



TNC-Black Assembly Instructions & Operating Tips

By John Hansen, W2FS

With contributions by John Wiseman G8BPQ

Document editing support by

Ed Slingland, N2WD

Website

<http://tnc-x.com/TNCBlack.htm>

Contents

Introduction	3
Assembly Instructions	4
Parts List	4
Assembly	6
Parts Layout.....	9
Setting the Serial Port	10
Using the pitnc_getparams and pitnc_setparams Programs	10
Connecting the Radio	12
Adjust transmit audio output.....	12
Configuration Tips for the Beaglebone Black.....	12
Using linbpq and minicom with the TNC-Black	13
Running Applications other than LinBPQ with TNC-Black in I2C mode.....	16
Please Help Improve this Manual	17

Introduction

Thank you for purchasing a TNC-Black: TNC-X for the Beaglebone Black. This assembly/operating manual is necessarily very preliminary as the product is just being introduced in June, 2014

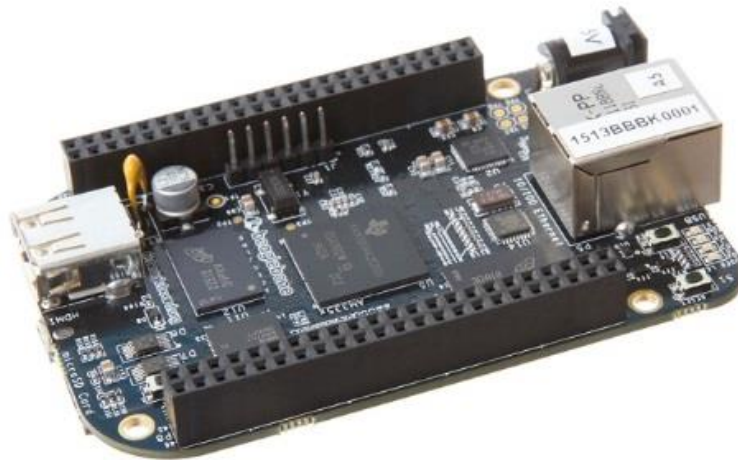


Figure 1: The Beaglebone Black Single Board Computer

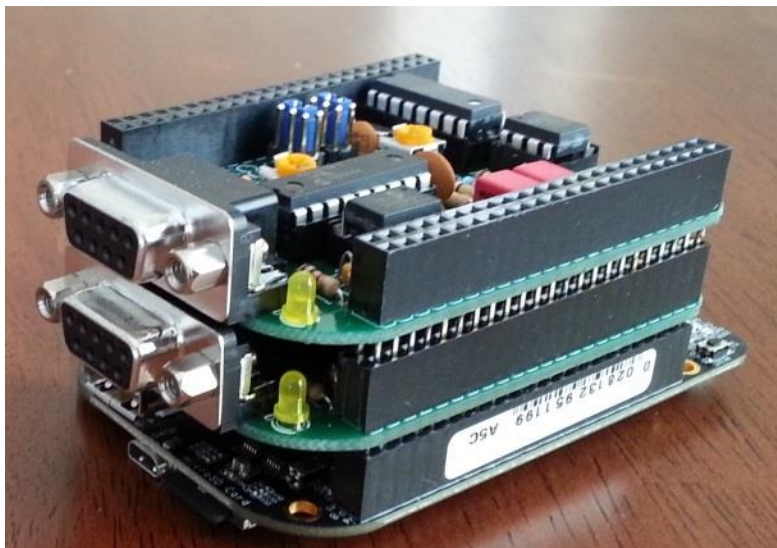


Figure 2: Two TNC-Black's stacked on one Beaglebone Black

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
<input type="checkbox"/>	<input type="checkbox"/>	C1, C2 , C4 , C5 , C6 , C11 , C14 , C15, C21	0.1 μ f monicap	
<input type="checkbox"/>	<input type="checkbox"/>	C7, C8	0.01 μ f 2.5%	Red and yellow or grey; if grey marked 10_n-F
<input type="checkbox"/>	<input type="checkbox"/>	C9, C10	18 pf ceramic disk	
<input type="checkbox"/>	<input type="checkbox"/>	C12, C13	22 pf ceramic disk	
<input type="checkbox"/>	<input type="checkbox"/>	C3	100 pf ceramic disk	
<input type="checkbox"/>	<input type="checkbox"/>	R1, R2, R3	100K resistor	Brown, black, yellow
<input type="checkbox"/>	<input type="checkbox"/>	R4, R17, R19,	1K resistor	Brown, black, red
<input type="checkbox"/>	<input type="checkbox"/>	R11, R14, R16, R18	10K resistor	Brown, black, orange
<input type="checkbox"/>	<input type="checkbox"/>	R8	24.9K resistor	Red, yellow, white, red, brown
<input type="checkbox"/>	<input type="checkbox"/>	R9	9.31K resistor	White, orange, brown, brown, brown
<input type="checkbox"/>	<input type="checkbox"/>	R10	18.7K resistor	Brown, grey, purple, red, brown

