FSK Modem using DDS and FIR Filters in an FPGA

Dan Kirkham

Introduction

A frequency-shift keying (FSK) modulator and demodulator is outlined, designed, simulated, verified, and ultimately executed on FPGA hardware. The design utilizes direct digital synthesis (DDS) and uses several different finite impulse response (FIR) filter designs.

For hardware, a Lattice iCE40 UP5K is used. This features 5280 LUTS, 30 block RAMs, and only 8 16x16 multiply-accumulate DSP slices (compared to a Xilinx Zynq RFSoC, which is typically around 5000).

The design is implemented using RustHDL. RustHDL is a HDL environment that leverages Rust's strong static typing system that catches many potential bugs at compile time. RustHDL will generate lower level Verilog code, which can be fed into the desired synthesis tool. RustHDL also ships with many useful cores, such as a rolled FIR implementation that uses a single hardware multiply-accumulate (MAC). The Yosys toolchain this used to synthesize this design infers the DSP slice automatically.

The source code for this project can be found at http://github.com/dankirkham/fsk_modem

Design Overview

A UART is used to receive a binary signal and modulate it with FSK. Since this is just a demonstration of the FPGA signal chain, the generated samples forgo digital-to-analog and analog-to-digital conversion. The samples are looped directly back into the demodulator. The signal is demodulated back to a TTL UART signal and is transmitted back to the original sender. Figure 1 shows a top level diagram of the design.

