

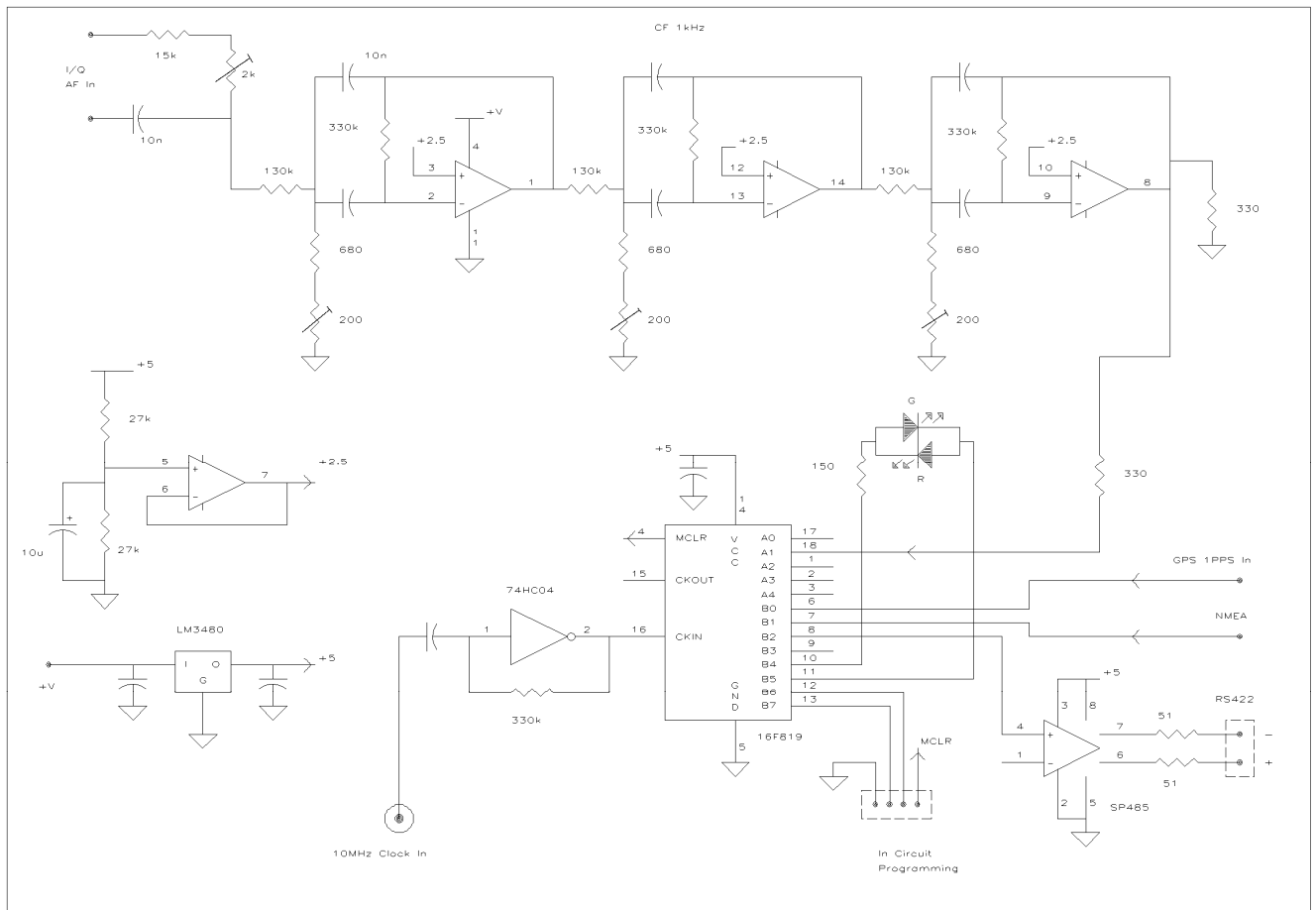
Overview

This digital receiver is designed specifically for ultra high stability reception of very narrow band LF signals. It was intended specifically for use with coherent signalling techniques involving a reference locked to a Rubidium or GPS source.

It is based around a Softrock type front end downconverting an LF input of up to 500kHz. The quadrature output of the receiver is at 1kHz for bandpass sampling.

The 1kHz bandpass sampling idea was first proposed many years ago by G3PLX. Since first trying that, a new version of the PIC code has been written for the more modern 16F819 PIC microcontroller with its 10 bit A/D converter. Time stamping using the one pulse-per-second and NMEA data stream from a GPS receiver has been added to the output data stream to assist in logging within any final application using the digitised data.

Figure 1 shows the circuit diagram of the final unit. The audio input first passes through a three stage opamp bandpass filter tuned to the centre frequency of 1000Hz. This serves to reduce any possible frequency components below 750Hz and above 1250Hz that would alias into the passband. The output from the filter passes to the PIC's analogue input.



