

Simple Circuits Inc.

# SIMPLE Raspberry Pi VHF TRANSCEIVER & TNC

2 Meter Transceiver & TNC

© Simple Circuits Inc. 2015-2018  
4/1/2018

---

---



## Simple Raspberry Pi VHF Transceiver and TNC

### Introduction:

This document describes the installation, how to operate, and theory of operation of the Simple Circuits Inc. Simple Raspberry Pi VHF Transceiver & TNC operating in the 144 to 148 MHz range. The transceiver architecture is based on software defined radio techniques and incorporates a Texas Instruments CC1120 RF transceiver for the RF front end, and an ARM Cortex M4F microcomputer for baseband processing.

The CC1120 device is a fully integrated single-chip radio transceiver for narrowband radio applications. It is normally used to send smaller amounts of data for short range links, such as wireless security alarms, medical monitoring, wireless metering, and home automation. It is not normally considered for analog FM audio use. In fact, it is a hidden feature of the part that allows the use for demodulating and modulating classic FM audio.

Besides the CC1120 RF device, an ARM Cortex M4F processor configures the radio and processes the baseband analog signals. The microcomputer is running at 168 MHz and has plenty of processing power to demodulate 1200 bps packet radio and to act as a TNC (terminal node controller), passing data packets to the Raspberry Pi. Adding this unit to the Pi turns the system into an iGATE with only power, antenna, and an internet connection needed.

This board is designed to be an experimentation platform. The capabilities can be expanded by adding appropriate firmware to the microcomputer to enable other forms of demodulation and modulation.



Figure 1: The complete transceiver and TNC fits on a 3.35 by 2.2 inch printed circuit board.

## **Specifications:**

Frequency range:	144 to 148 MHz
Adjacent channel selectivity:	64 dB at 12.5 KHz offset
Blocking performance:	91 dB at 10 MHz
MDS Sensitivity:	better than -120 dBm, -125 dBm typically
Maximum transmit power:	+14 dBm (25 mW) typically
Modulation formats supported:	2-FSK, 2-GFSK, 4-FKS, 4-GFSK, MSK, and OOK
Power requirements:	100mA typical @ +5 volts DC, no speaker attached 250mA typical @ +5 volts DC, 8 ohm speaker attached

## **Current Firmware Status (version 01):**

Receive Only: Although the hardware has the capability to transmit a power levels up to approximately +14 dBm (25 mW) there is no transmit feature in the current version of firmware.

Although the receiver synthesizer can be programmed to tune from 136 MHz to 1 GHz, there are some frequencies that the receiver performs poorly. The firmware does tune the entire 144 to 148 MHz (2 meter) band, the 222 to 225 MHz (1.25 meter) band, the 420 to 450 MHz (70 cm) band, and the 902 to 928 MHz (33 cm) band properly.

Packet Radio: Only 1200 bit per second packet radio demodulation is currently performed in the software. No modulation is currently performed.

## **Installation:**

This board can be attached to the Raspberry Pi Model B, Model B+, or the Pi 2 B. It would be wise to have the Pi setup with Raspbian (or your preferred operating system) before connecting the radio board.

Before installation, make sure that the Pi is powered down.

Plug the radio into the Pi, making sure of the pin alignment as to not bend any pins on the Pi. The corner pins of the radio board should be plugged into the corner pins of the Pi. On

the Model B+ and Pi 2 versions, there are unused connector pins towards the USB connector end of the radio.

Attach a 2 meter antenna to the SMA connector on the radio. Power is supplied to the radio board via the Pi connections. When the Pi is powered, the radio is powered.

A speaker (8 or 16 ohms) can be soldered to the large pads labelled TP2 and TP3. If using an external speaker, note that the +5 volt current requirement is on the order of 250 mA for the board. Please make sure that the Pi power supply has the extra capacity to support the radio. As an alternative audio output, a PC-style amplifier can be plugged into the 3.5 mm connector J4. This is a 600 ohm output and will not drive a speaker directly.

The two jumpers J3 (USB PWR) and J7 (BOOT) should NOT be installed for normal operations.

### **Operating the Transceiver as an iGATE:**

The radio powers up to 144.390 MHz which is the US frequency for APRS.

There are multiple programs that can be installed on the Raspberry Pi to run the system as an APRS iGATE. Xastir is widely used and there are some valuable tutorials on its installation, such as AmateurLogic episode #72. The difference is that this radio firmware version is receive only and Xastir should be configured as a receive-only iGate, not a digipeater.

### **Configuring the Raspberry Pi serial port for use with the Simple RPi Radio**

The radio uses the Pi “ttyAMA0” serial port. Before any of the common APRS programs can be run, the Pi does have to be configured to disconnect the console from the serial port.

