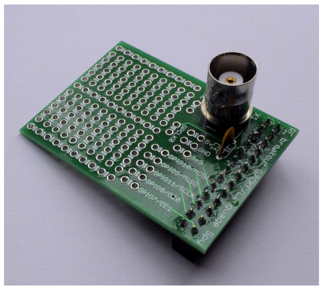
RF breakout kit instructions

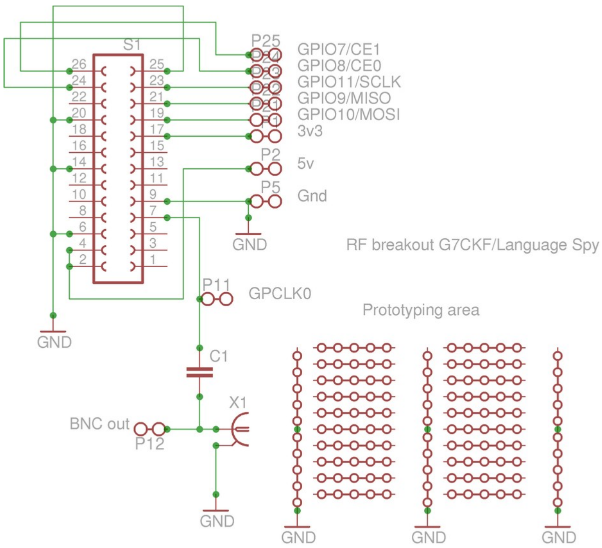
Introduction



The Raspberry Pi has a built-in clock generator, which can be accessed through pin 7 of the expansion connector. This clock generator is a powerful frequency synthesiser which can generate frequencies up to 250MHz at the Pi's 3.3v logic levels. This is enough to provide a useful RF signal source for experimentation, or given suitable filtering and antennas to allow the Pi to be used as a low-power amateur radio transmitter.

This kit provides a breakout board for the clock generator and some of the the SPI/GPIO lines, with a small prototyping space and a BNC socket for RF output. A DC blocking capacitor is supplied allowing the construction of a basic square wave source, while the prototyping area may be used for the construction of low pass filters, transmitter control logic, or other circuitry.

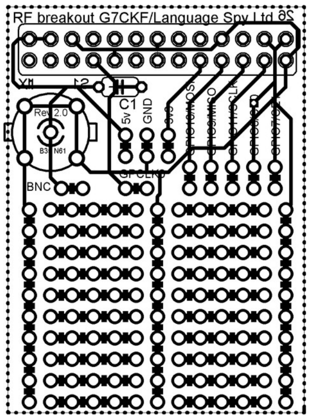
The circuit



The circuit diagram is shown to the right. The connector SV1 plugs into to the Pi's expansion port, the GPIOs, clock line and 3v3 rail are brought out to solderable connection points, and there is provision for a DC blocking capacitor between GPCLCK0 and the BNC socket. A small area of 0.1” pitch prototyping space with three earthed buses for easy grounding of RF components is also provided for the assembly of your own circuits.

The DC blocking capacitor C1 serves to protect the Pi from any DC voltages or short circuits present on whatever you may connect to the BNC socket. It is essential if you are using the circuit in its simplest form as an RF source but should be omitted from its marked position on the board if you are using the GPCLK0 line in your own circuits. It is **STRONGLY RECOMMENDED** that you use the DC blocking capacitor in any circuits you may build yourself, as the GPCLK0 line comes direct from the Pi's processor and any abuse of it will kill your Pi.

Building your board



This is an RoHS and WEEE compliant product, so to maintain that compliance it should be soldered with a lead-free solder. Any decent quality flux cored lead-free solder designed for electronic use should be suitable for assembling this kit. Solder intended for plumbing may not be suitable.

The board layout(top view) is shown on the right. The Pi expansion connector SV1 is mounted on the bottom of the board, and the BNC socket and capacitor on top. Basic soldering skills are required to assemble this kit. Teaching soldering is beyond the scope of this document, however it is assumed that since this kit is marketed at radio amateurs and electronics hobbyists as a self assembly item you will already have these skills.

It is recommended that you start with the connector SV1. Push the pins through the holes in the bottom of the board , and carefully holding it so that the connector is flush with the board solder the corner pins on the top of the board. You may find a clamp, set of “helping hands”, or even a clothes peg handy at this point to provide a third hand. Once the connector is held in place by the corner pins you can then continue and solder the rest of the pins.
The BNC connector can then be pushed through its holes on the top of the board and its pins soldered on the bottom of the board. The outer pins may need more heat than the inner pin as the metal body of the connector

may need to heat up. Yet again you may need a clamp or similar to help you in this step. Watch out – the BNC connector will get hot.

The next step is optional depending on whether you want to use the board as a simple RF source or not. If you want the GPCLK0 output to go direct to the BNC socket then the DC blocking capacitor C1 should be installed, and its leads clipped close to the solder joints on the bottom of the board. If however you wish to route the GPCLK0 output to the prototyping area for a filter or similar, it is **STRONGLY RECOMMENDED** that you use the capacitor to do this in order to protect the GPCLK0 line from DC voltages or short circuits.

You should now be able to build your own circuits on the prototyping area. When you have done so, it is **very important** that you examine your work carefully, if necessary using a magnifier. Pay close attention to any solder bridges that may have formed between adjacent components, the orientation of any semiconductors, and the orientation of electrolytic capacitors. Remove any surplus solder or solder bridges with desoldering braid.

