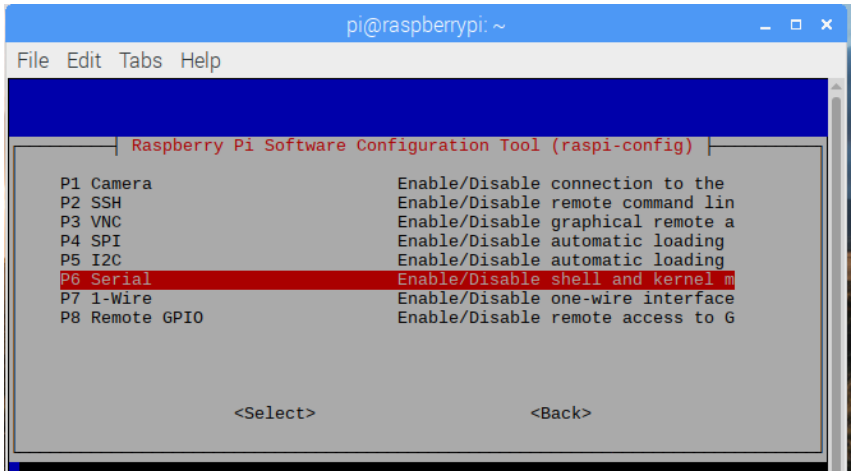
Remember that Linux is case sensitive. Sudo tells the system that you want to run the following command as "root" or administrative user.



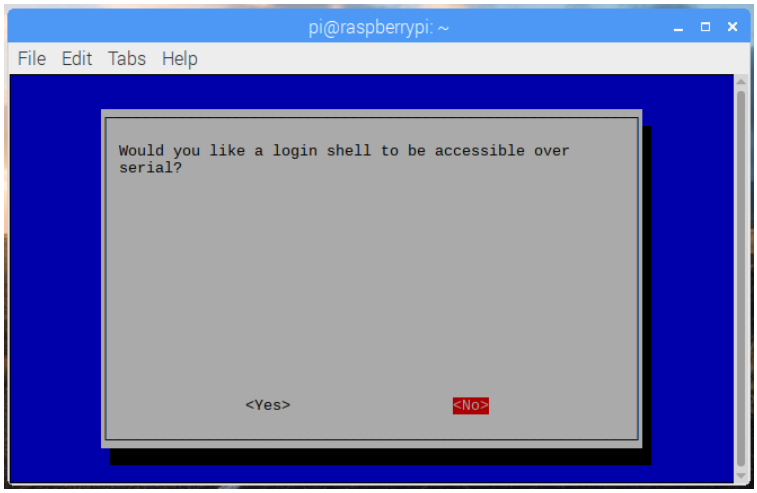
The following box will appear:



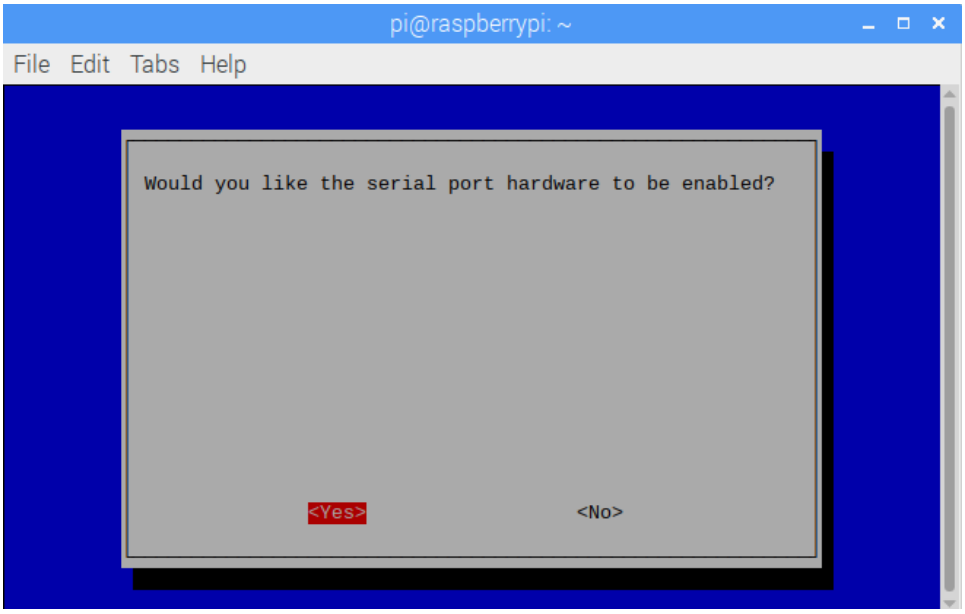
Use the cursor keys to scroll down to Item 5, Interfacing Options. Then, on the next screen scroll down and select P6 Serial.



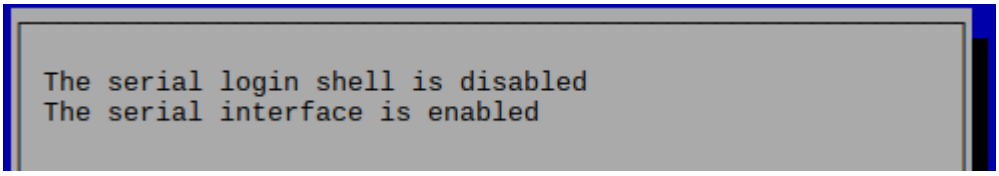
When you do this you'll get the following:



Using the "TAB" key highlight No. Press Enter. You will then see:



Tab to highlight Yes. Press Enter.