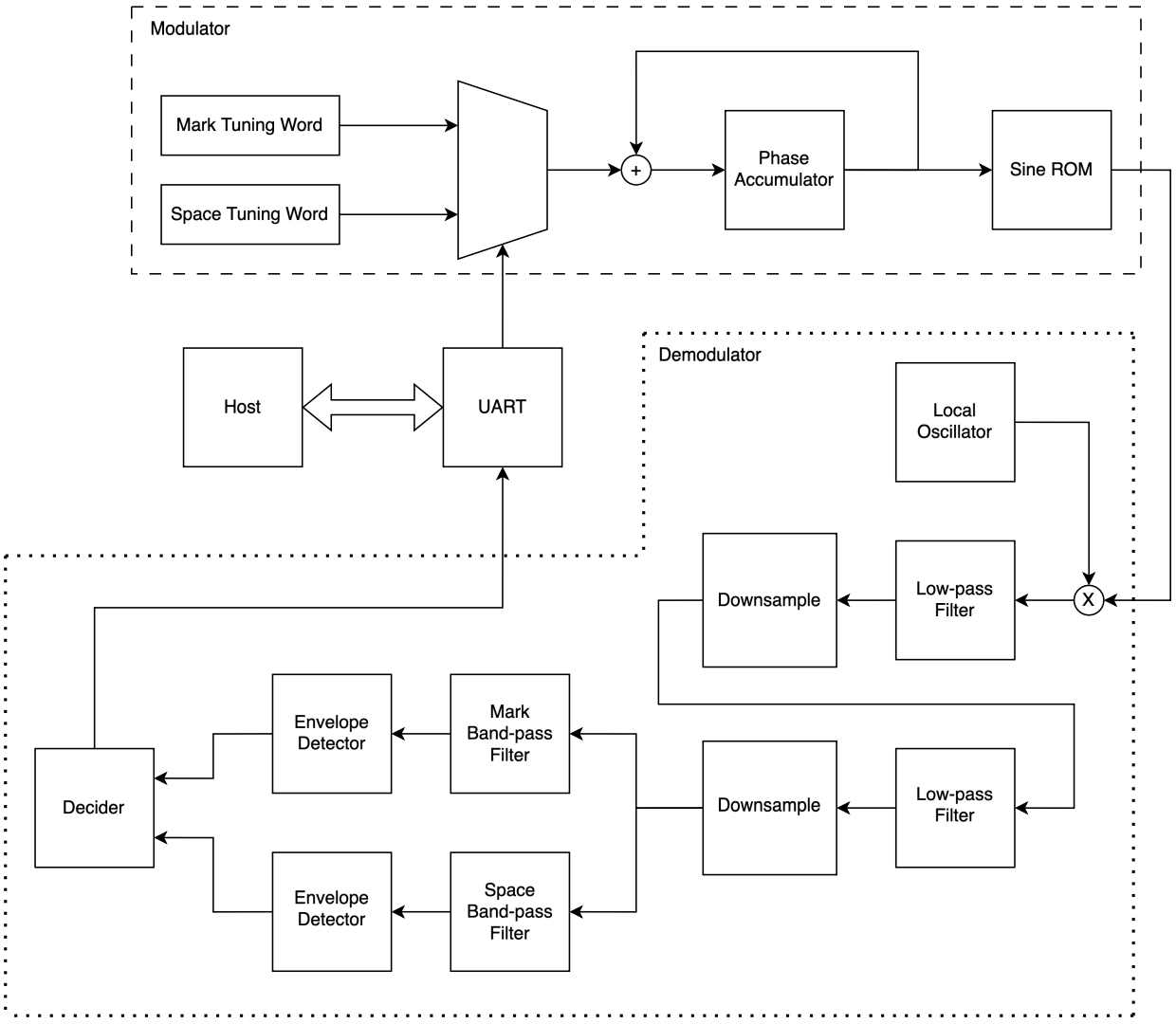

Figure 1: Top level block diagram.

Modulator

The modulation design used here is an example of DDS. A phase accumulator is used to provide a linearly increasing ramp function. This ramp function is used to select a sample from the sine ROM. With each cycle of the reference clock, the phase accumulator is either incremented by the mark tuning word or the space tuning word. Changing between these two values will change the frequency of the waveform, thus providing frequency-shift keying (FSK). Equation 1 shows how the reference clock, tuning word, and phase accumulator bit-width relate to the frequency output of the DDS. Figure 2 shows a simulated waveform of this modulator design.

$$f_o = \frac{M * f_c}{2^N} \tag{1}$$

Tuning Word Selection

A 307.2 kHz and 316.8 kHz are arbitrarily chosen for the space and mark frequency, respectively. Because they are 9600 Hz apart, this can support a baudrate of a UART, 9600 Hz. 307.2 kHz is both an

integer divisor of the DDS reference clock frequency of 2.4 MHz and a multiple of the baudrate. The tuning word, M, can be found with Equation 2, which is a form of Equation 1:

$$M = \frac{f_o * 2^N}{f_c} \tag{2}$$

$$M = \frac{307.2\ \text{kHz} * 2^{24}}{2.4\ \text{MHz}} = 2,147,483 \tag{3}$$

$$M = \frac{316.8\ \text{kHz} * 2^{24}}{2.4\ \text{MHz}} = 2,214,592 \tag{4}$$

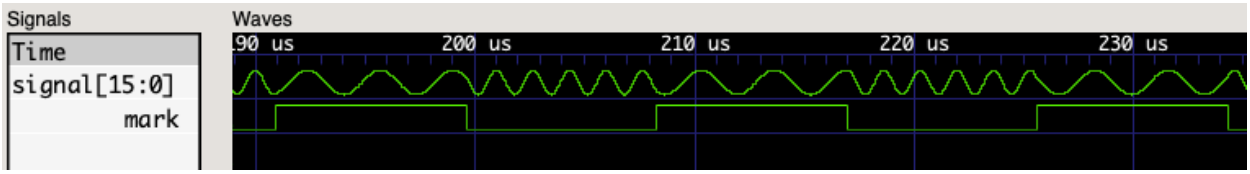

Figure 2: Simulation of modulation circuit, with exaggerated mark-to-space frequency separation.

Demodulator

The demodulation design used here is an example of a filter-type FSK demodulator. The signal is filtered with two separate band-pass or matched filters, one for the space and one for the mark. The resultant signals are envelope filtered and then the magnitude is compared. The symbol with the greater magnitude is output by the decider.