<input type="checkbox"/>	<input type="checkbox"/>	R6, R7	10K trimmer potentiometer	Orange or Blue
<input type="checkbox"/>	<input type="checkbox"/>	X1	3.57 MHz crystal	
<input type="checkbox"/>	<input type="checkbox"/>	X2	20.00 MHz crystal	
<input type="checkbox"/>	<input type="checkbox"/>	D4	Red LED (PTT)	Polarized
<input type="checkbox"/>	<input type="checkbox"/>	D5	Yellow LED (DCD)	Polarized
<input type="checkbox"/>	<input type="checkbox"/>	Q1	PN2222 transistor	3 pin, flat side
<input type="checkbox"/>	<input type="checkbox"/>	U2	CML MX-614 Modem	16 pin IC
<input type="checkbox"/>	<input type="checkbox"/>	U3	PIC16F1847 microcontroller	18 pin IC
<input type="checkbox"/>	<input type="checkbox"/>	U4	MCP6023 Op Amp	8 pin IC
<input type="checkbox"/>	<input type="checkbox"/>	U5	23K640 Memory	8 pin IC
<input type="checkbox"/>	<input type="checkbox"/>	IC Sockets	For U2, U3, U4, U5	One 16-pin, one 18-pin and two 8-pin sockets
<input type="checkbox"/>	<input type="checkbox"/>	JP1, JP2	2x3-pin header	
<input type="checkbox"/>	<input type="checkbox"/>		2 x 23 extra long headers (qty 2)	
<input type="checkbox"/>	<input type="checkbox"/>		9 Pin D-Sub connector	
<input type="checkbox"/>	<input type="checkbox"/>		Printed Circuit Board	
<input type="checkbox"/>	<input type="checkbox"/>		2 shorting jumpers	

Assembly

- Start by installing the parts that lie flat on the board. This includes:
 - a. all of the 0.1uf monocats and
 - b. all of the resistors except for R6 and R7.

Note that all of the components including the 46 pin headers 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. Also make sure that the 28 pin IC sockets are installed at U4 and U5. **No socket should be installed in the 8 pin outline next to R19**

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

- Next install the rest of the capacitors.
- Next install the two potentiometers (R6 and R7). You may find these to be a bit of a tight fit. If so, take a pair of needle nose pliers and flatten the leads before inserting them into the board.
- Next install the two LEDs.

IMPORTANT: The red LED (PTT) must be installed so that the longest lead is closest to the center of the board where the 9 Pin D-Sub connector will go. The yellow LED (DCD) must be installed so that the shortest lead is closest to the center of the board where the 9 Pin D-Sub connector will go.

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.
- Install the transistor. Make certain that the flat side of the transistor aligns with the flat side of the outline on the PCB..

- 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.

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

- Now install the 2 x 3 jumpers (JP1 and JP2).
- Solder in the 2 x23 pin headers. This part is somewhat tricky because there are so many pins. If you look at the schematic, you will be able to see which pins are actually used by the TNC-Black. These are the only pins that it is essential to solder. Start by soldering just two pins in of each connector, one on each end. Then look to make sure that the connector is fully flush against the PC board. If necessary reheat the solder joint while gently pressing on the connector to get it to snap in flush against the board.

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

Connect the board to the BBB and power it up. 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 BBB and install the 4 ICs.