Tab to highlight finish. It will ask to reboot. Choose yes.

It might be a good idea to make sure your Pi operating system is up to date. You can do this from the command line by entering the following (this assumes you have either hard wired or wireless network connectivity):

sudo apt-get update
sudo apt-get upgrade

With previous versions of the operating system, it was necessary to edit some of the configuration files in order to get the serial part to work. With the Stretch version of Raspian, this is no longer necessary.

Ok, there is one catch. When you address the serial port you have to do it as /dev/serial0. Using the older version of the operating system it was called /dev/ttyAMA0.

Note that these configuration instructions work with any version of the Pi.

Note added 12/4/2018: Some individuals have reported that there is a problem with the November, 2018 revision of Raspbian that results in the serial port not being accessible. You would notice this, for example, if you run pitnc_getparams and you keep getting "retrying…" instead of the menu. In this case you need to edit the config.txt file in the /boot/ directory. You can do this by running the following from the command line:

sudo leafpad /boot/config.txt

Add the following line at the bottom of this file:
dtoverlay=pi3-miniuart-bt

This problem seems to be fixed with more recent versions of the Stretch version of Raspian.

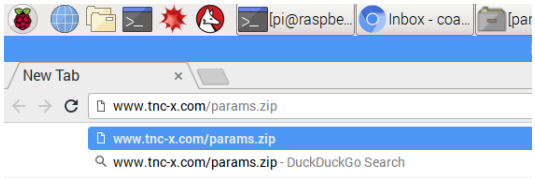
Using pitnc_getparams and pitnc_setparams

The programs pitnc_getparams and pitnc_setparams are used to display and change parameters stored in the TNC-Pi board, including the TXDelay and the I2C address. The pitnc_getparams program will also provide you with a convenient way to make sure that your TNC use successfully talking to the Raspberry Pi.

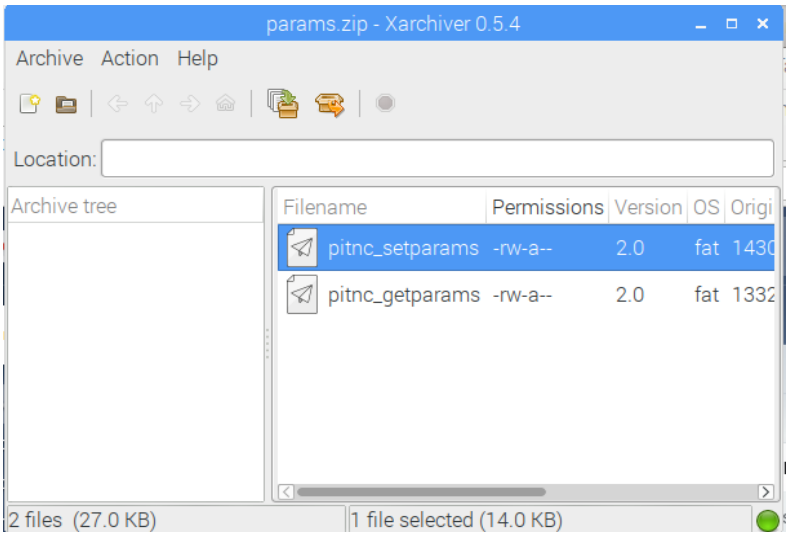
You will need to download these programs. To do you will need to start a web browser. You can do this by clicking on the globe icon in the top left corner of your screen.

Then type in the following:

www.tnc-x.com/params.zip

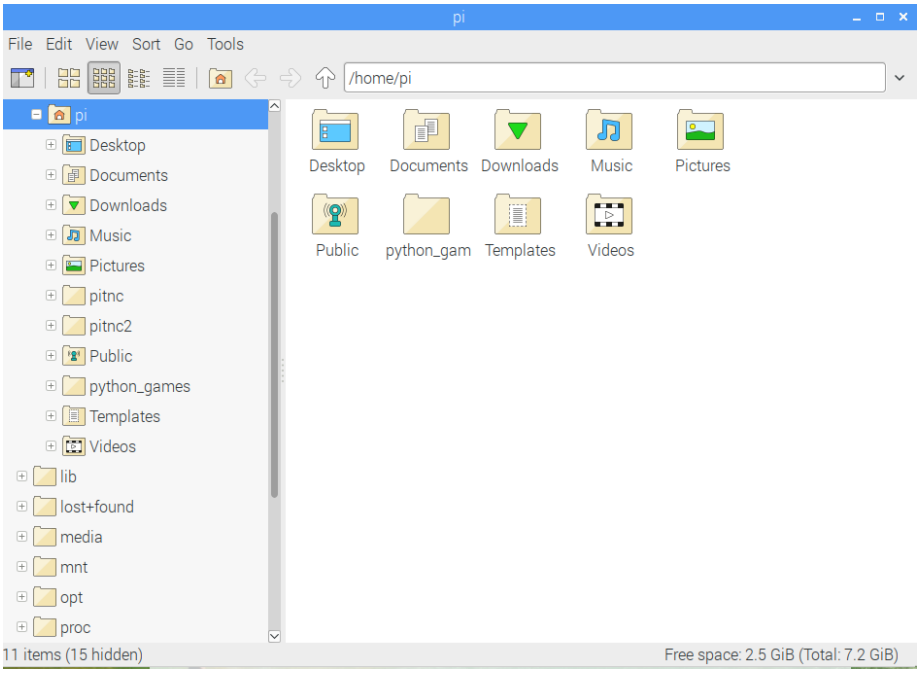


This will download the params.zip file which will appear at the bottom of your screen. Double click on this zip file and its contents will be displayed.

