Research Project Week 4 Report

June 30, 2023

Adam Kamrath

1 BACKGROUND

This report covers the fourth week of the ongoing USRP research project. During the third week, MATLAB was set up in conjunction with two USRP devices. By the end of that week, a sine wave was successfully transmitted using one USRP and received on another. An attempt was also made to receive a signal from the ISC-MR102-USB card reader in the previous week, but unfortunately, it was unsuccessful.

2 INTRODUCTION

During the past week, notable advancements were made in the successful reception of signals from the card reader and the subsequent decoding of the data they contained. The primary focus of this report is to provide a comprehensive account of the signal reception process from the card reader and the subsequent analysis employed to extract a binary sequence.

3 METHODS

During the third week, Simulink was primarily used to operate the USRP devices. However, this week a decision was made to utilize MATLAB scripts instead of Simulink, as it provided greater flexibility for customizing the devices and signals.

At the beginning of this week, another sine wave was transmitted and successfully received on the USRP devices. This was done with the aim of gaining a better understanding of sample rates, sample time, and samples per frame. The following code snippet showcases the transmission of the sine wave:

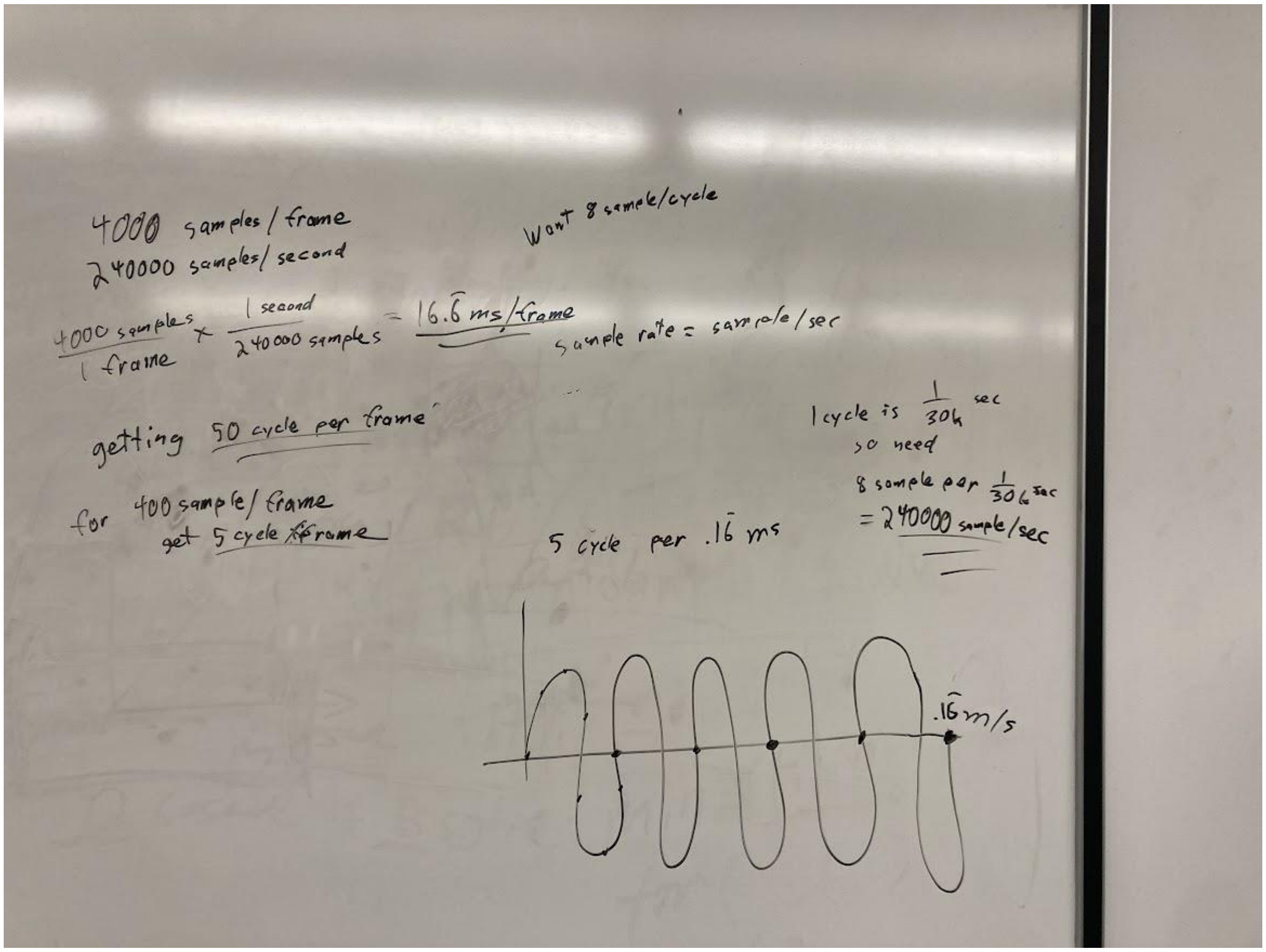
```
center_frequency = 13560000;
sampling_rate = 240000;
interpolation_factor = round(100000000/sampling_rate);

tx = comm.SDRuTransmitter(...
                    Platform = "N200/N210/USRP2", ...
                    IPAddress = '192.168.10.2', ...
                    MasterClockRate = 100e6, ...
                    InterpolationFactor = interpolation_factor, ...
                    Gain = 0, ...
                    CenterFrequency = center_frequency, ...
                    TransportDataType = "int16");

sinewave = dsp.SineWave(1,30e3);
sinewave.SampleRate = sampling_rate;
sinewave.SamplesPerFrame = 400;
sinewave.OutputDataType = 'double';
sinewave.ComplexOutput = true;
data = step(sinewave);

frameDuration = (sinewave.SamplesPerFrame)/(sinewave.SampleRate);
time = 0;
disp("Transmission Started");

while time < 30
    tx(data);
    time = time+frameDuration;
end
disp("Transmission Stopped");
release(tx);
```