Mixer

To make the sample rate slower and easier to work with, it will be heterodyned to a lower intermediate frequency (IF). This IF must be higher than the symbol rate of 9600 Hz. In this case, 15 kHz will be used. The local oscillator (LO) frequency is selected using Equation 5, which ends up being 292.2 kHz. The mixer is implemented by multiplication. The LO and received waveform are multiplied using a 16x16 fixed-point hardware multipier. After mixing the space and mark frequencies will be 15 kHz and 25 kHz, respectively.

$$f_{\text{LO}} = f_0 - f_{\text{IF}} \tag{5}$$

$$f_{\text{LO}} = 307.2\ \text{kHz} - 15\ \text{kHz} = 292.2\ \text{kHz} \tag{6}$$

To generate the LO, a second DDS module is used, and the tuning word is found using Equation 2.

$$M = \frac{292.2\ \text{kHz} * 2^{24}}{2.4\ \text{MHz}} = 2,042,626 \tag{7}$$

Low-pass Filtering and 4x Downsampling

The signal is low-pass filtered and 4x downsampled *twice*. This yields a final sample rate of 150 kHz, which is more than twice the maximum deviation of the signal. The low-pass filter is required to prevent higher frequencies from aliasing into the decimated signal. The first iteration uses a cutoff frequency of 300 kHz, as shown in Figure 3. The same filter is reused for the second stage, this time having a cutoff frequency of 75 kHz. This is still more than twice the maximum deviation of the signal, as per Equation 9.

$$f_s = \frac{2.4 \text{ MHz}}{4 * 4} = 150 \text{ kHz} \tag{8}$$

$$75 \text{ kHz} > 25 \text{ kHz} + \left(\frac{9600 \text{ Hz}}{2}\right) \tag{9}$$

$$75 \text{ kHz} > 29.8 \text{ kHz} \tag{10}$$

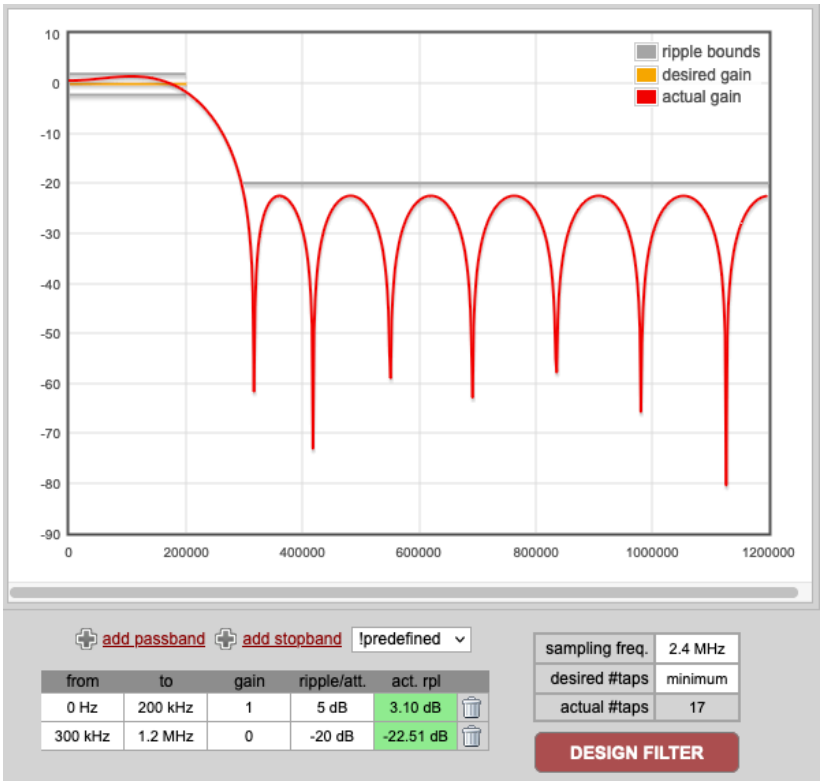

Figure 3: Frequency response of low-pass filter. The stopband attenuation had to be raised to −20 dB in order to execute fast enough in the first stage. The same filter is used for both the first and second stages.