Bandpass Sampling

The PIC is clocked with an externally supplied 10MHz master reference which is divided down internally to generate an interrupt at 4kHz. At each of these the input signal centred on 1kHz is sampled to give a 10 bit value. Groups of four successive samples are combined as $I = S1+S2-S3-S4$ and $Q = S1-S2-S3+S4$ to generate two 12 bit numbers from -2048 to +2047 corresponding to the downconverted baseband In-phase and Quadrature (I/Q) values, delivered 1000 times per second. One major advantage of doing the digitisation this way is that the add and subtract process associated with generating each output sample causes any DC offset to be exactly cancelled out

The data to the host PC is sent using stop–start signalling over an RS422 interface using twisted pair interconnection at 115200 baud. RS422 ('RS485' when operated bi-directionally) with its balanced and matched transmission line is more immune to interference pickup, and can transmit over vastly further distances than the traditional RS232 ever could. At 115200 baud signalling, an 8 bit byte with its associated start and extended stop bits takes a little under 100us to send. So there is potentially space to fit up to ten such bytes into each sample. The 12 bit samples of I/Q data are coded as follows:

Interface Protocol

The first byte contains the lowest 7 bits of the I data with the MS Bit set to '0' giving '**0iiiiii**'. The second byte takes the five highest order bits, with its three remaining Most Significant bits set to '100' '**100iiii**'

The third and fourth bytes follow the same pattern with the Q data, but the header is then '101', so it appears as '**0qqqqqq**' followed by '**101qqqqq**'. Formatting the data this way means the host PC knows that any byte with a zero in its MS position must be one of the low order ones, and any with 100 or 101 in the leading bit positions must contain I or Q high order data respectively. If data gets scrambled, or bytes are lost, the receiver can rapidly resynchronise based on the known format.

Time stamping

Timing information is delivered to the PIC via the 1 PPS and NMEA data from a GPS receiver. This data is read once at startup and used to initialise internal timekeeping registers. Subsequently, and only started after it is synchronised to the 1 PPS signal, time keeping is then maintained in software using the 4000Hz interrupt after being further divided down. This makes it absolutely synchronous with the sampled data.

Immediately after the seconds value increments, the hours, minutes and seconds (UTC) as well as the day of the month are sent in four additional bytes on the output stream. The time is encoded using a similar principle to the I and Q data except that only one header, '111' is used for the whole four byte burst which appears in the format. '**00sssss**' '**000mmmm**' '**0000hhhh**' '**111dddd**'. The remaining timestamp information of month and year is sent 32ms after this as '**0yyyyyy**' '**1100nnnn**'. s, h, m, d, n, y are the individual bits making up the value of seconds, minute, hour, day, month and year (two-digits) respectively. So the majority of bursts 1000 times every second carry four bytes of I/Q data, with occasional ones every second carrying 8 or 6 bytes. The headers are separated out in the host to decode the timestamp. Since timing is synchronised to the PPS at

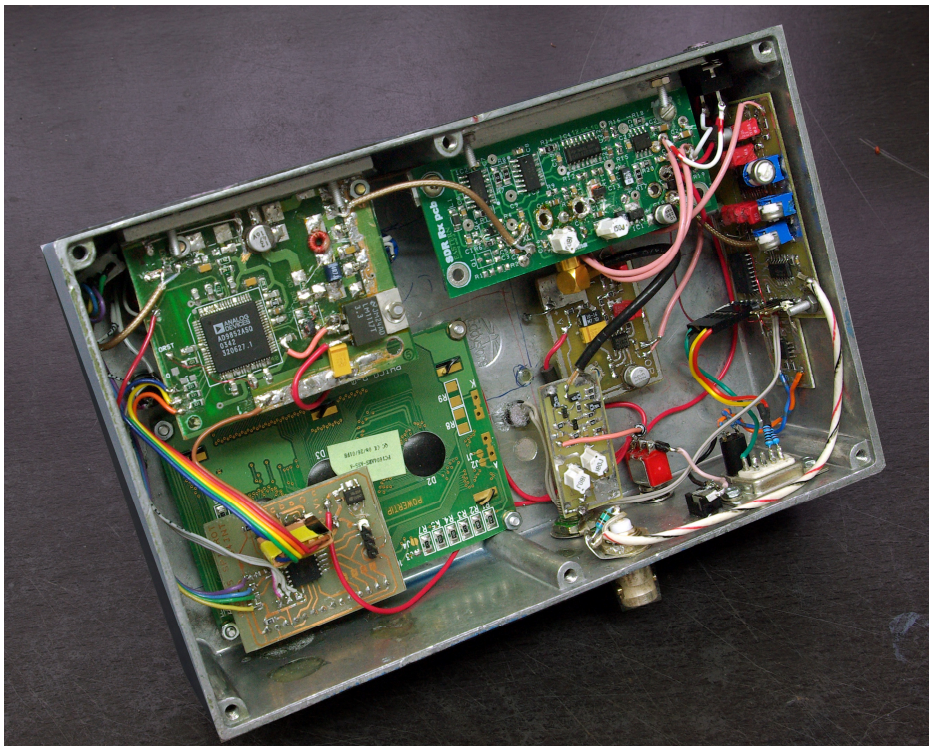
source, timing accuracy is determined by the one millisecond resolution of the sample rate. PIC source code for the Quadrature Digitiser can be found at [1]