You should also take a look to ensure that the any protruding pins from your through-hole components do not foul any of the components on the Pi itself. If necessary cut a piece of thin card to act as an insulator between the bottom of the breakout board and the Pi.

These checks are **very important** because your Pi can be damaged by short circuits or other mishaps on your board. **If that happens it is your responsibility as the builder of the board**. It is better to have to rework or desolder something than to damage your Pi with your work.

Square waves and filters: some theory

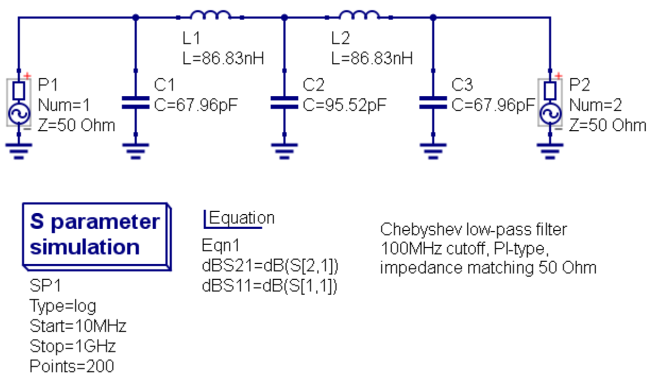
The GPCLK0 output from the Pi is a logic level output. This means it has two states, logic 1(3.3v) or logic 0(0v). The clock signal is thus a continuous oscillation between those states called a square wave after its square appearance on an oscilloscope.

Square waves pose a problem from an RF viewpoint. The majority frequency component of a square wave is their fundamental switching frequency which is usually the frequency you want. Unfortunately they also contain a significant proportion of harmonic frequencies, multiples of the fundamental frequency. (The full mathematical derivation of this can be found at https://en.wikipedia.org/wiki/Square_wave)

If you make a radio transmitter that generates square waves, those harmonics are going to cause interference on multiples of the frequency you want to transmit on. Your neighbour's TV, your local police radios, your local airport's emergency channel, all sorts of other users are going to get very upset and you are likely to receive a visit from the authorities which will not end well for you.

It is thus rather important if you are building a radio transmitter to ensure that it incorporates a filter to remove those unwanted harmonic frequencies and leave only the fundamental frequency you are interested in. The most usual filter employed for this purpose is a low pass filter, so called because it allows frequencies below a particular point to pass through it and attenuates those above that point. The cut-off point is usually designed to be just above the desired frequency and well below the first harmonic frequency.

The intricacies of filter design are beyond this document. Fortunately there is plenty of available information on the subject and a web search for low pass filters and the band you are interested in will usually return pages with other people's tested designs ready for you to build. If you fancy trying your hand at filter design then there are plenty of amateur radio publications from the likes of the RSGB or the ARRL containing the information you will need, or you can use an electronics CAD package like QUCS (http://qucs.sourceforge.net/) which has a set of built-in filter design tools. The circuit diagram shown is of a 100MHz Chebyshev low pass filter designed using QUCS.



Using the Raspberry Pi to generate RF

The Raspbian operating system used by most Raspberry Pi users does not come with any software to generate RF. If you wish to use the board you will usually have to download, install or compile some some software from other sources, or even write some yourself. Fortunately this is usually not a difficult process.

A few pieces of software that can use this board are listed below.

Freq_pi and PiVFO

https://github.com/JennyList/LanguageSpy/tree/master/RaspberryPi/rf/freq_pi
We have produced PiVFO, a simple graphical front-end for Jan Panteltje's command line frequency generator freq_pi. You can find both pieces of software on GitHub at the address above. Follow the instructions on the GitHub page to install, compile and run both packages on your Pi. You should install freq_pi first, then PiVFO. You will need to be able to type some commands on the Pi command line.

WsprryPi

https://github.com/JamesP6000/WsprryPi
This is a transmitter for the amateur radio WSPR protocol (http://wsprnet.org/drupal/), a beacon protocol designed to test and explore atmospheric radio propagation modes. Receiving stations around the world post their spots to the web site, and it is possible for a Raspberry Pi transmitter to be heard all around the globe given the right band and atmospheric conditions. This software requires an amateur radio licence for use.

Rpitx

https://github.com/F5OEO/rpitx
This software from F5OEO is an exciting development for radio on the Raspberry Pi, using software defined radio techniques to generate AM, SSB, and SSTV modes. Instructions can be found on the GitHub page above. This software requires an amateur radio licence for use.

Appendix: Declaration of conformity

Organisation: Language Spy Ltd, Neve's Croft, Godington, Bicester, OX27 9AF, UK
Product description: RF breakout kit the Raspberry Pi, a self-assembly kit of electronic parts.

This product has been designed to comply with the following directives:

Directive 2001/95/EC on general product safety
Directive 2011/65/EU on the restriction of the use of certain hazardous substances in electrical and electronic equipment

As a piece of electronic equipment requiring less than 50V DC it is not applicable under article 1 of directive 2014/35/EU on low voltage.

As a self-assembly kit for use by radio amateurs it is exempt from the following directives under the following clauses:
Annex 1 paragraph 1 of 2014/53/EU on radio & telecommunications equipment.
Article 1 paragraph 2(c) of 2004/108/EC on electromagnetic compatibility.
Article 2 paragraph 2(c) of 2014/30/EU on electromagnetic compatibility (When it becomes applicable).

Date: 2015-11-01

Jenny List, director