Band-pass Filters

The space and the mark must be band-pass filtered from 15 kHz and 25 kHz using the filters in Figure 4 and Figure 5, respectively. A design concern is prevention of intersymbol interference, thus the maximum kernel length for each FIR filter should only be able to fit a single FSK symbol interval inside of it. Therefore, these band-pass filters should be limited to 15 elements or less, as shown in Equation 11. Improvements to these filters could come from pipelining multiple filters together, but due to the limited number of MACs available on the target device, this has not been done.

$$\frac{150 \text{ kHz}}{9600 \text{ Hz}} = 15.65 \text{ samples} \tag{11}$$

Envelope Detector

The signals coming out of each band-pass filter are first converted into an absolute-valued signal. After this, they are low pass-filtered using the filter in Figure 6.

Decider

The decider simply compares the values of the two envelope detectors and outputs the bit corresponding to the envelope of greater magnitude.

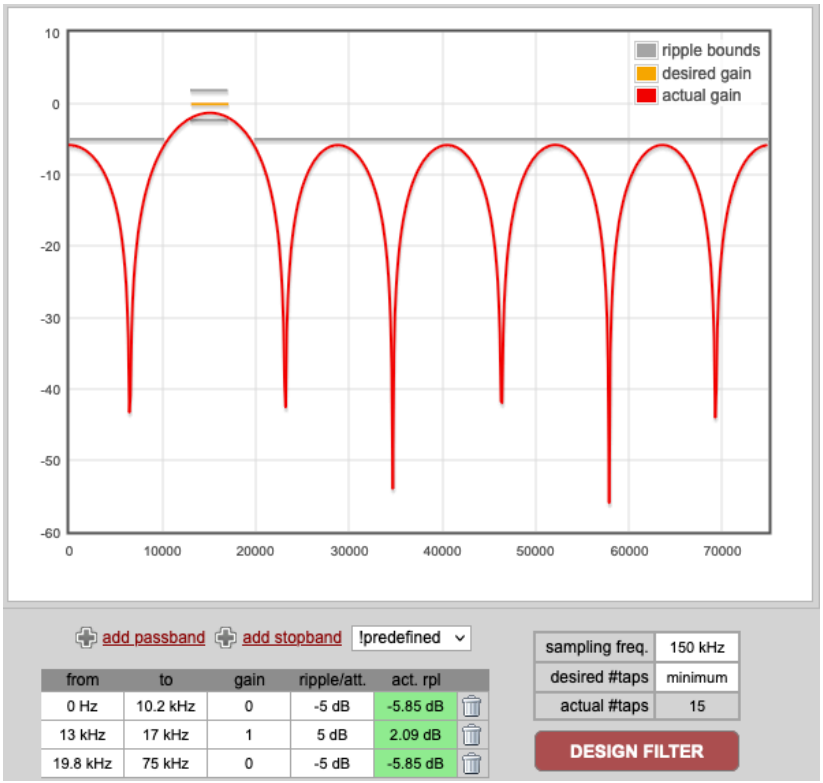