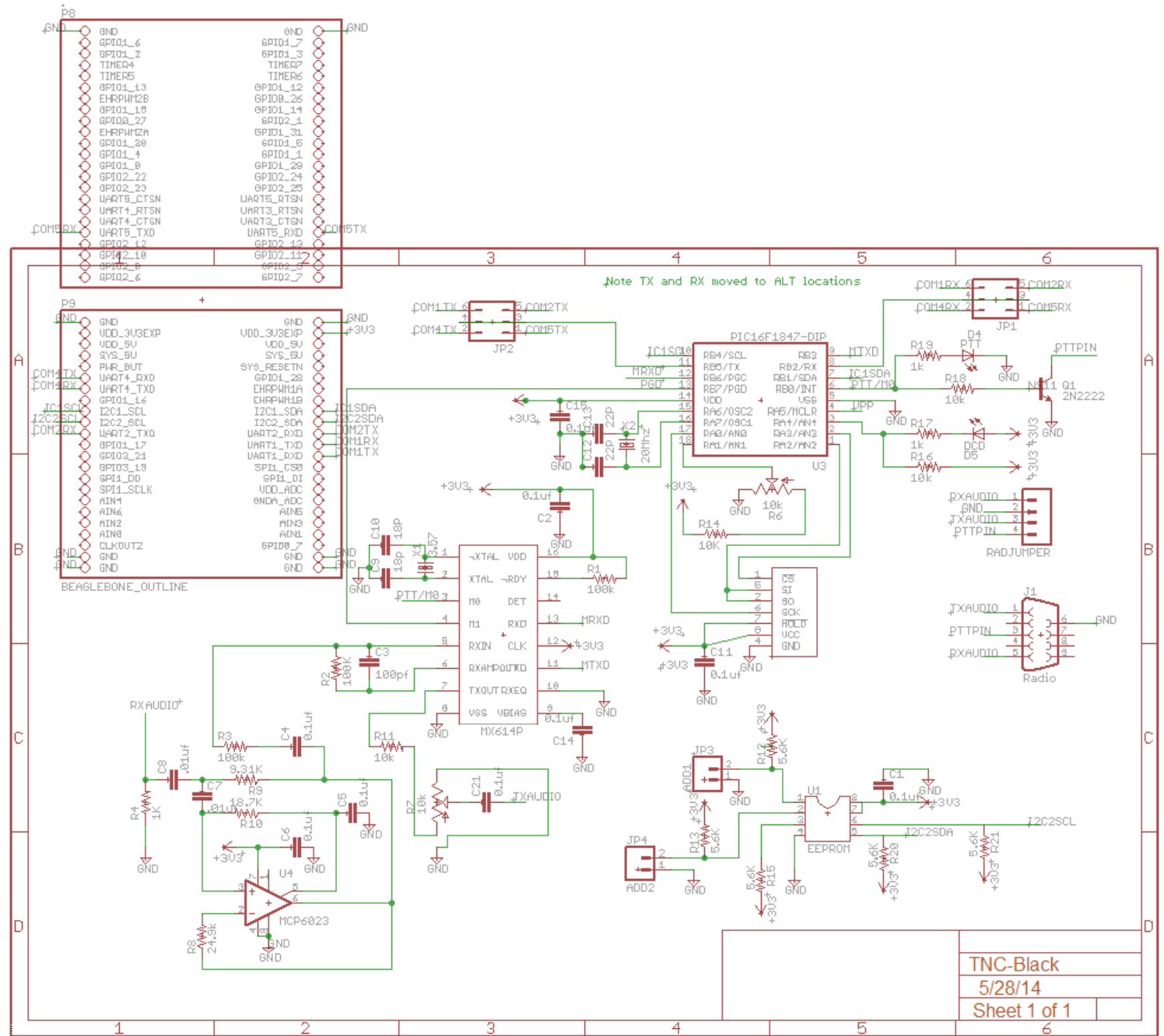
You may wish to bend the pins of the four ICs slightly inward to facilitate inserting them into their sockets.

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.

No parts need to be installed in the 8 pin socket outline that is next to R19, or in the 5 resistor outlines next that socket outline. At one time it was necessary that every BBB cape have an EEPROM to be read by the BBB at boot time. That does not appear to any longer be the case. Hence, these parts are not included in the kit.