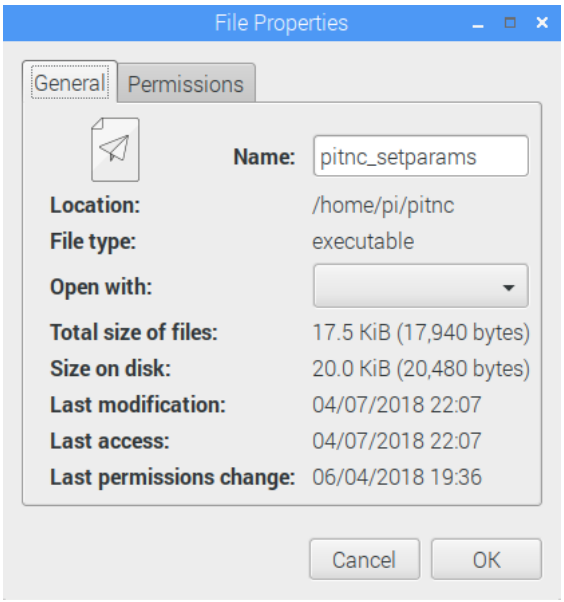


Next click on the file folders icon at the top of your screen to open File Manager program.

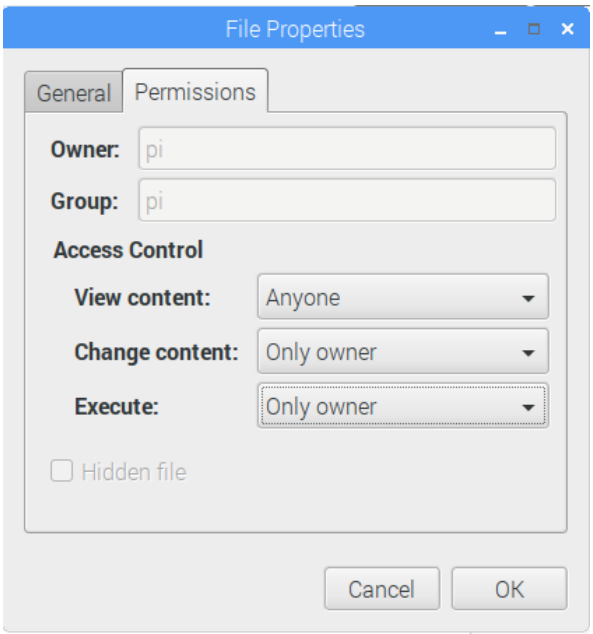
The contents of your home (Pi) directory will be displayed.



You will need to create a new folder to hold these programs. You can this by right clicking on the background of the home directory and selecting Create New from the menu. Create a new folder and give it a convenient name (I use pitnc). Then double click your new folder to open it. You will then be able to drag and drop the pitnc_getparams and pitnc_setparams programs from the zip file to this new folder. Right click on the pitnc_getparams file after you have moved it and select Properties. This will open the File Properties dialog box.



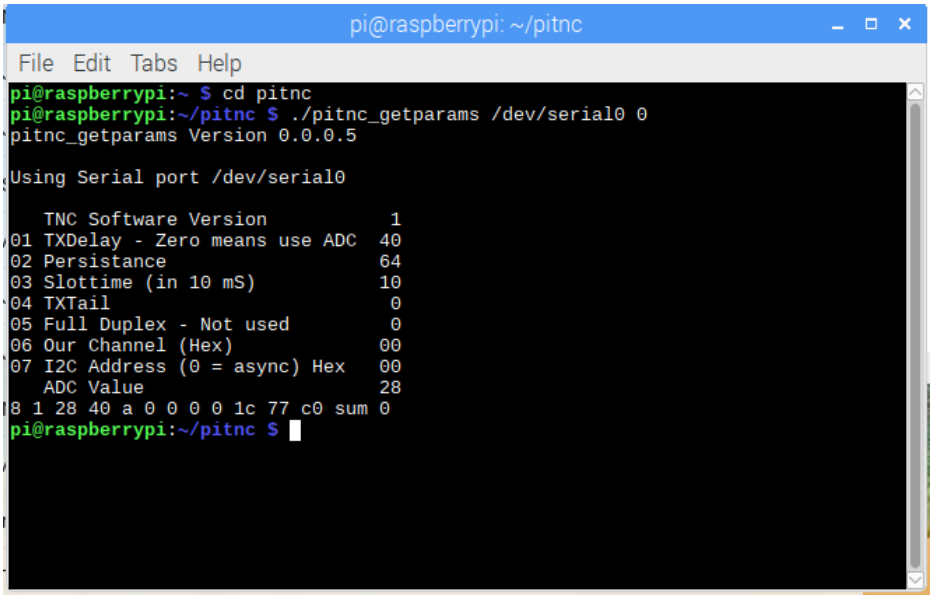
Select the Permissions tab. Change the execute setting from Nobody to Only Owner, and press the OK button. Repeat this procedure with the pitnc_setparams program.