Figure 4: Band-pass filter for the space frequency, 15 kHz

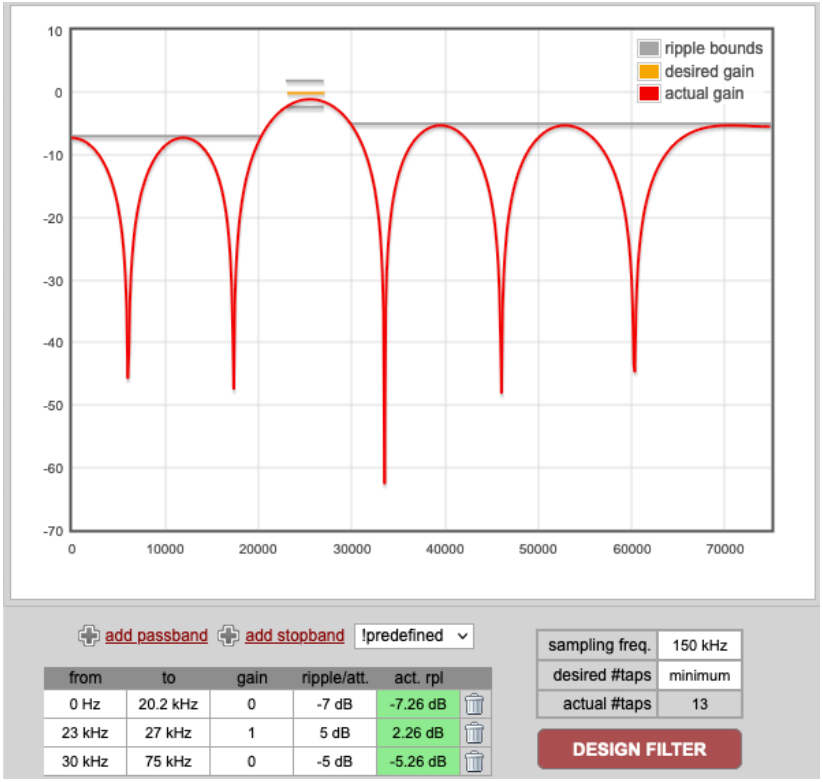

Figure 5: Band-pass filter for the mark frequency, 25 kHz

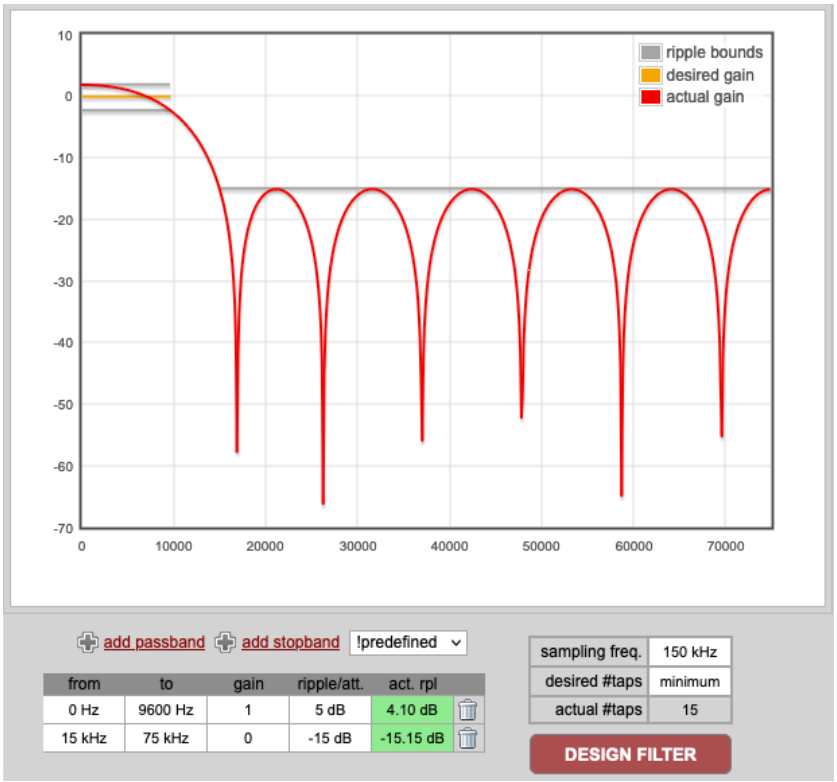

Figure 6: Envelope filter, a low-pass with 15 kHz cutoff

Simulation

The design was simulated, and the output waveforms can be seen in Figure 7. The “mark (uart rx)” line represents the bit going into the modulator. For simulation, this was randomly varied at the 9600 Hz baudrate. The “mixed” line is the signal coming out of the mixer. The 15 kHz and 25 kHz signals can be seen riding on top of the much higher frequency harmonics. These harmonics are filtered out, and the signal is downsampled in the “LPF & Downsample” section. After this, the signal branches off into both the “Space Pipeline” and the “Mark Pipeline”. These envelopes are demonstrated to be the opposite of one another. These envelopes are then compared by the decider. The “mark (uart tx)” line represents the decision of the decider that is ultimately sent back via the same UART that it was received from.