As illustrated above, a sampling rate of 240,000 Hz was employed for transmitting the sine wave. The interpolation factor was determined by dividing the master clock of the transmitter, which operated at 100 MHz, by our sample rate (rounded to the nearest integer as the transmitter requires an integer value). The sine wave was transmitted over a frequency of 13.56 MHz, using a frequency of 30 KHz, and each frame consisted of 400 samples.

Here is the MATLAB code for the receiver:

```matlab
sampling_rate = 240000;
decimation_factor = 100000000/sampling_rate;
center_frequency = 13560000;


rx = comm.SDRuReceiver(...
                    Platform = "N200/N210/USRP2", ...
                    IPAddress = '192.168.10.3', ...
                    OutputDataType = "double", ...
                    DecimationFactor = decimation_factor, ...
                    Gain = 0, ...
                    CenterFrequency = center_frequency, ...
                    SamplesPerFrame = 400);

frameduration = (rx.SamplesPerFrame)/(sampling_rate);
time = 0;
timeScope = timescope(TimeSpanSource = "Property",...
                      TimeSpan = 5/30e3,SampleRate = sampling_rate);
timeScope.YLimits = [-1,1];
disp("Reception Started");

while time < 10
  data = rx();
  timeScope(data)
  time = time + frameduration;
end

disp("Reception Stopped");
release(timeScope);
release(rx);
```

The sampling rate for the receiver was once again configured at 240,000, and the samples per frame were set to 400. The choice of a sampling rate of 240,000 was motivated by the desire to capture 8 samples per cycle of the sine wave. This selection ensures an accurate representation of the sine wave's shape without introducing excessive jaggedness in the signal. Similarly, the decision to use 400 samples per frame was made to accommodate 5 cycles of the sine wave within each frame. The mathematical calculation used to arrive at this value is presented below:



The subsequent task focused on receiving a signal from the card reader, which necessitated the connection of an antenna to the reader. Initially, it was assumed that the card reader was transmitting the signal while the attached antenna received it. However, it was discovered that the attached antenna actually transmits the signal, and the reader functions as the receiver. The MATLAB code utilized for signal reception is provided below:

```
center_frequency = 13560000;
sampling_rate = 200000;
decimation_factor = round(100000000/sampling_rate);

rx = comm.SDRuReceiver(...
                    Platform = "N200/N210/USRP2", ...
                    IPAddress = '192.168.10.2', ...
                    OutputDataType = "double", ...
                    DecimationFactor = decimation_factor, ...
                    Gain = 0, ...
                    CenterFrequency = center_frequency, ...
                    SamplesPerFrame = 200000);

timeScope = timescope(TimeSpanSource = "auto",...
                      TimeSpan = 5,SampleRate = sampling_rate, ...
                      Position=[20,200,800,500]);
timeScope.YLimits = [-.6,.6];


disp("Reception Started");
frameduration = (rx.SamplesPerFrame)/(sampling_rate);
time = 0;
while time < 5
  disp(time);
  data = rx();
  timeScope(data)
  time = time + frameduration;
end

disp("Reception Stopped");
release(timeScope);
release(rx);
```

These signals were also written to a .bb file so then could be analyzed without having to use the receiver. The code to save the waveform is here:

```
sampling_rate = 240000;
decimation_factor = round(100000000/sampling_rate);
center_frequency = 13560000;

rx = comm.SDRuReceiver(...
                    Platform = "N200/N210/USRP2", ...
                    IPAddress = '192.168.10.2', ...
                    OutputDataType = "double", ...
                    DecimationFactor = decimation_factor, ...
                    Gain = 0, ...
                    CenterFrequency = center_frequency, ...
                    SamplesPerFrame = 400);

rxWriter = comm.BasebandFileWriter('capture.bb', ...
         SampleRate=sampling_rate,CenterFrequency=center_frequency);

frameduration = (rx.SamplesPerFrame)/(sampling_rate);
frames = 4000;
end_time = frames * frameduration;

timeScope = timescope(TimeSpanSource = "Property",...
                      TimeSpan = 5/30e3,SampleRate = sampling_rate);
timeScope.YLimits = [-.2,.2];
disp("Reception Started");

time = 0;
while time < end_time
  data = rx();
  timeScope(data)
  rxWriter(data);
  time = time + frameduration;
end

disp("Reception Stopped");
info(rxWriter)
release(rxWriter);
release(timeScope);
release(rx);
```

And here is the code to plot the data in the .bb files:

```
sampling_rate = 240000;
decimation_factor = 100000000/sampling_rate;
center_frequency = 13560000;

reader = comm.BasebandFileReader('capture.bb', 400);

timeScope = timescope(TimeSpanSource = "Property",...
                      TimeSpan = 5/30e3,SampleRate = sampling_rate);
timeScope.YLimits = [-.2,.2];
data = reader();
timeScope(data);

release(timeScope);
release(reader);
```

After successfully receiving the signal from the card reader, the waveform was thoroughly analyzed to determine the underlying data transmission mechanism. This analysis was conducted in alignment with the ISO15693 documents, which served as a valuable reference. Subsequently, the binary sequence corresponding to the Inventory command was deciphered as it is the primary command used it this research project.

