

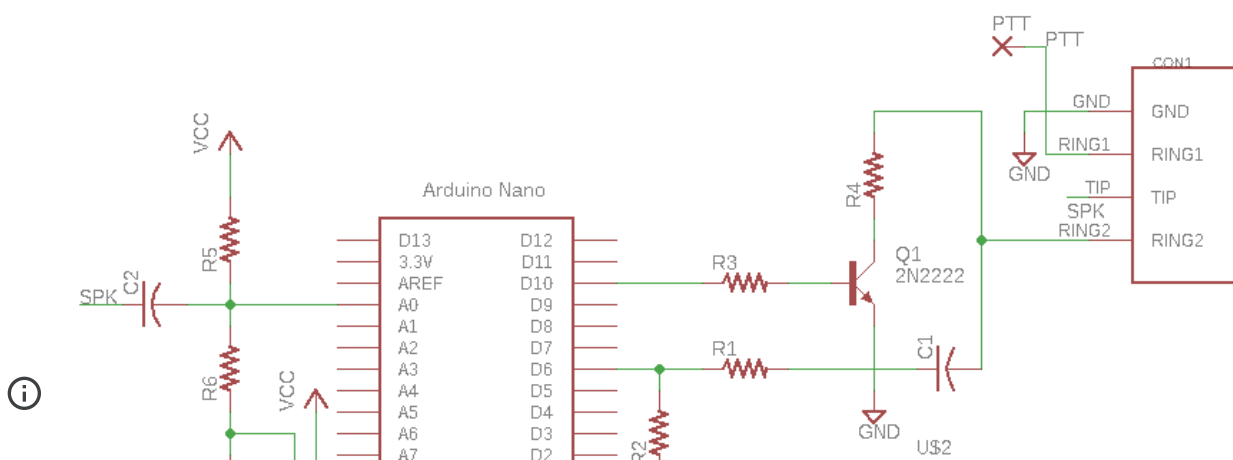
## APRS Tracker / TNC

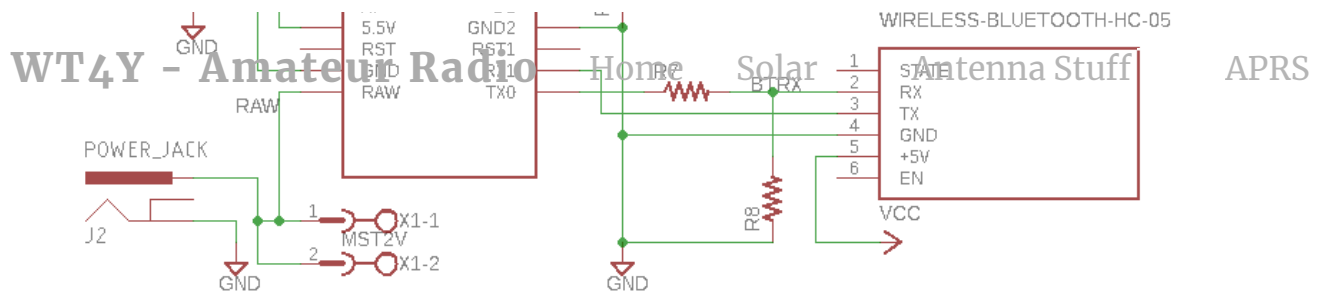
21 April 2018

Here is a very simple Arduino based APRS Tracker created with an Arduino Nano, an HC-05 Bluetooth Module, 8 resistors, 2 capacitors and a general purpose NPN transistor. Used in conjunction with a Smartphone running APRSdroid, it makes for a compact tracking package. The Arduino Tracker is based on an earlier Mobilinkd Tracker design you can find at <http://www.mobilinkd.com/2014/09/11/arduino-kiss-tnc/>. I added the HC-05 Bluetooth Module so a Smartphone can control the tracker without a cable attached to the phone. The only cable is between the tracker and the radio.

If you remove the Bluetooth module and connect the Nano directly to a computer using the USB port, no power supply is required. To a computer, the tracker will appear as a KISS modem running a baud rate of 38400,8,N,1. APRX and Xastir are both compatible with this modem.

Here's the schematic:





R1-100k, R2-10k, R3-1k, R4-2.2K\*, R5 and R6-10K, R7-1k, R8-2.2K

C1-100nf, C2-10nf

Q1-NPN Transistor such as 2N2222, 2N3904 etc.

Fully assembled, a straight 3.5mm TRRS cable will plug directly into a Yaesu FT-60R HT and probably several others brands.

\*Note: For a separate PTT line, leave out R4 and run a wire from Q1 collector (where R4 normally connects) to a pad labeled PTT (goes to Ring-1 on 3.5mm receptacle).

Another option is to leave out the 3.5mm receptacle and hard wire a cable to the pads.