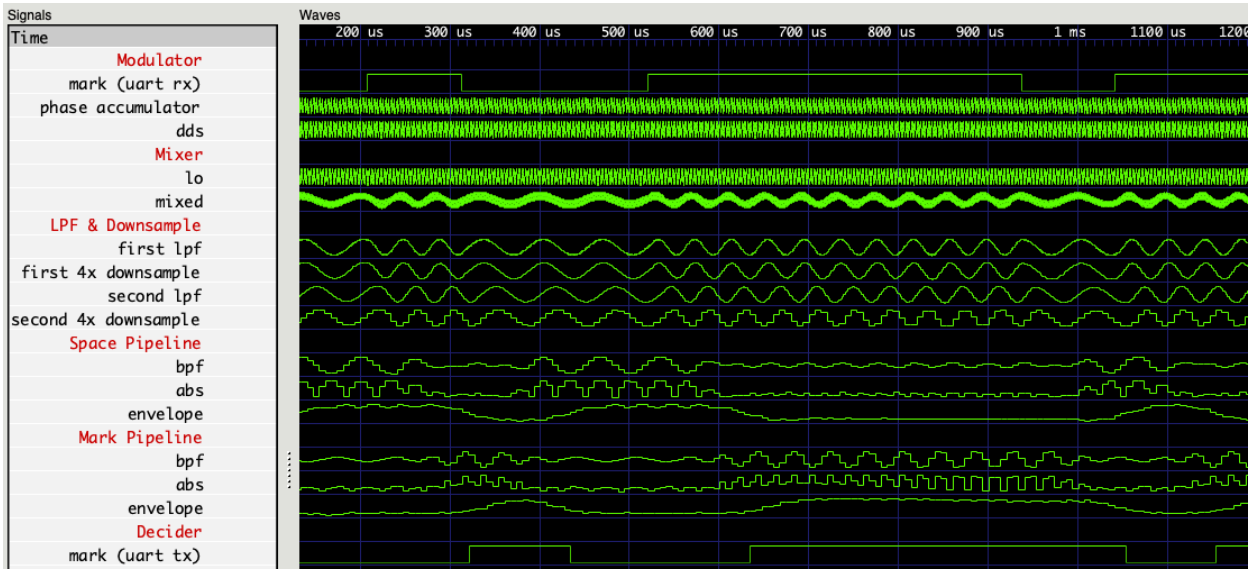

Figure 7: Full simulation waveform

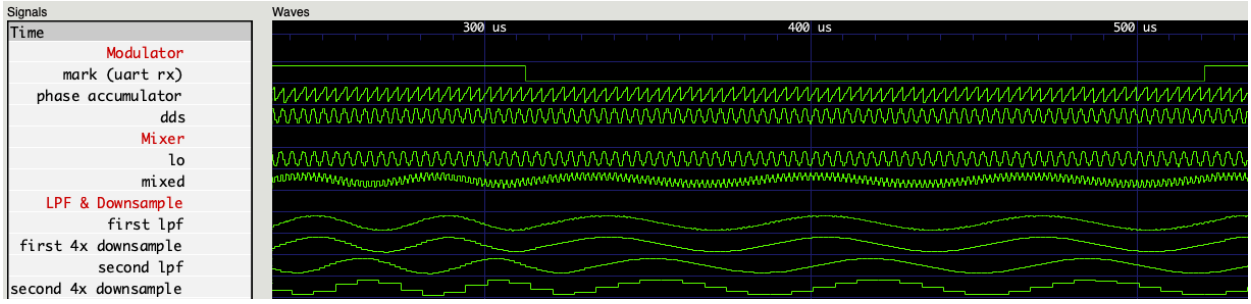

Figure 8: Close up of fast signals

Synthesis/Execution

The design synthesizes and meets timing. The maximum frequency is 53.11 MHz, faster than the 48 MHz reference that is on the board. Seven of the eight multipliers are used.

```
Device utilisation:
   ICESTORM_LC:  1635/ 5280    30%
  ICESTORM_RAM:    18/   30    60%
         SB_IO:     3/   96     3%
         SB_GB:     8/    8   100%
  ICESTORM_PLL:     0/    1     0%
```

```
   SB_WARMBOOT:      0/     1      0%
  ICESTORM_DSP:      7/     8     87%
ICESTORM_HFOSC:      0/     1      0%
ICESTORM_LFOSC:      0/     1      0%
        SB_I2C:      0/     2      0%
        SB_SPI:      0/     2      0%
        IO_I3C:      0/     2      0%
   SB_LEDDA_IP:      0/     1      0%
   SB_RGBA_DRV:      0/     1      0%
ICESTORM_SPRAM:      0/     4      0%
```

```
Max frequency for clock 'clock$SB_IO_IN_$glb_clk': 53.11 MHz (PASS at 48.00 MHz)
```