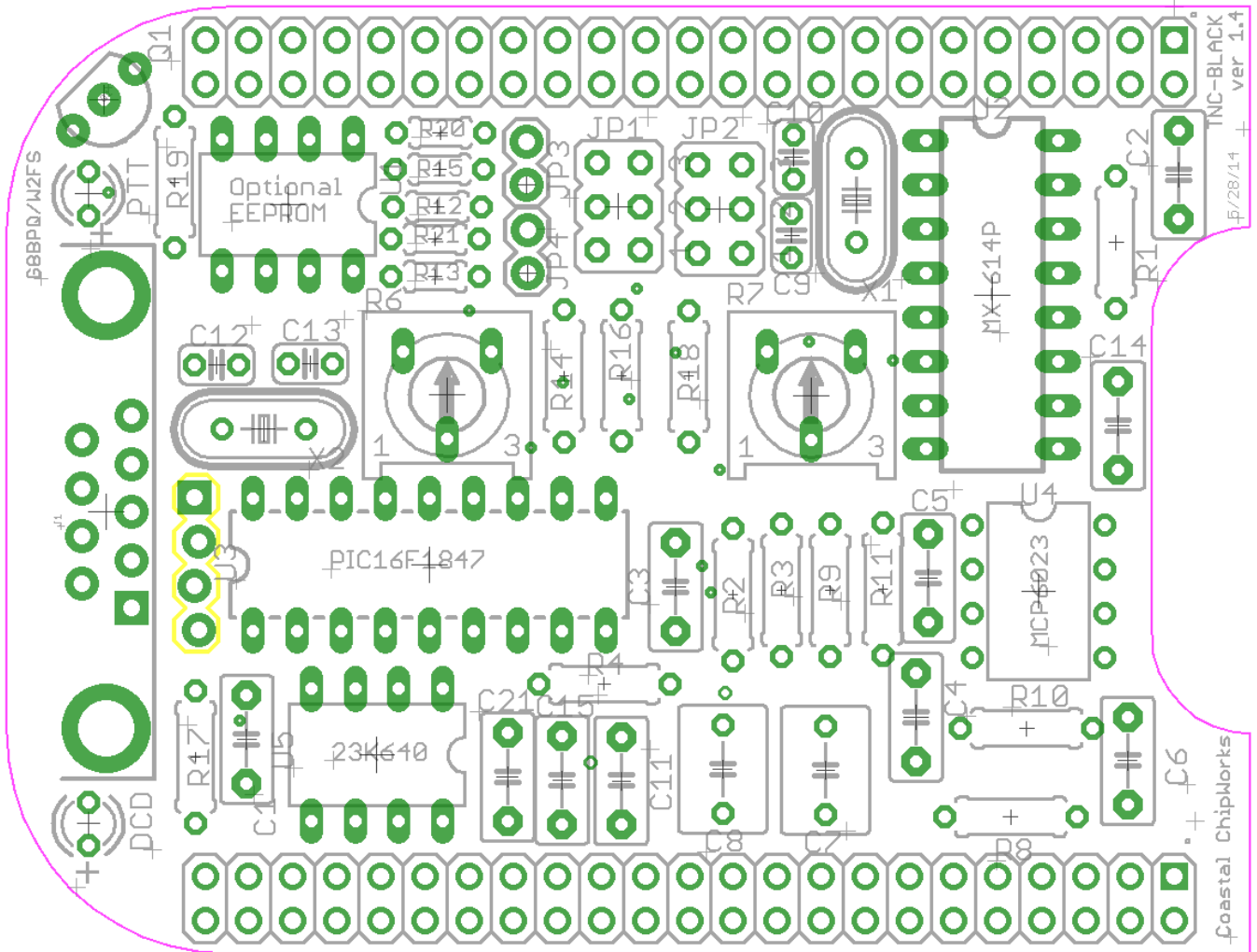
Congratulations, you're done assembling the TNC-Black.

Schematic Diagram



(see next page for parts layout diagram)

Parts Layout



Setting the Serial Port

The TNC-Black can be connected to any one of four serial ports on the Beaglebone Black: UART1, UART2, UART4 or UART5. However, be aware that if you select

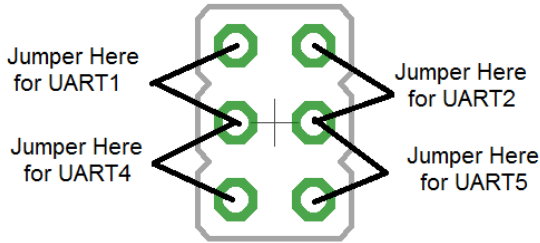


Figure 3: Serial Port Jumper Positions

UART5, the video output of the BBB will no longer work (not a problem if you are running it headless). The four port options are available to facilitate stacking multiple TNC-Blacks on a single BBB. There are two headers on the TNC-Black that allow you to select the port, one for transmitting data and one for receiving. So selecting a port requires that you place one of the included jumpers on each of the headers.

Figure 5 shows how to set up these jumpers for the desired serial port. Communication with the BBB can also be done using it's I2C bus. To do this you must set the I2C address using the `pitnc_setparams` program (see below). Note that the baud rate of the serial port on the TNC-Black is fixed at 19.2K.