Now open a command line window and switch to the directory that you placed these programs in by using the cd command (see screen below). Run the following command:

./pitnc_getparams /dev/serial0 0

The ./ at the beginning of the line tells the Pi to look in the current directory. If you don't use this, the Pi will not find the pitnc_getparams program. A menu should come up listing some parameters. If this happens, your Pi and TNC are talking to each other.



If you just get a bunch of lines that say:

retrying …

The there is either an issue with your TNC or the way the Pi is configured.

Assuming all is well, you can now set the TXDelay. This is optional, generally speaking the default value will work just fine. If you choose to alter it you can do it with the pitnc_setparams program as follows:

./pitnc_setparams /dev/serial0 0 1 50

This will set the value of parameter 1 (TXDelay) to 50. The txdelay can be set anywhere from 0 to 255. **If you set TXDelay to 0, it will result in the TXDelay being set using potentiometer R6 rather than the value in this menu.**

After you have made a change in the pitnc_setparams program, you must reboot the TNC in order for it to go into effect. You do not have to reboot the Pi to do this, just enter the following command:

./pitnc_setparams /dev/serial0 0 15 2

Connecting the Radio

You can either wire up a 9 pin D-Sub plug to mate with the one on the TNC-Pi, or, if you'd prefer, you can use the four holes below **R7** labeled **Radio** to hard wire a radio connection. (No header is provided in the kit for this.)

If you use the 9 pin D-Sub connection, it should be wired as follows:

Pin 1 (the square pad): TX Audio
Pin 3: Push to Talk (PTT)
Pin 5: Receive Audio
Pin 6: Ground

If you are using the holes marked "Radio" the connections should be:

Pin 1 (the square pad): Receive Audio
Pin 2: Ground
Pin 3: TX Audio
Pin 4: Push to Talk (PTT)

Adjust transmit audio output

Potentiometer **R7** adjusts the level of the transmit audio output. Most people will find that they need to set it fairly near the minimum setting.

One way to set this is to use two radios, one to monitor the transmitted signal and the other connected to the TNC-Pi.

1. Key the radio connected to the TNC-Pi manually by pushing the PTT button on it.
2. On the other radio you will hear a continuous tone (even though no data is being transmitted… you don't need to be running any software on the Raspberry Pi to do this).
3. Turn R7 all the way down and the tone will go away.

4. Then slowly turn it up until the volume doesn't increase any further in the monitor receiver. When you reach this point, back it off just a little and you should have it about right.

A second way to have the TNC-Pi to send tones is to use the following command:

./pitnc_setparams /dev/serial0 0 15 3

This will cause the TNC to send 2200 hz and 1200 hz tones for about 5 seconds each.

Installing YAAC for Doing APRS

Contributed by Peter Klotzback, AF9FA

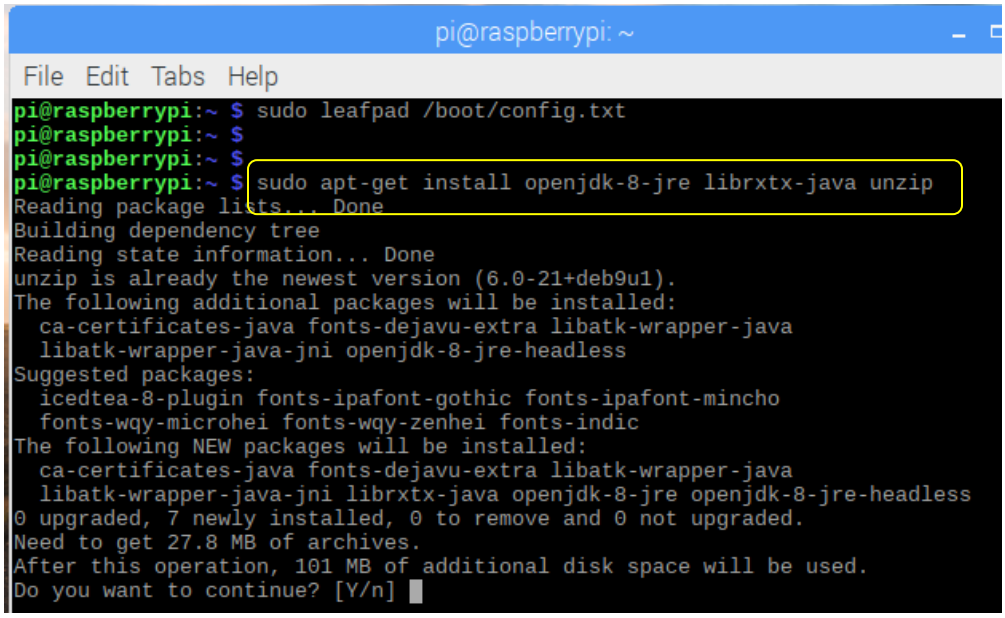
Much of this information is take from the YAAC web site:

(http://www.ka2ddo.org/ka2ddo/YAAC.html) .

Andrew has done an excellent job on this software and site! There is a LOT of good information there! YAAC is a Java based platform so we have to install Java first. In the terminal window enter:

sudo apt-get install openjdk-8-jre librxtx-java unzip

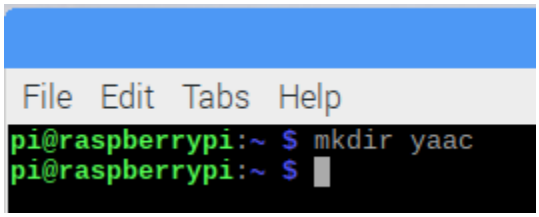
(On the website he mentions that you should change to super user, but it's easier to preface the command with sudo. Remember Raspian / Linux are CAsE seNSiTive.)