4 DISCUSSION OF RESULTS

The following image is the waveform the sine wave being transmitted and received by the USRP devices:

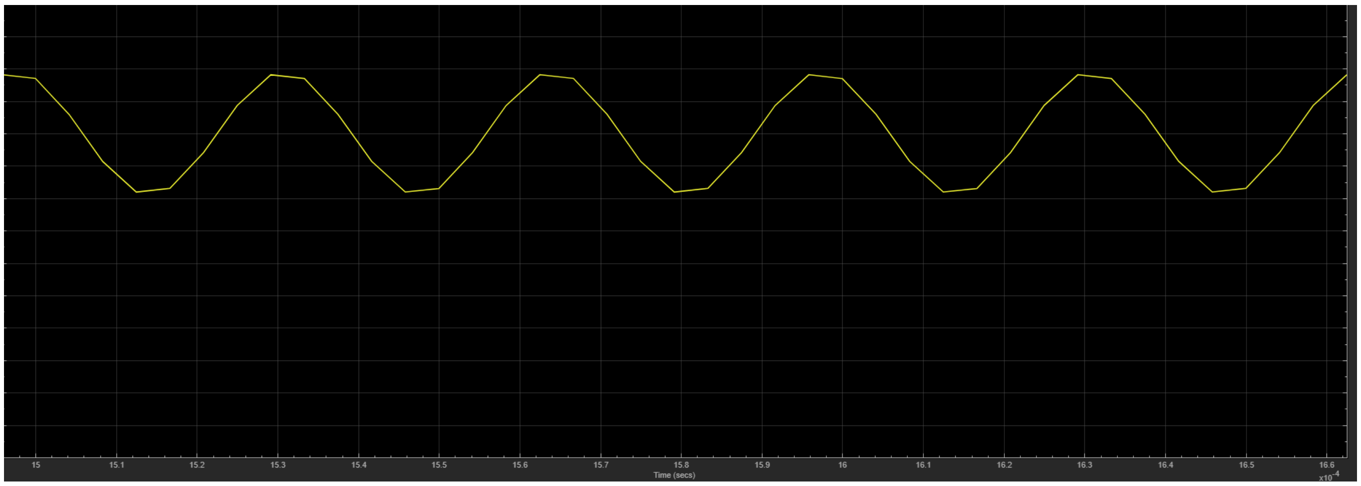
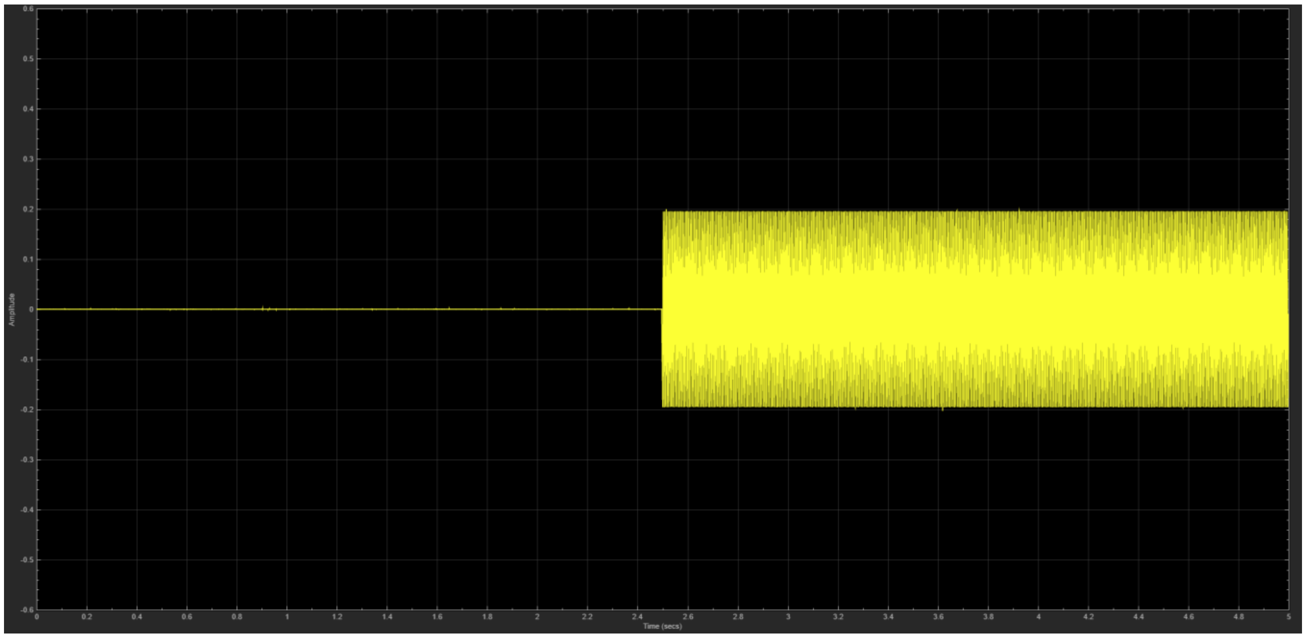