Using the `pitnc_getparams` and `pitnc_setparams` Programs

There is non-volatile memory in the PIC16F1847 chip that holds a number of parameters. These parameters can be read using the `pitnc_getparams` program and can be set using the `pitnc_setparams` program. These programs are available at:

www.tnc-x.com/pitnc.zip

As shipped, the TNC-Black is set to use a serial port to communicate with the Beaglebone Black. Because there are four available serial ports, you can stack up to four TNC-Blacks on a single BBB without running out of ports. However, if you run out of serial ports or prefer to use them for something else, you can communicate with the BBB using it's I2C bus. To do this, you must set the I2C bus value to something other than 0.

Furthermore, as shipped the TXDelay is set in software. You can change that and use the potentiometer R6 to set the TXDelay, but to do this you must set the software value for TXDelay to 0. I2C bus and TXDelay are the parameters in the

PIC's EEPROM that you are most likely to want to change. However, most users will probably find that there never is any reason to adjust these.

To review the current value of the I2C parameters run the `pitnc_getparams` program as follows (the `./` below assumes you are currently in the directory that contains the program):

```
./pitnc_getparams X Y
```

Y here indicates either that we are using a serial port (if it is 0) or it is the I2C address. If Y is 0, then X is the serial port number (1, 2, 4 or 5). If Y is non-zero, then X is 1 (the I2C bus number). So for example, since the factory defaults are to use a serial port, if you are using UART1 you should use the command:

```
./pitnc_getparams 1 0
```

The parameters you are most likely to want to change are 1 and 7:

Parameter value	Description	Notes
1	Sets the TXDelay	With TNC-Black you can set the TXDelay either in software using the <code>pitnc_setparams</code> program or in hardware using R6. Setting this parameter to 0 causes the TNC-Pi R6 potentiometer to determine the value.
7	Sets the I2C address	Setting this to 0 causes data communication on the serial port (default), any other value up to 63 will cause I2C to be used with the TNC using this address.

To set a parameter use the `pitnc_setparams` program as follows:

```
./pitnc_setparams X Y A B
```

where X and Y are as above and A is the number of the parameter you want to change and B is the new value. So, for example, to change the TXDelay to 20 if you have specified UART1 you would enter:

```
./pitnc_setparams 1 0 1 20
```

Connecting the Radio

You can either wire up a 9 pin D-Sub plug to mate with the one on the TNC-Black, or, if you'd prefer, you can use the four holes to the left of U3 to hard wire a radio connection. If you use the 9 pin D-Sub connection, it should be wired as follows:

Pin 1: TX Audio

Pin 3: Push to Talk (PTT)

Pin 5: Receive Audio

Pin 6: Ground

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-Black.

- 1.** Key the radio connected to the TNC-Black 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 Beaglebone Black 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.
- 5.** When you reach this point, back it off just a little and you should have it about right.

Configuration Tips for the Beaglebone Black

A cursory Google search will reveal that a lot of people using the Beaglebone Black are having problems getting the HDMI video to work. I've found the only way to be certain that it will work is to use equipment that has been previously tested and found to work. For example, I have been totally unsuccessful with an HDMI monitor that I have routinely used with my Raspberry Pi. In looking at the specs, I found

that there was only one video mode that was supported by the monitor that was on the list of those that worked with the BBB and I've had no luck getting it to work. However, this monitor also has a VGA input and I have had success using the Cable Matters Active HDMI to VGA adapter that was recommended in the BBB accessories page. Here's a link to this adapter:

http://www.amazon.com/Cable-Matters-Active-Female-Adapter/dp/B00879EZJI/ref=sr_1_2?ie=UTF8&qid=1381610066&sr=8-2&keywords=micro-hdmi+to+vga

and here's a link to the accessories page:

http://elinux.org/Beagleboard:BeagleBone_Black_Accessories#Monitors_and_Resolutions

This page also has a list of monitors and televisions that have been tested and are known to work.

I've also been successful using putty to run an ssh session on my PC and telnet into the BBB. One word of warning here, though. I initially tried this by plugging my Edimax USB Wifi adapter into the BBB's only USB port. It worked, but the performance was dreadfully slow. Googling this problem I found that others had found this to be the case as well, but it could be solved by moving the Wifi adapter to a USB hub and connecting the hub to the BBB usb port. This worked for me as well.

At the time I purchased my BBB, it shipped with an angstrom distribution of Linux on it. Since I'd been working quite a bit with a Raspberry Pi, I thought I would prefer to have Debian Wheezy installed. It was not difficult to do. As I understand it, as of May, 2014, all BBB's are now being shipped with Debian.