The following instructions were based on a downloaded image of the “Jessie” version of the Raspbian operating system.

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
```

Changes to the Raspberry Pi serial port configuration:

Edit the `/boot/cmdline.txt` file in the boot directory and make the following change:

Remove the following (if it exists): `console=serial0,115200`

Note: `cmdline.txt` is a single line of text.

Edit the `/boot/config.txt` file to make the following change:

Add the following: `enable_uart=1`

In addition, if you are using a Raspberry Pi 3 do the following:

Add the following lines to `/boot/config.txt`:

`dtoverlay=pi3-miniuart-bt`

`core_freq=250`

Add the following lines to `/lib/systemd/system/hciattach.service` (Note: this file does not exist in the most recent version of the OS. In that case, create the file in the directory listed above and copy these lines into it):

[Unit]

`ConditionPathIsDirectory=/proc/device-tree/soc/gpio@7e200000/bt_pins`

`Before=bluetooth.service`

`After=dev-ttyS0.device`

[Service]

`Type=forking`

`ExecStart=/usr/bin/hciattach /dev/ttyS0 bcm43xx 921600 noflow -`

[Install]

`WantedBy=multi-user.target`

You may have read some material on line saying the serial port will show up as `ttyS0` on the Pi 3. The above changes will cause it to show up as port `ttyAMA0`, just as it did with the Pi 2 and earlier. Bluetooth will be moved to `ttyS0`.

The following applies to all versions of the Pi:

After making these changes you will have to reboot the Pi so they can take effect. Unless you like to do command line editing, the easiest way to edit these files will be to start the raspberry Pi GUI interface. Then run LXTerminal and type the following:

```
sudo leafpad
```

So, for example to edit the `cmdline.txt` file type:

```
sudo leafpad /boot/cmdline.txt
```

This will run a very nice GUI editor with superuser privileges to allow you to edit system files.

Remember to use your preferred text editor in **sudo** mode and don't forget to reboot the Pi after changing the files.

### **Operating the Transceiver as a general purpose VHF radio:**

A Python script is available to operate the radio from the Pi user interface. It allows any frequency between 136 MHz and 999 MHz to be tuned and adds volume control to the audio outputs. It is easy to modify this code without understanding Python. You will see a table of numbers representing scan frequencies. They can be changed to your preferred scan frequencies. The startup frequency can also be changed.

### **Theory of Operation:**

The RF device receiver is an IQ architecture. It contains quadrature mixers, followed by amplifiers and low pass filters as shown in the block diagram of figure 2. The quadrature mixers driven from a local oscillator with 90 degree phase differences between the two. The local oscillator is a fully integrated fractional-N frequency synthesizer that can tune the 136 MHz to 999 MHz range. Only the radio's front-end filter circuit limits the frequency range.

The CC1120 device digitizes the output of the mixers. From this point, all processing is in the digital domain. The channel filters limit the frequency of the baseband signals and frequency offset is computed. It is this data that is read by the external ARM processor. For classic analog FM demodulation, the other FSK and OOK (on off keying) demodulator circuits in the CC1120 are not needed. The CC1120 user's guide, Texas Instruments publication SWRU295E, has a full explanation of the component's operational capabilities.

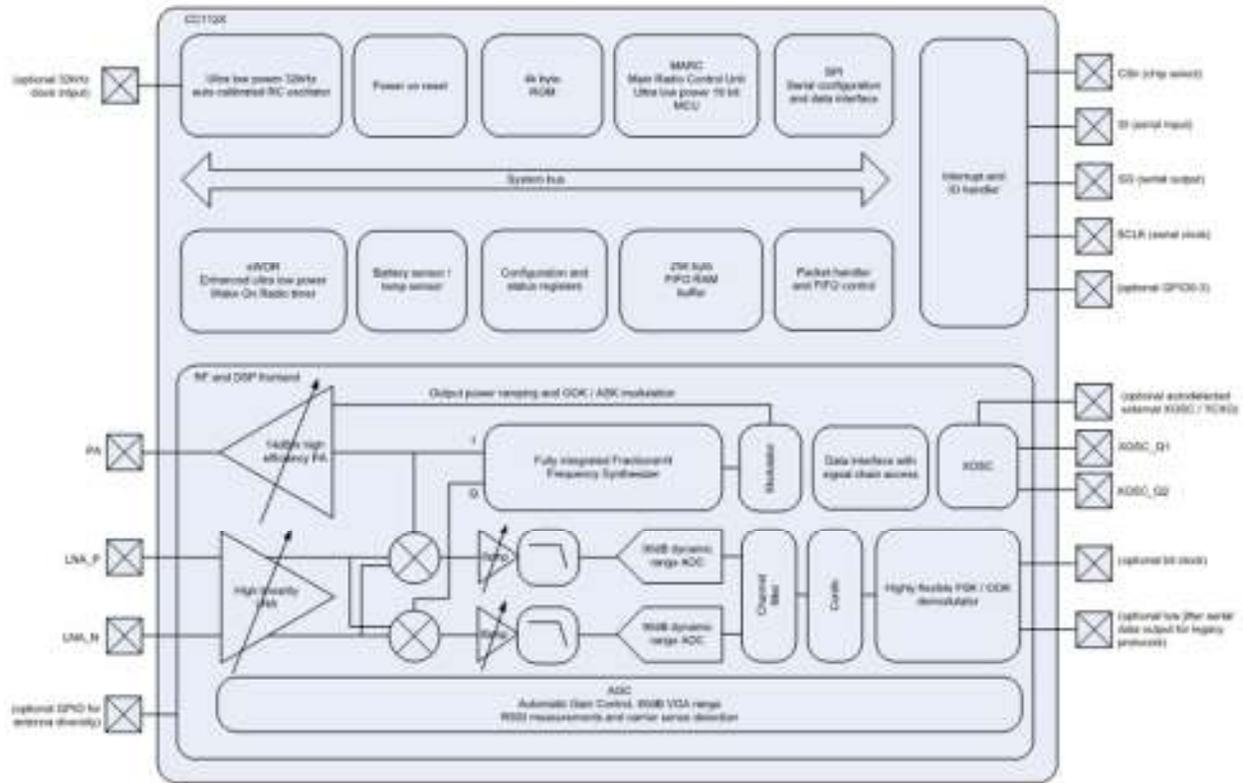


Figure 2. CC1120 RF Transceiver block diagram.

For 1200 baud packet data demodulation and the filtering of the audio, all processing is performed in the ARM processor.