Figure 1: Sine Wave Reception

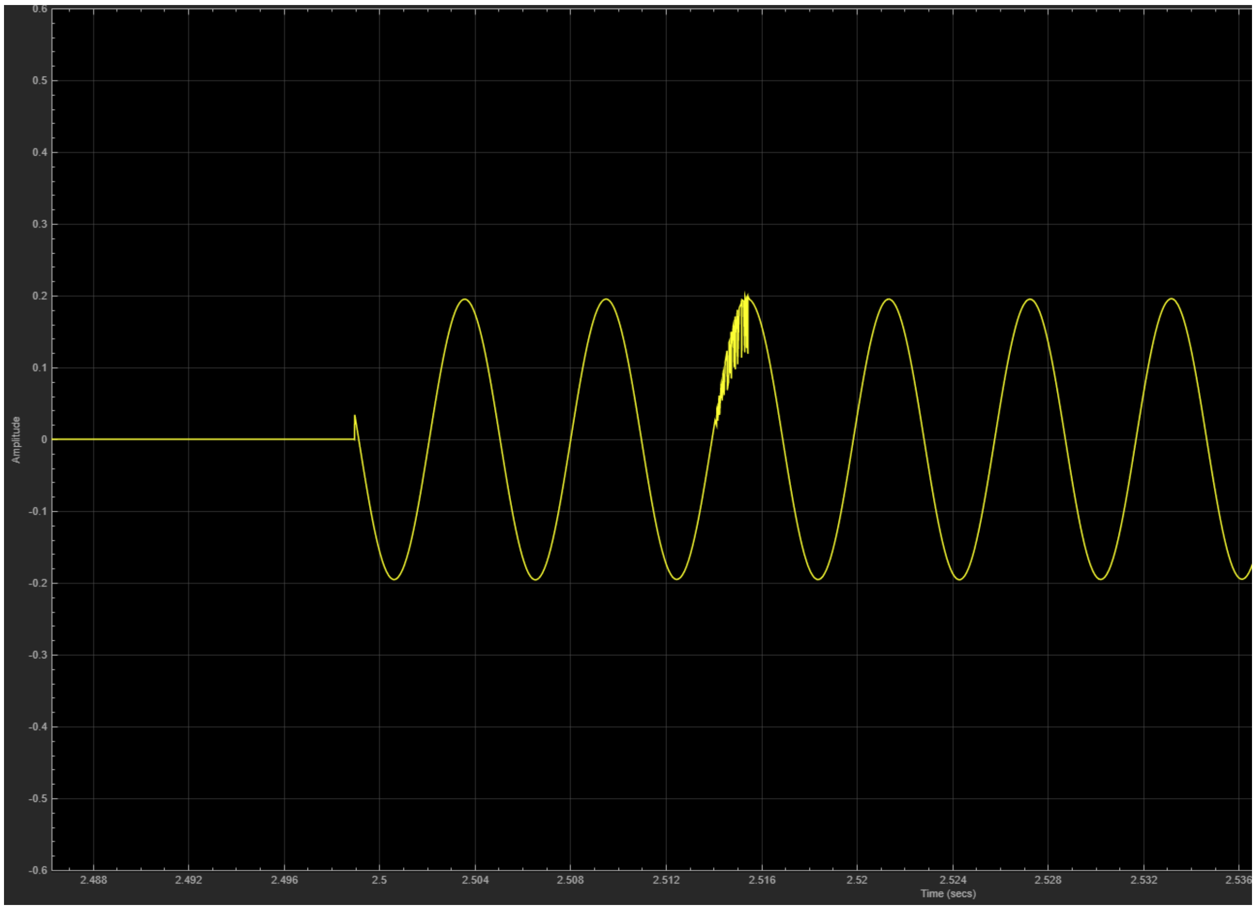
Figure 1 displays a single frame of the transmitted sine wave, demonstrating approximately 5 cycles as anticipated based on the aforementioned calculations. The figure also provides a clear representation of the eight samples per cycle, although some slight jaggedness is observed due to the limited number of samples. Nevertheless, the sine wave's characteristic shape is discernible, aligning with the primary objective of this experiment.