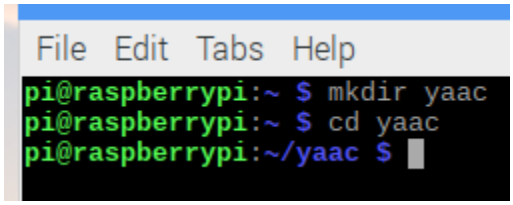
Press enter and the install will begin. This can also take a few minutes.
Java is installed, now we just need to download and install the YAAC software. Open the Chromium Browser.  and enter this in the address bar:

http://www.ka2ddo.org/ka2ddo/YAAC.zip

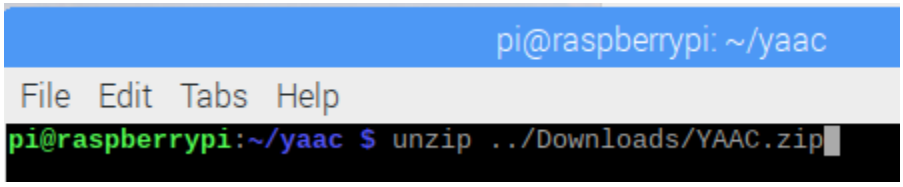
This will download the install zip file to your home/Downloads folder. Note: this is the same procedure you used to download the params.zip file. Open a terminal session and create the new directory.



Then change to that directory.



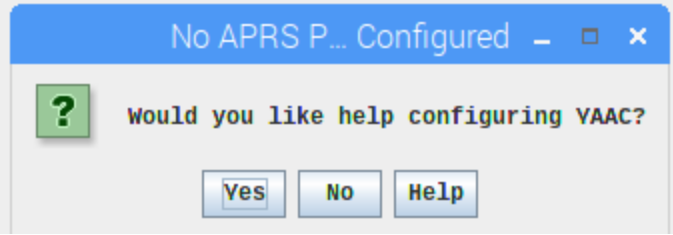
In the terminal window enter
unzip ../Downloads/YAAC.zip
this unzips all of the required files into your new directory.



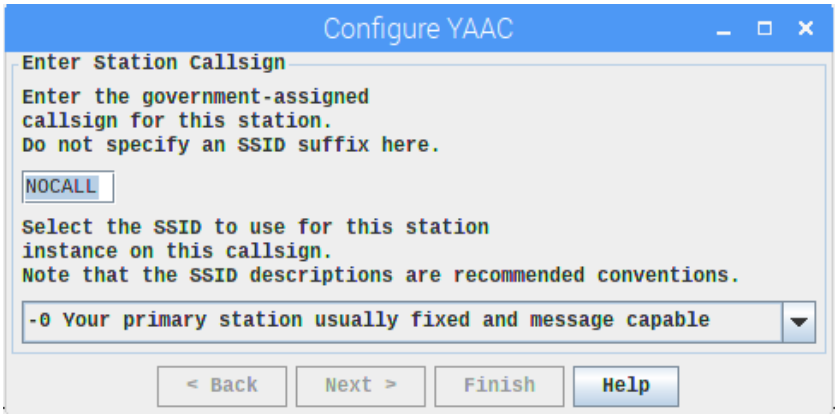
At this point, YAAC is installed and ready to run. Use the command:

java -jar YAAC.jar

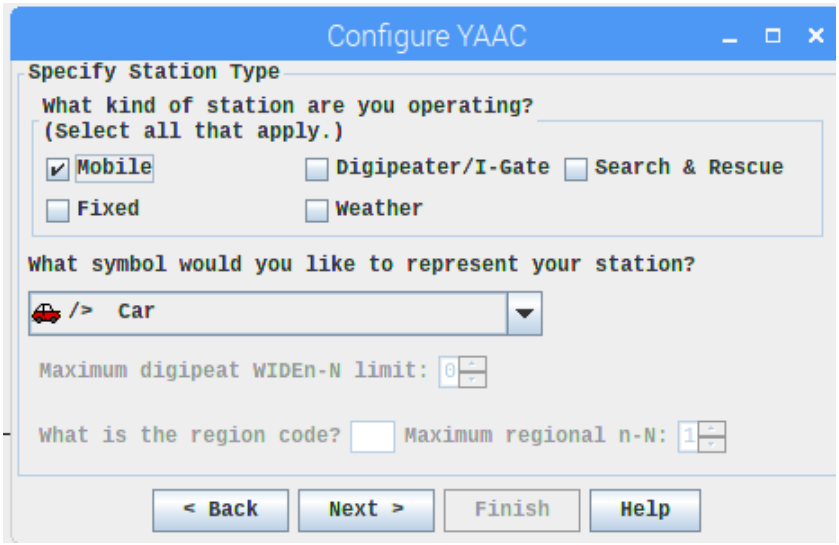
This will start YAAC and begin a configuration wizard.