At the time that I installed Debian, Xastir was not available in the apt-get repository that came with my Debian image. If this has changed, or if someone would like to write up instructions on installing Xastir, I would love to include it in this manual (with full credits, of course). The same goes for YACC.

Using linbpq and minicom with the TNC-Black

A cursory Google search will reveal that a lot of people using the Beaglebone Black. For keyboard to keyboard QSO's, I have been installing G8BPQ's linbpq software and then running minicom to access it and connect to other stations. To do this, first download the necessary files:

www.tnc-x.com/linbpq.zip

Create a directory for the files (for example, we'll use linbpq in the examples that follow) and then unzip the file into that directory and you will find a configuration file called bpq32.cfg. This is a text file that has the following contents:

```
;      Basic LinBPQ configuration for a KISS Terminal Program

SIMPLE          ; This set a whole load of parameters to reasonable defaults

LOCATOR=NONE

NODECALL=MYCALL-1

TNCPORT
  COMPORT=/home/pi/linbpq/com10
  TYPE=TNC2
  APPLNUM=32
  APPLFLAGS=6
ENDPORT

TNCPORT
  COMPORT=/home/pi/linbpq/com11
  TYPE=TNC2
  APPLNUM=32
  APPLFLAGS=6
ENDPORT

PORT
  ID=TNC-BLACK KISS Port
  TYPE=ASYNC
  PROTOCOL=KISS
  COMPORT=/dev/ttyO0
  SPEED=19200
  CHANNEL=A
  MAXFRAME=2
  FRACK=7000
  RESPTIME=1000
  RETRIES=10
  PACLEN=128
  TXDELAY=500
  SLOTTIME=100
  PERSIST=64
```

KISSOPTIONS=PITNC,NOPARAMS

ENDPORT

APPLICATION 32,TALK,,MYCALL

The items in red are the items that you will have to edit. There are two callsigns here, one for the node itself (MYCALL-1) and another for you to use in contacting other stations through the node (MYCALL). The red text on the lines in the TNCPORT sections contain the full path to the directory that holds the unzipped linbpq files. The text in red in the PORT section contains the serial port that your BBB is connected to.

`/dev/ttyO1`

would refer to UART1. The character before the 1 is a capital O, not a zero.

To connect to another station, you will need to have two terminal windows open, one for running the linbpq program and one for running minicom. If you are doing this locally, you can simply open two terminal sessions in the GUI. If you are use ssh to access the BBB remotely, start two separate ssh sessions. In one of these windows run linbpq:

`./linbpq`

And in the text that follows you should see:

```
slave device: /dev/pts/7. symlink to /root/linbpq/com10 created
slave device: /dev/pts/8. symlink to /root/linbpq/com11 created
TNC Emulator Init Complete
```

The directory specified will be different depending on what user you are using to run this.

Two virtual COM ports have been created, and attached to the TNC2 emulator within linbpq. You only really need one, but with two you can be using one terminal, and still be able to accept calls from others on the second. I've used minicom as a terminal, but other similar terminals should work.

You need to leave linbpq running and in the other terminal run minicom:

`./minicom -D com10 -S minicombpq`

You can now enter TNC2 commands. You can turn monitoring on with MON ON, and you should then see any packet activity on the channel. You can connect to other stations, using C Callsign. One difference from a real TNC2 is that you are connecting through a node, and if a connect fails, you remain connected to the node. You need to enter CTRL/C D to get back to the tnc prompt.

So long as you have CONOK set to ON, people can connect to you, but only if the session is idle. If you want to be able to receive calls while connected, you can open another window and run another copy of minicom. Just replace com10 in the command line with com11. If someone connects to you, LinBPQ will send a bell character to the terminal, but there seems to be a problem with the console driver on the PI, and no sound is heard. I have patched minicom to play a sound file called Ring.wav when a bell is received. The patched version is called piminicom. I've included a sample Ring.wav, but if you don't like it you can use another.

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

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

The TNC-Black 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. The 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-Black. 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**. For additional information on this, contact the author, John Wiseman, directly.

Please Help Improve this Manual

Because more the TNC-Pi for the Raspberry Pi has been available for over a year and TNC-Black for the BBB is being introduced in June of 2014, for some time you will probably find more step by step help with the Pi than with the BBB. As additional application documentation becomes available, I will include it here. If you run across additional useful documentation, please bring it to my attention at: john@hansen.net so I can include it here.