Next, the signal from the card reader was successfully received. It was discovered that when receiving the signal in real-time, the time scope only displayed the final frame of data. To visualize all the frames of data, it was necessary to save the signal to a .bb file and subsequently read and display it. The following example provides a zoomed-out view of the system information command signal:

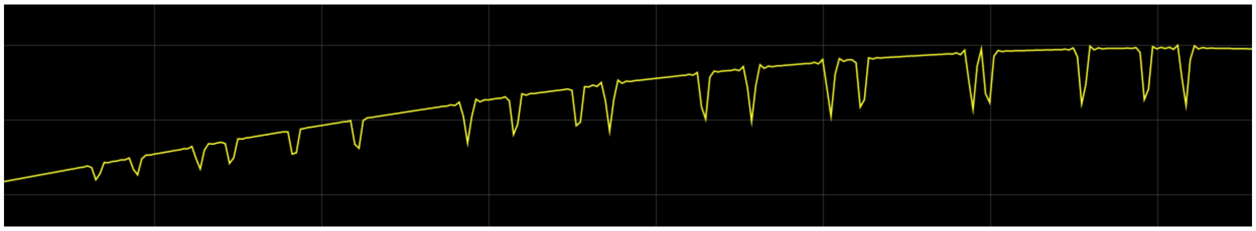


The initial portion of the signal appears to be blank or devoid of any significant data. This observation can be attributed to the fact that the command was not transmitted until approximately halfway through the receiving process. Thus, the absence of content in the signal during the initial period is a result of the delayed transmission of the command.

Below is the zoomed in version of the waveform:

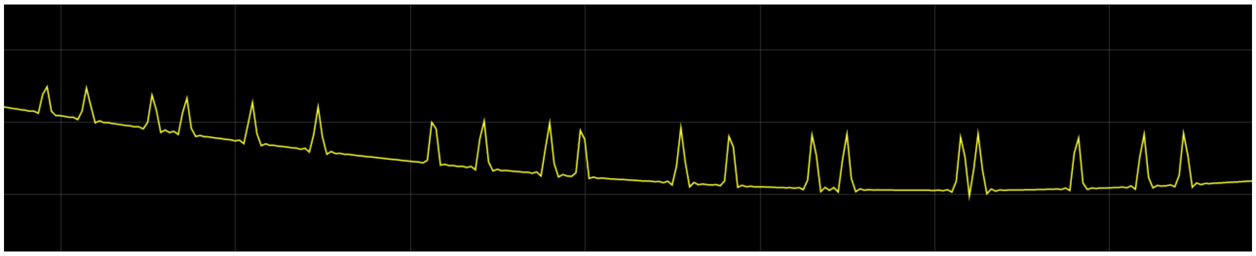


An evident observation is the presence of a continuous sine wave emitted by the card reader. This sine wave represents the RF field generated by the reader and is not part of the actual data being transmitted. However, upon closer inspection, in the middle of the displayed waveform, distinct pulses become apparent. These pulses correspond to the data contained within the signal. When zoomed in, the waveform reveals a clearer representation of these data pulses:



Indeed, it is evident that certain pulses within the waveform exhibit varying amplitudes. This behavior can be attributed to the utilization of amplitude shift keying (ASK) by the card reader, with a modulation index of 10%. The variation in pulse size occurs because a 10% decrease from a smaller amplitude results in a smaller absolute change compared to a 10% decrease from a larger amplitude.

Moreover, the displayed waveform predominantly represents the positive part of the sine wave, which explains why the pulses are shown as downward deflections. When the command occurs during the negative part of the sine wave, it is expected that the pulses will be depicted as upward deflections. This behavior can be visualized in the waveform shown below:



To decode the content of the system information command sent by the card reader, reference is made to the ISO15693 documents. Within these documents, a specific figure is employed to illustrate the start of frame for the 1 out of 4 coding method, which is utilized by the card reader:

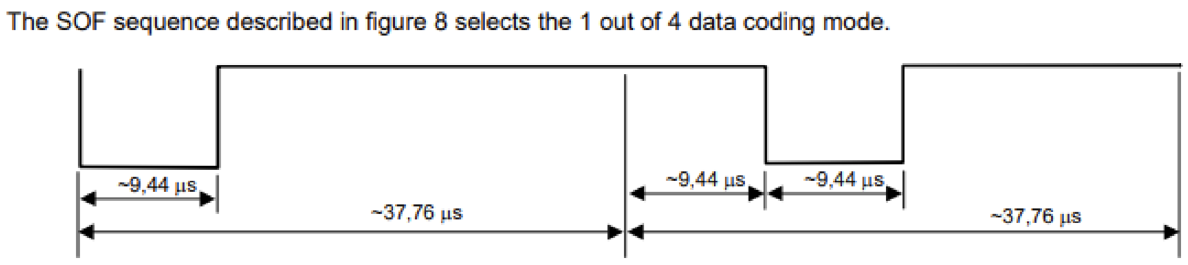
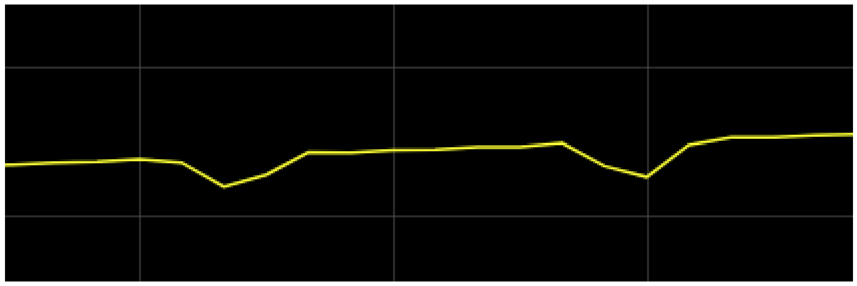

The SOF sequence described in figure 8 selects the 1 out of 4 data coding mode.
~9,44 µs
~37,76 µs
~9,44 µs
~9,44 µs
~37,76 µs