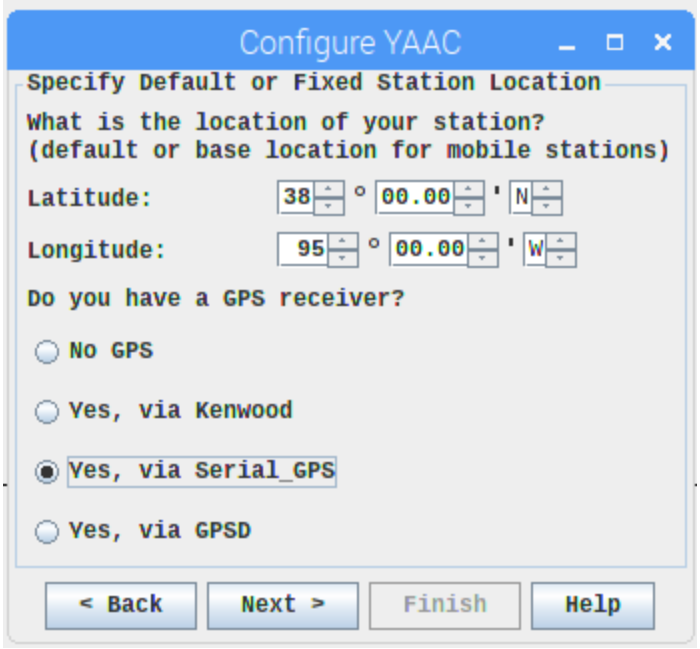
This is a good place to start and get the generic information loaded.



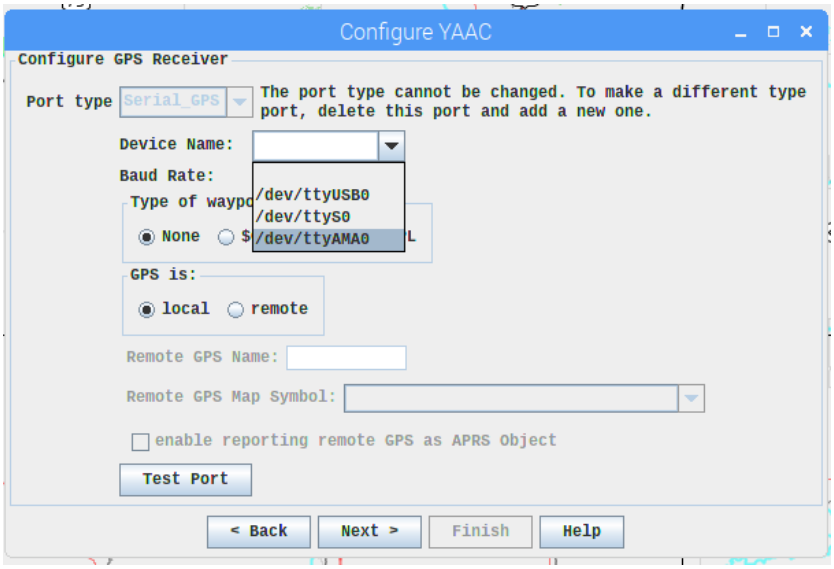
Enter your call and an SSID. Again the YAAC website and WWW.APRS.ORG are excellent resources If you're not sure what to use.



Next enter the type of installation you're going to use. In my case I chose a mobile install.

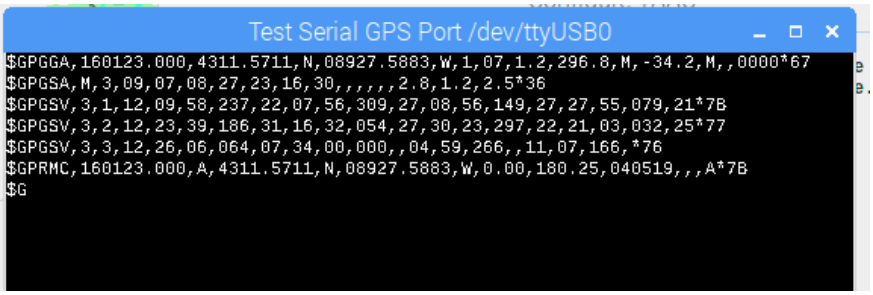


I'm using a generic USB serial GPS.

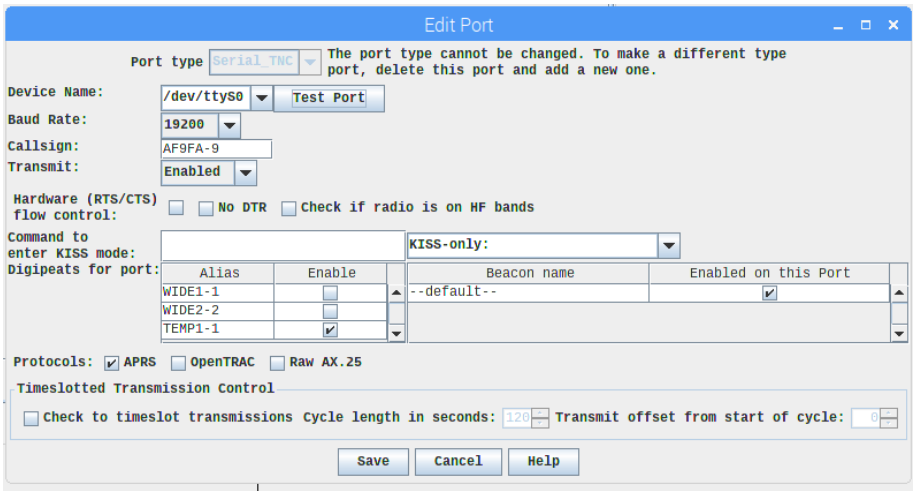


My favorite part of this software is that you don't have to be a linux genius to use it. Instead of choosing from 30-40 different ports that may or may not have anything attached, it limits it to what it knows about. In this case it's the /dev/ttyUSB0 because that's where the GPS is plugged in. This configuration does not use GPSD which is the cause of a lot of headaches.