A TTL UART is connected to the TX/RX pins, and a character is sent over the interface. Figure 9 shows this captured with an oscilloscope.

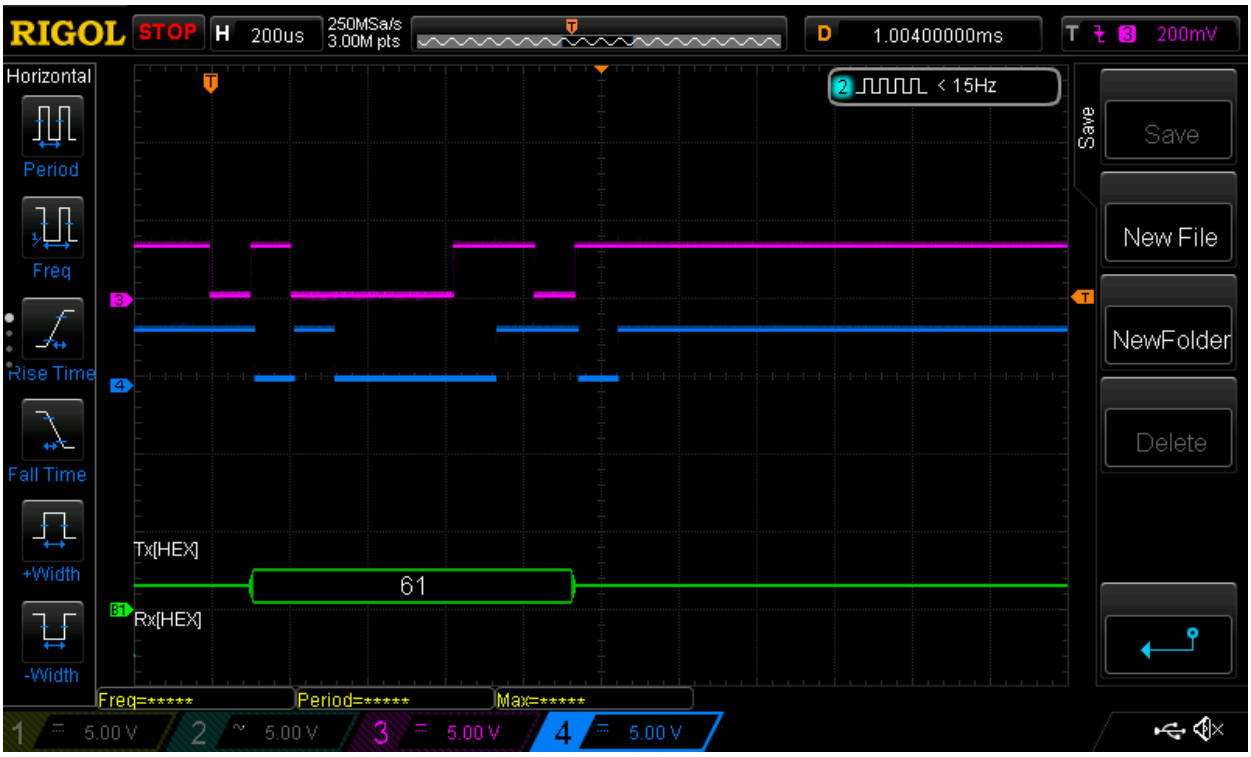

Figure 9: Oscilloscope capture of loopback of the 'a' character (0x61, 0b0110_0001). Note that UART is LSB first. Pink trace is UART RX, blue trace is UART TX