Data is read from the CC1120 over an SPI bus into the ARM processor at a sample rate of 38,400 samples per second. The data is fed into a CIC filter that decimates the data by a factor of 4 resulting in a data rate of 9600 samples per second. From this point, the data is fed into two separate paths: the 1200 baud demodulator and a digital 64 tap FIR band-pass filter. The data fed into the FIR filter is the signal that goes to the speaker outputs. The band-pass filter removes any CTCSS tones that are present on the received signal and it removes audio frequencies above 2500 Hz. The filter magnitude plot and specifications are shown in figure 3. The output of the FIR band-pass filter is fed into the ARM's 12 bit digital to analog converter (DAC) that converts the data stream back into an analog signal. This signal drives the input of a class D audio amplifier, and also passed to the output connector. The signals sent to the 1200 baud demodulator need the full audio bandwidth and are not filtered, except for the low-pass effects of the CIC filter.

The TNC operation is performed on the output of the 1200 baud demodulator. Packet data follows the AX-25 protocol. The processor converts the data from AX-25 to the KISS protocol and passes this data over an asynchronous UART port to the Pi.

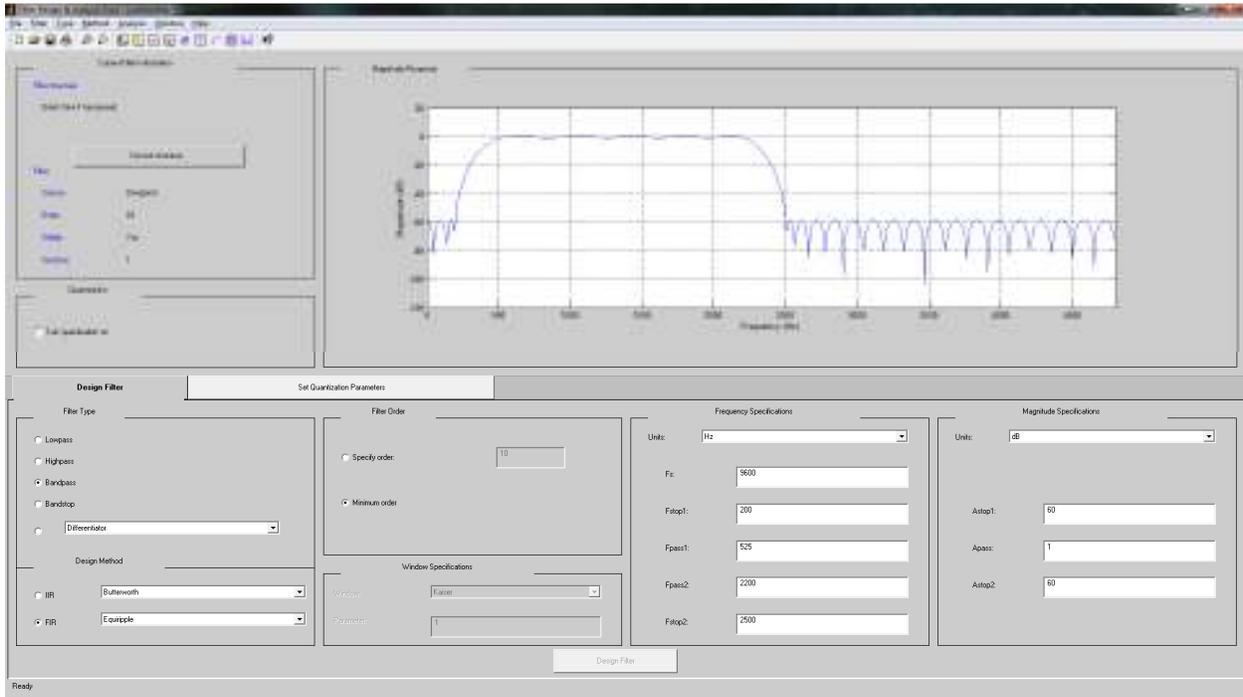


Figure 3: Audio digital band-pass filter response.

### Firmware Upgrades:

The ARM processor's firmware can be modified and/or upgraded without the need for any special hardware development tools. The radio (not the Pi) USB port can be connected to a PC and by using ST Microelectronics programming software, the processor's flash memory can be written with new contents. The jumper J7 is installed to enter the programming mode. This jumper should never be installed during normal operation.

## Wideband Receiver Modifications:

The front end RF filtering is designed to attenuate frequencies above 148 MHz. Figure 4 shows the frequency response for the filter.

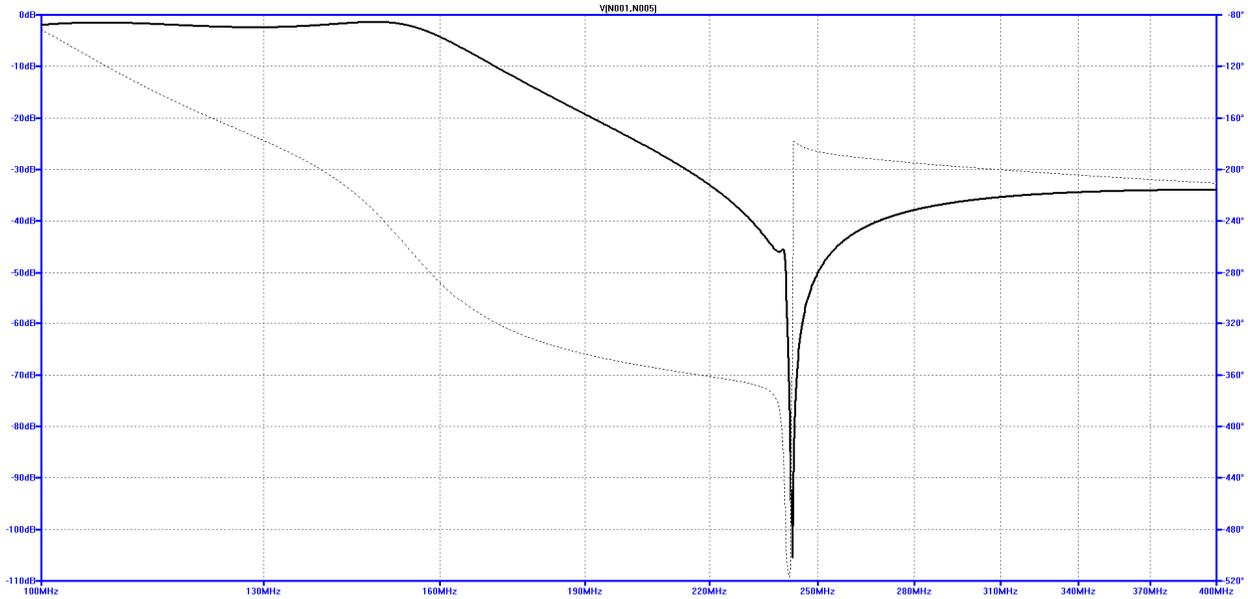


Figure 4: 144 MHz Input RF Filter Response.