Click on the port and then click "Test Port". You should see this:

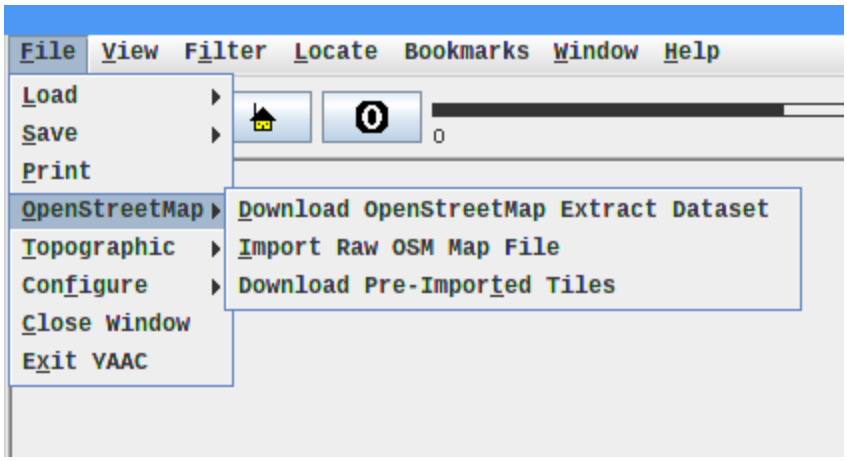


Next we'll set up the connection to the TNC-Pi. Select the port and Baud rate.

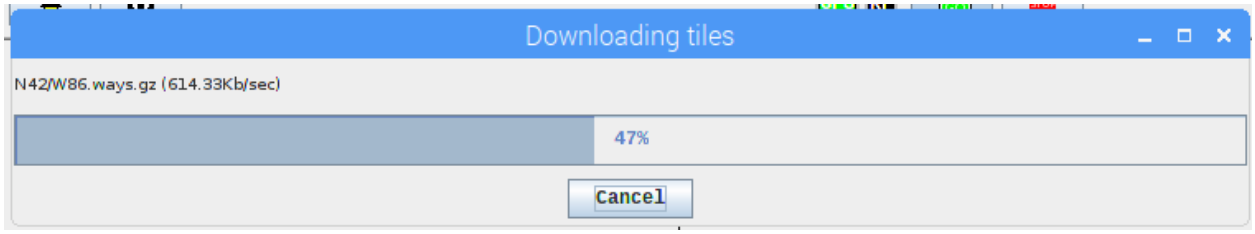


Make sure the rate is set to 19,200. That 's the comm speed used by the TNC-Pi serial connection. The packet speed is 1200 Baud. When you choose the serial port, the laws of the universe state that ALL APRS traffic must stop. your normally noisy radio will go dead silent while testing. I just choose the port and click "Save". There are many more settings under File>Configure. This is documented well on Andrew’s website.

You are just about finished. You can pull down maps to store locally. If you do this an internet connection is not required to display the maps after the initial download.



To begin to download the maps, click on the pre-imported tiles. it will bring up a menu for what distance you want and you can filter the types of points as well. Be patient, downloading the tiles can take a while. once they are downloaded they also take some time to 'render', or make useable to the software. You may have to restart YAAC to bring them up. trust me. they'll show up.



The one part of the whole setup that I was unable to find was how to force a beacon packet to test the transmitter link. You can test the transmit link by clicking a blank space in the map and hit the spacebar. This sends a beacon packet to the radio. Then look at www.APRS.fi and see if you show up.

Again, this is just a generic configuration to get your system running. PLEASE check the websites and find out what other cool stuff this hardware and software will do!

Using TNC-Pi for Winlink

APRS and Winlink are the two most widely used applications in packet radio today. Most people who are using APRS with the Raspberry Pi are using either YAAC or Xastir. Winlink is used extensively for sending email over packet radio. It is often used by people who are in areas without an available Internet connection, for example, maritime mobile stations and those who want to have this capability for emergency communications systems. While Winlink was originally designed for computers running the Windows OS (that's where the "Win" comes from) there is a version of this software that is supported on Linux called paclink-unix. There is a

group that supports this on groups.yahoo.com called paclink-unix. Mark Griffith, KD0QYN, has developed an easy to use email system that is designed specifically for the Raspberry Pi and TNC-Pi called PiGate. You can find information on this at:

www.pigate.net.

Keyboard to Keyboard Connections

While keyboard to keyboard communication was very popular in the 1980s and 1990s, it is now mostly gone. There still is software left from this time period that will support this called ax25-apps. This software is buggy and not particularly easy to use. The following is a description of the basics of getting this going. In addition I would recommend doing a search on "linux ax25-apps" for more information. **Note: unless you have a particular need for this type of communication and are willing to do a fair amount of experimentation, I recommend simply skipping this section.**

The first step to getting this going is to install the ax25 apps and tools. From the command line enter the following:

sudo apt-get install ax25-tools
sudo apt-get install ax25-apps

You'll need to configure it for your callsign by editing the **/etc/ax25/axports** file. You'll find two lines that allow you to enter your callsign. I did so as follows:

1 W2FS-1 19200 236 2 TNC 1
2 W2FS-2 19200 236 2 TNC 2

The serial rate must be at **19200**, because that is the only baud rate supported by the TNC-Pi. The next two values (**236** and **2** in the above example) are the values for **paclen** and **maxframe**. **Don't leave any blank lines in this file.**

Now attach the serial port to the AX.25 system using **kissattach**:

sudo kissattach /dev/serial0 1 10.1.1.1

The number in blue above matches the port number from the axports file above. The number in red is an IP address. It's required here even though you aren't using the IP protocol on it.