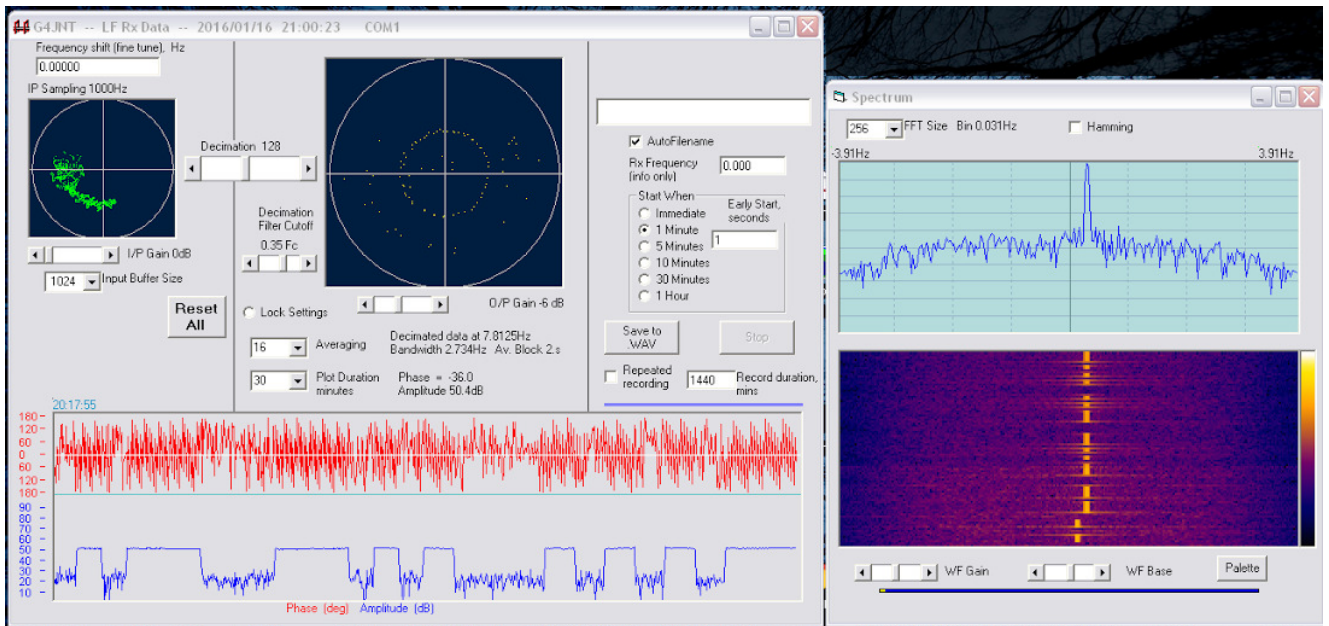
A Suitable Receiver

The input filter provides 50dB of anti-alias filtering at its edges which means for most applications with narrowband LF signals, no additional filtering is necessary. So any fully frequency locked radio (upper sideband) can be used, tuned exactly 1kHz below the wanted signal. Alternatively a Softrock or other QSD type can be used, again with its LO 1kHz low. The 15k Ω + preset resistor and 10nF capacitor shown at the top left of Figure 1 provide a simple phasing network for combining the I/Q audio streams from such a receiver. A simple CR network will only allow a worthwhile level of cancellation of the image 2kHz away over a bandwidth of a few tens of Hz either side of centre but, with the ultra-narrow signals of the type we will be using this receiver for, the simplicity is acceptable. With a carrier tuned to deliver exactly 1kHz IF, over 50dB of sideband rejection was measured using the modified Finningley Dongle receiver module. The finished version can be seen in the photograph.

PC Software

At the time of writing, PC software exists that takes the serial input stream from a COM port and decimates this. The resulting I/Q samples can be stored in a .WAV file in a similar format to that generated by Spectrum Lab, and suitable for reading by the EbNaut receiver software. This is undocumented, but a complete Setup archive written in VB6 can be found at [2]. Hopefully the user screen, shown, should be reasonably self-explanatory.





LFRx2 Software User Screen

References

- [1] www.g4jnt.com/Download/1kHz_IQ_Sampler.zip (note underscores, no spaces in that URL)
- [2] <http://www.g4jnt.com/Download/LFRx2.zip>