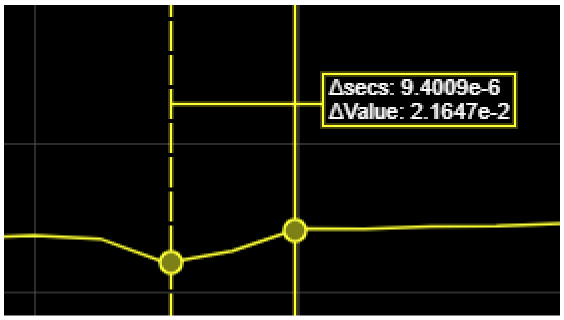
Figure 2: 1 out of 4 Start of Frame

It can be seen that the start of frame has two dips that are 9.44us long. The following is a zoom in picture of the start of frame of the signal:

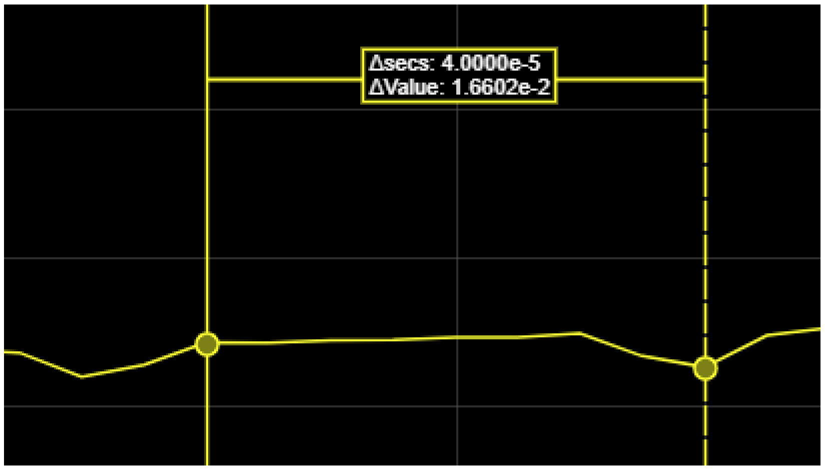


Note: The **bottom or top of a dip or peak** indicates when the signal goes **low** in a diagram like Figure 2. When the signal gets **out of** the dip or peak is when the signal is high again.

With this in mind we can see in Figure 2 that the signal starts LOW for 9.44us. So the signal starts at the **bottom** of the dip.

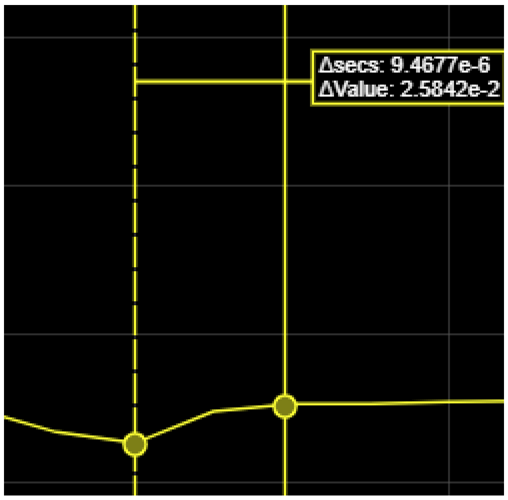


Then Figure two shows that the signal will be HIGH for 37.76us.

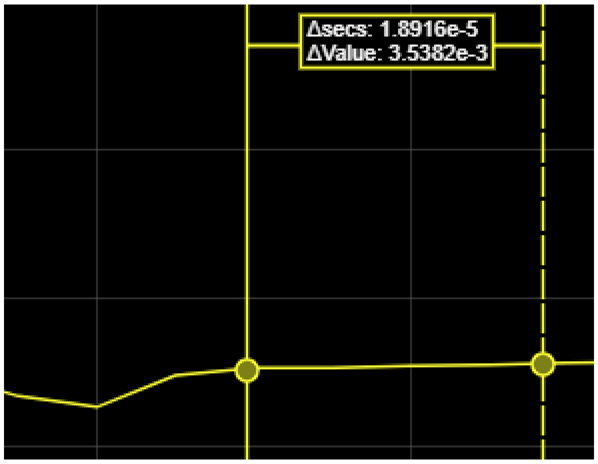


Technically the change in seconds should be 37.76us based on the given duration in Figure 2. However, due to potential limitations or inaccuracies in the waveform representation, the precise measurement may not always be perfectly accurate. Nonetheless, despite any minor discrepancies, the waveform provides a close approximation that allows for a clear understanding of the underlying signal characteristics and patterns.