It is possible to remove 4 components from the front end antenna filtering circuit and convert the receiver from a 2 meter receiver into a wide band receiver with only minimal attenuation between 136 MHz and 1 GHz. See figure 5 below. As mentioned before, the receiver has poor performance at some frequencies due to internal spurs in the CC1120 receiver component, but works on the amateur bands of 144 MHz, 220 MHz, 440 MHz and 902 MHz.

If this radio is used for 2 meter APRS, it would not be wise to make this modification since any strong signals above the 2 meter band may over load the receiver input, negatively affecting the APRS performance.

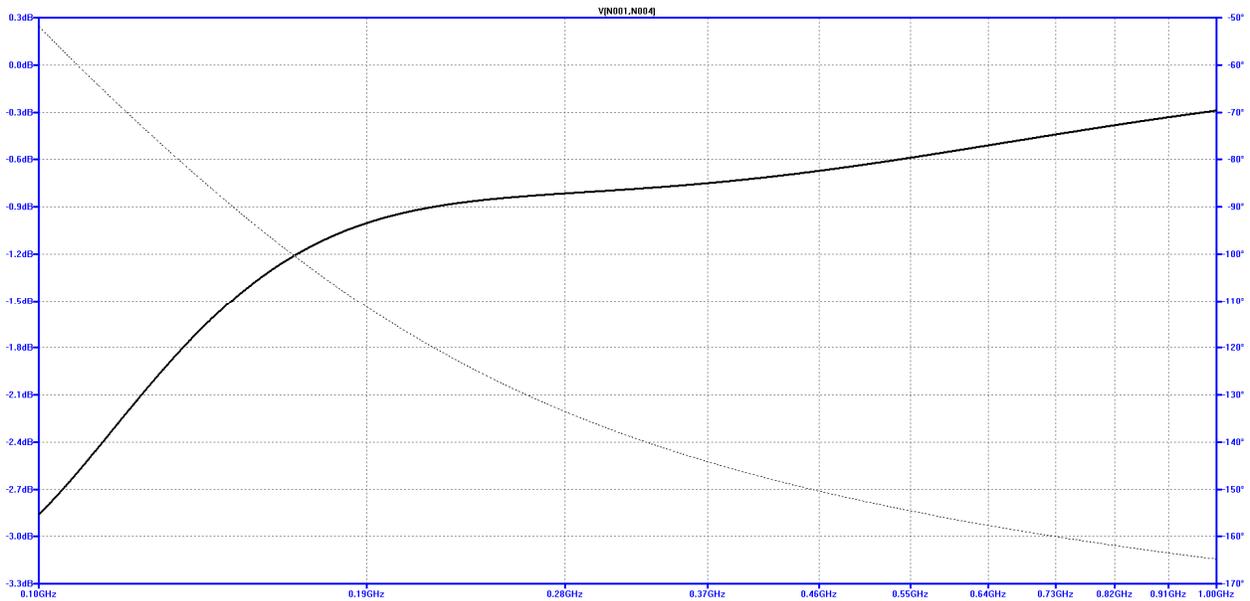


Figure 5: Wideband version of the Input RF filter response after wide band component modifications.

The modifications are:

1. Remove L2, L5, L8, and C32. These parts are shown in red in figure 6.
2. Add a jump wire between the center pin of J6 and the right end of L5 as viewed as in figure 6.

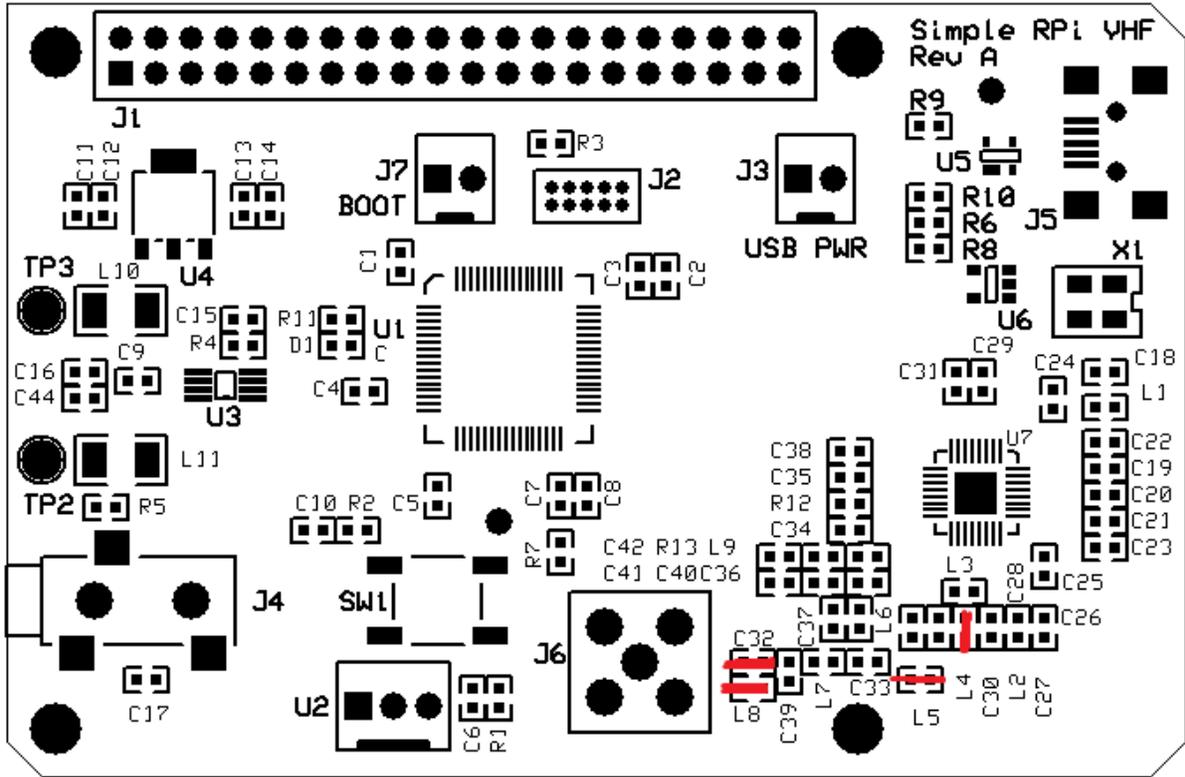
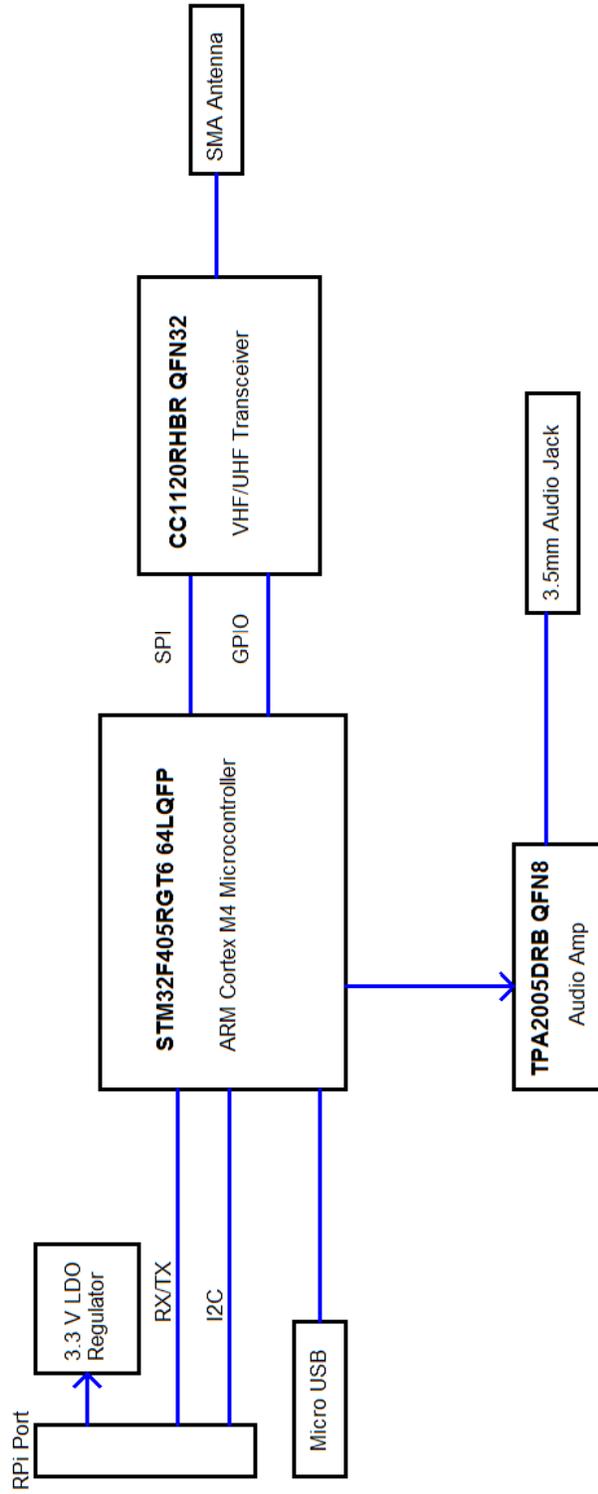


Figure 6: Board silkscreen showing parts to remove in red.