Connections are:

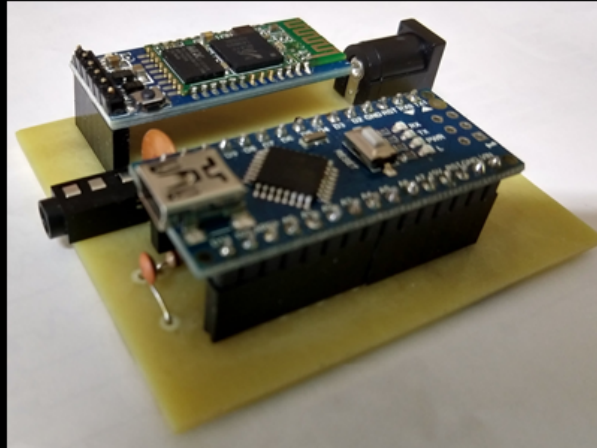
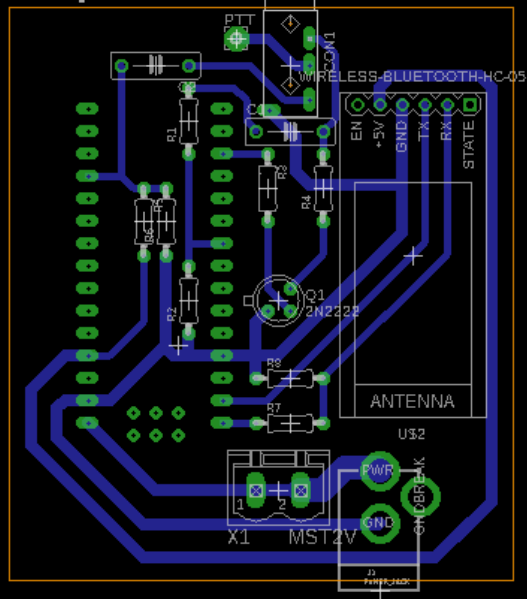
1. Tip = Audio input from the radio's speaker
2. Ring1 = PTT if you opt for the separate PTT line. (otherwise no connection)
3. Ring2 = Microphone input to radio + PTT (or just microphone if using separate PTT)
4. Sleeve = Ground

I created a single layer PC board using Eagle CAD:



# WT4Y - Amateur Radio

Home · Solar · Antenna Stuff · APRS



and a photo:





I also created a 3d printed enclosure:







As a power supply, I'm using a battery pack composed of 6 ea. AA NiMH batteries, but a single 9v battery will also work. You can also power the tracker with a USB power pack through the Nano USB port.

To flash the firmware into the Nano, you will first need to download the compiled HEX file from Github.

The file is at <https://raw.githubusercontent.com/mobilinkd/tnc1/arduino/images/mobilinkd-473-arduino.hex>

For instructions on flashing the firmware, check out <http://www.wt4y.com/arduino-bootloader>. This is by no means the only method and you may have your own favorite programmer so have at it.

Your Nano should now be an APRS KISS TNC.

The Nano TNC communicates with the Bluetooth at a baud rate of 38400,8,N,1 but the default communications baud rate of the HC-05 module is 9600. To change this and optionally change the Bluetooth's Name and Password, see <http://www.martyncurrey.com/arduino-with-hc-05-bluetooth-module-at-mode/> for a tutorial on changing these parameters. The AT command to change the baud rate is  
AT+UART=3800,1,0

Note: AT+UART=<Param>,<Param2>,<Param3> (Param1: Baud, Param2: Stop bit, Param3: Parity) see

[https://www.itead.cc/wiki/Serial\\_Port\\_Bluetooth\\_Module\\_\(Master/Slave\):\\_HC-05](https://www.itead.cc/wiki/Serial_Port_Bluetooth_Module_(Master/Slave):_HC-05)

The last component is a Smartphone running APRSdroid. There are many tutorials on the web for configuring APRSdroid to interface a KISS TNC via Bluetooth.

Here are some specialized parts you will need:



5.5mmx2.1mm 3Pins PCB Mounting Female DC Power Jack -

<https://www.amazon.com/gp/product/B00MJVIFS2/>

Black 4 Pin 3.5mm Stereo Jack Socket PCB Mount Connector -

<https://www.amazon.com/gp/product/B00CQMGBJE/>

2.54mm Breakaway PCB Board 40Pin Male and Female Header Connector for Arduino Shield - <https://www.amazon.com/dp/B01MQ48T2V/>

To keep the PCB as small as possible and keep it single sided, I raised up the Arduino Nano on headers so I could place some components under the Arduino. That also made the Arduino removable for easy replacement or re-purposing.

And finally, you can download the Eagle schematic and board and STL files to 3d print the enclosure. - [APRS-Tracker.zip](#)