Note: If you are using the IP protocol on your TNC-Pi your address should conform to the local convention for IP routing. The address in that case will almost certainly start with 44.

You can monitor packets on this channel by entering:

sudo axlisten –a

You can connect to another station for keyboard to keyboard QSO's as follows:

axcall 1 hiscall

where **hiscall** is the call station of the station to which you want to connect.

There seems to be a bug in the axcall routine. The first time you use it after calling kissattach, it will take significantly longer for the Pi to send a valid connect string to the TNC than it does in subsequent attempts. You may have to wait 10 – 15 seconds or longer. Further attempts occur instantaneously. You can abort this first try by issuing a **Ctrl-C** and then issuing the **axcall** command again. It will then connect immediately.

Another (BETTER) approach is to use the linBPQ program written by John Wiseman, G8BPQ. This program is the Linux version of the BPQ program that was very widely used back in the 80s and 90s by hams who wanted to support Bulletin Board systems and/or building a network of packet switches. You can also use this to allow the TNC-Pi to work somewhat like other TNC's that are not KISS mode only using a terminal program such as minicom. You can find information concerning this here:

http://www.cantab.net/users/john.wiseman/Documents/LinBPQGuides.html

Configuring TNC-Pi for Use with the I2C Protocol

The TNC-Pi can be configured to communicate with the Raspberry Pi using the I2C protocol. The only reason you would ever want to do this is if the Raspberry Pi serial port was already in use. For example, if you already have a TNC-Pi stacked on a Raspberry Pi and you want to add one (or more) TNC-Pi's to the same Pi, you'll need to address those additional TNC-Pi's using I2C. The general procedure for doing this is to start with each TNC set so that it communicates via the serial port (this is the factory setting) and then configure each additional TNC to have a unique

I2C address on it. You would need to set these boards one at a time because only one serial TNC-Pi can be on the TNC-Pi at a time.

To configure the TNC-Pi for I2C, you will need a configuration programs **pitnc_setparams** and **pitnc_getparams** as described in an early section of this manual. The **getparams** program reads the parameters from the TNC-Pi while the **setparams** program allows you to set them.

Before using them, however, it will be necessary to make some additional configuration changes to the Pi. First, make the following changes to the following files:

1. In the **/etc/modprobe.d/raspi-blacklist.conf** file, remove the line: **blacklist I2C-bcm2708**
2. In the **etc/modules** file, add the line: **i2c-dev**

Note: Before running the **getparams** and **setparams** programs, ensure that **kissattach** is not running.

You may recall that the serial mode format for the pitnc_getparams program was:

./pitnc_getparams /dev/serial0 0

And the format in serial mode for the pitnc_setparams program was:

./pitnc_setparams /dev/serial0 0 p v

Where p is the parameter that you want to change and v is the new value of that parameter. So if you want to change the I2C address from 0 to, say 3, you would enter:

./pitnc_setparams /dev/serial0 0 7 3

After you reboot the TNC, it will now be communicating over the I2C interface. So to then run pitnc_getparams you would need to enter:

./pitnc_getparams 1 3

Here 1 is the number of the I2C bus and 3 is the I2C address that you have sent. To change a value when the TNC is communicating with I2C, you simply add two more parameters: the parameter number you want to change and the new value.

Note: On Version 1 Pi boards (without mounting holes) the I2C bus number is zero, for the Version 2 boards it is 1.

The BPQ software for Raspberry Pi is currently in beta and can be found at:

http://www.tnc-x.com/InstallingLINBPQ.htm

If you ever need to reset these parameters to their original factory values, this can be done by powering down, turning the TXDelay potentiometer all the way to minimum and then powering back up. You should see the yellow LED flash once per second. When you see it flash, you'll know that the parameters have all been reset. The power the device back down, move the TXDelay off of minimum and power the device back up.

Running Applications other than LinBPQ with TNC-Pi in I2C mode.

John Wiseman, G8BPQ, sends along the following information about running other applications using I2C:

The TNC-Pi can be used with applications that use the Linux ax.25 stack, or applications that expect to see a KISS TNC on a serial port. Program I2ckiss converts the I2C protocol to a standard KISS presentation on a virtual serial (pty) port. It is available here:

www.tnc-x.com/i2ckiss.zip

One copy is run for each TNC-Pi. The first two parameters to I2ckiss are I2C bus, I2C device. If using the kernel ax.25 code, then specify the port number (from axports) and the ip address. To use with other software, specify symlink and a symbolic name - I suggest com1 - com255

For example, to use with the Linux ax.25 stack:

sudo ./i2ckiss 0 16 1 10.1.1.1

i2ckiss will create a **pty** pair, and execute kissattach on the slave half, using the 3rd and 4th parameters

To use with a KISS application

./i2ckiss 0 16 symlink com1

i2ckiss will create a **pty** pair, then create a **symlink** to com1. The application would then be configured to use port **com1**.

Support

If you have any questions about your TNC-Pi or are having hardware issues with it, please contact John Hansen, W2FS at john@coastalchip.com. Software issues, particularly with regard to the Linux version of BPQ are best addressed to the author, John Wiseman, G8BPQ.