**SIMPLE RPi VHF/UHF Transceiver Block Diagram**



© Simple Circuits Inc. 2015

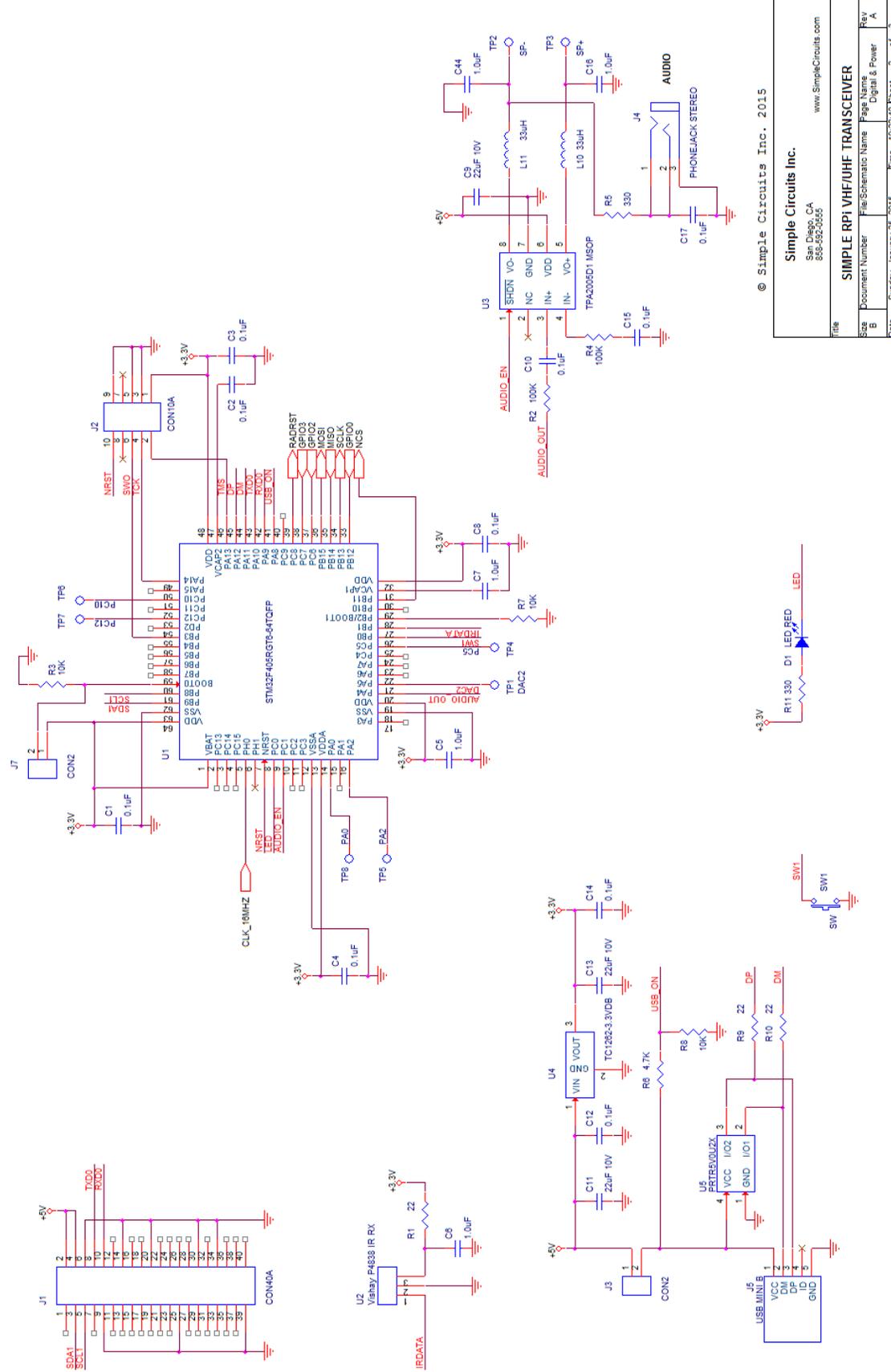
**Simple Circuits Inc.**

San Diego, CA  
858-592-0855

www.SimpleCircuits.com

**SIMPLE RPi VHF/UHF TRANSCEIVER**

File	File Schematic Name	Page Name	Rev
Size	Block Diagram	Block Diagram	A
Document Number			
Date	Thursday, January 22, 2015	Time: 18:05:54	Sheet 1 of 3



© Simple Circuits Inc. 2015

Simple Circuits Inc.  
 San Diego, CA  
 858-592-2045

File  
 Name: Simple Circuits Inc. 2015

Doc Number  
 B

File Schematic Name  
 Simple RPi VHF/UHF Transceiver

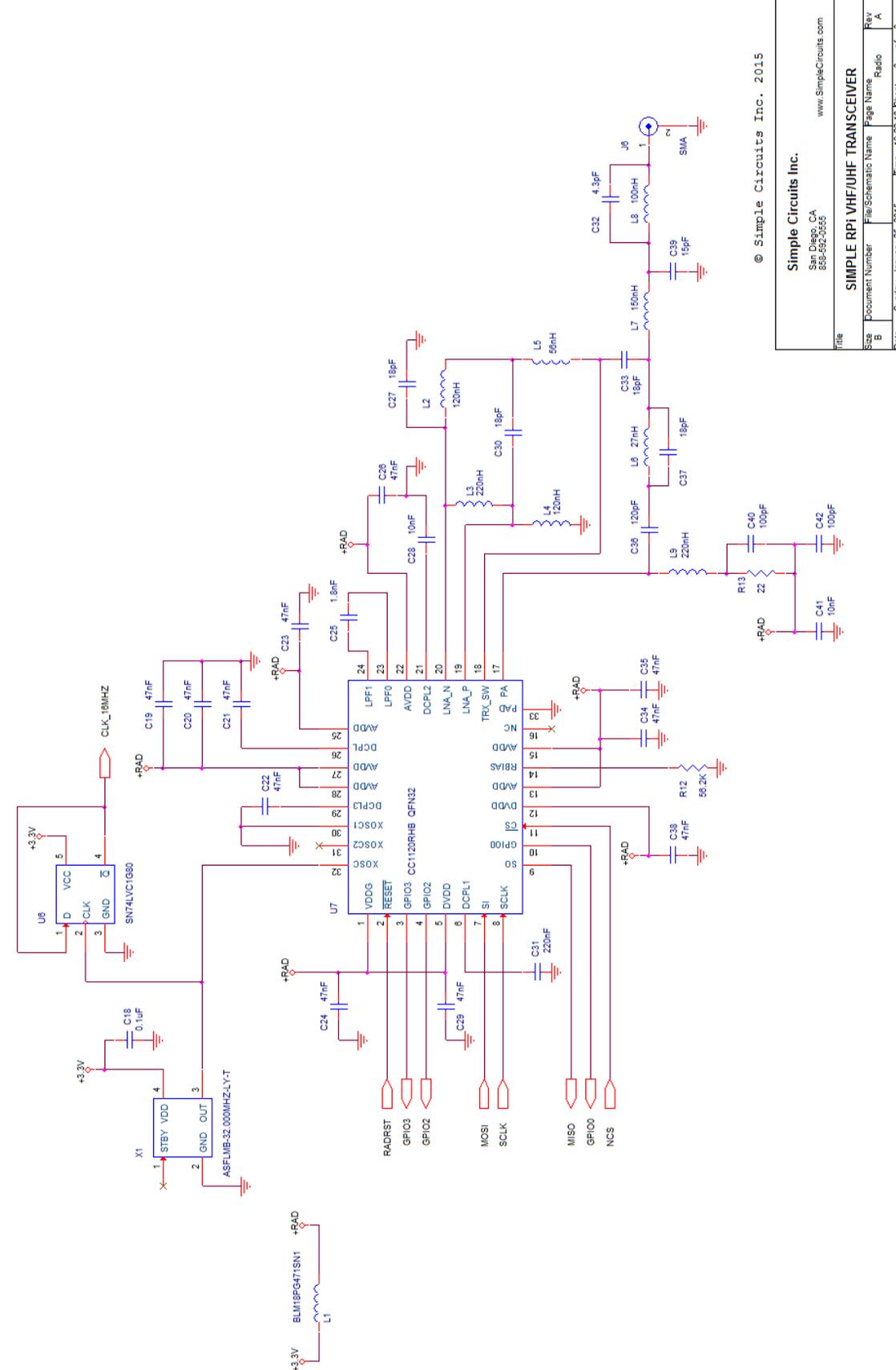
Page Name  
 Digital & Power

Rev  
 A

Date  
 Sunday, January 25, 2015

Time  
 19:22:40

Sheet  
 2 of 3



© Simple Circuits Inc. 2015

Simple Circuits Inc.  
San Diego, CA  
858-592-0950  
www.SimpleCircuits.com

File	SIMPLE RPi VHF/UHF TRANSCEIVER		
Size	Document Number	File Schematic Name	Page Name
B		Radio	Rev A
Date	Sunday, January 25, 2015	Time	13:22:13 Sheet 3 of 3