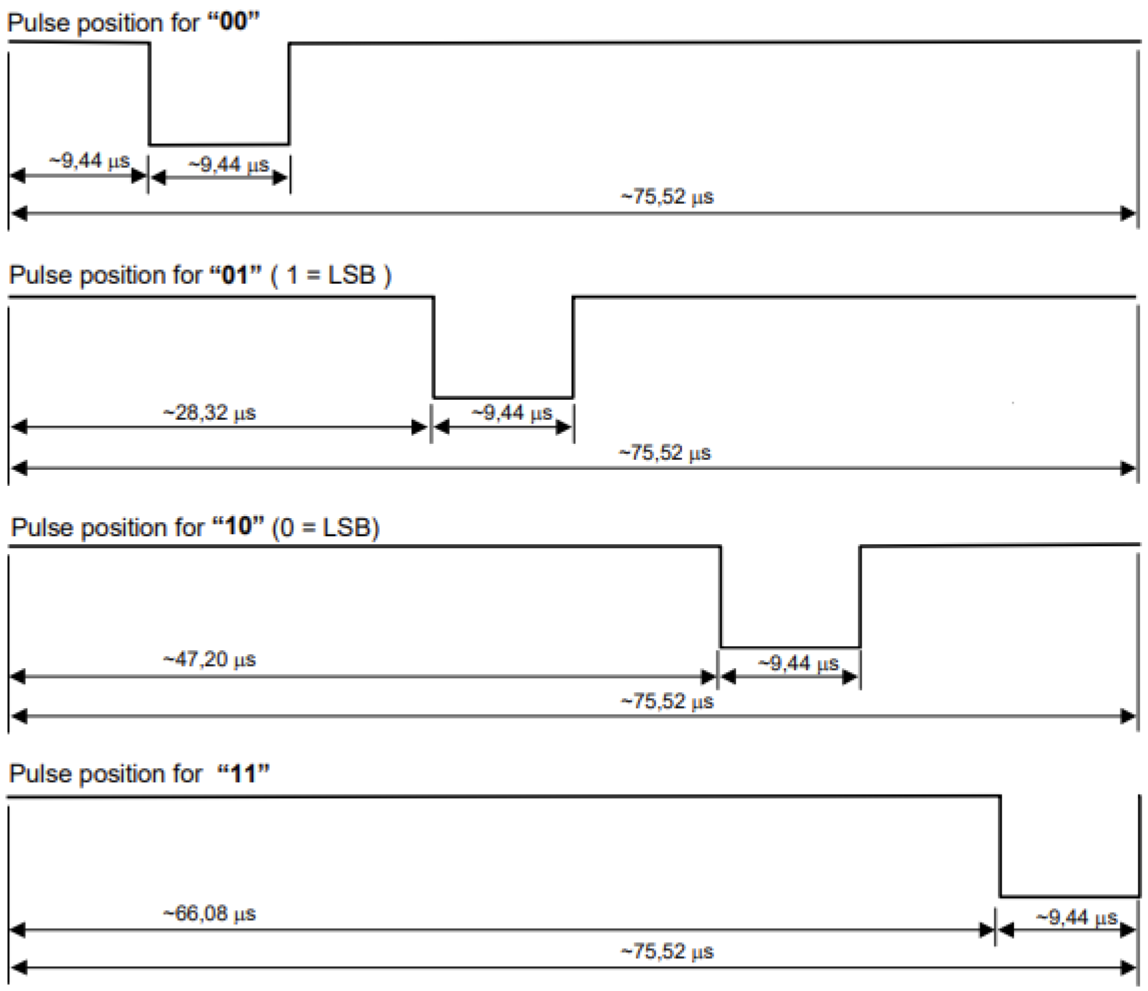
Then the signal is LOW for 9.44us



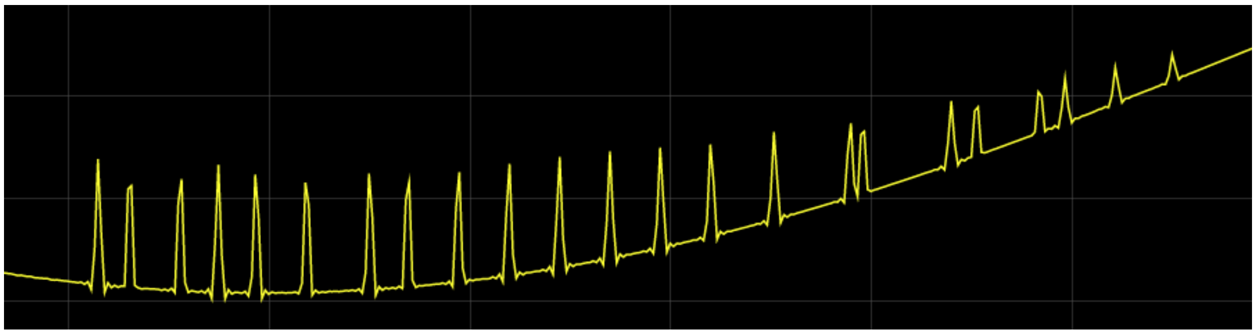
Then it is HIGH for 18.88us



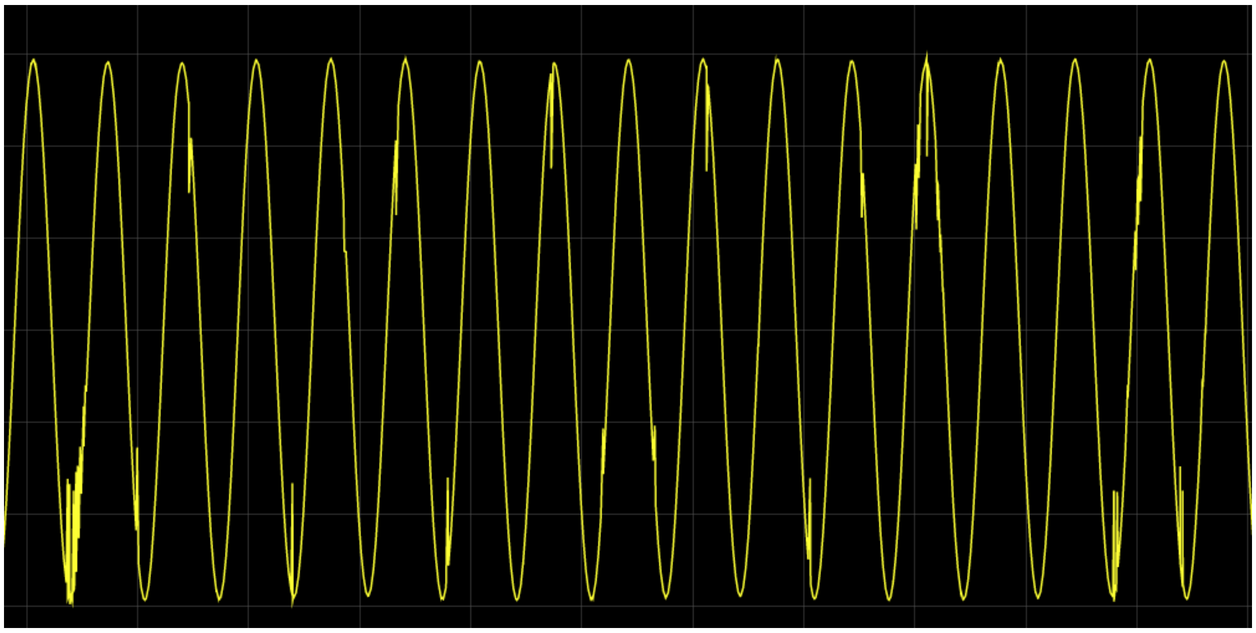
Following the completion of the start of frame, a similar technique can be applied to decipher the remaining portions of the waveform. The following figures serve as guides to extract binary bits from the waveforms, with the transmission order being the Least Significant Byte (LSB) first:



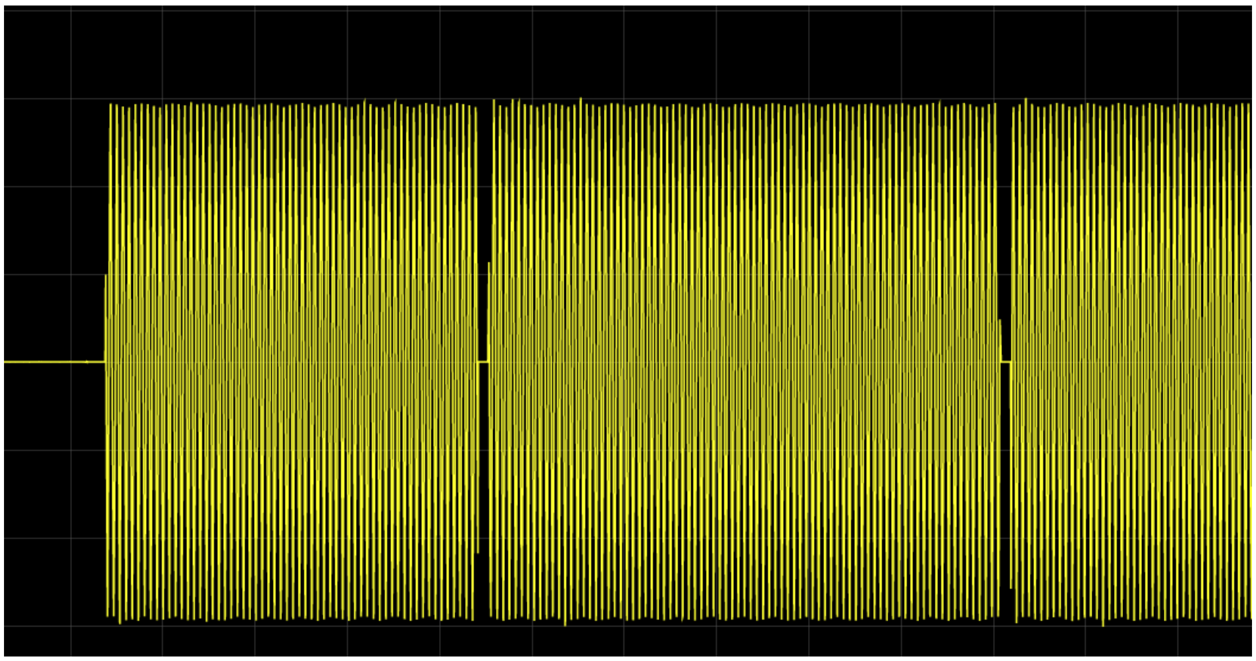
Upon applying the same technique to decode the inventory command, it was determined that the card reader transmitted the hexadecimal value 0x06010009CD. The corresponding waveform for this command is depicted below:



For the inventory command specifically there is more pulses in the waveform after the data is transmitted:



The provided sequence represents a complete cycle of an inventory request. It is believed that the pulses following the data correspond to the anticollision sequence, which requires further investigation to fully understand its behavior. Additionally, it is worth noting that before the transmission of each inventory command, there is consistently a pause observed before the waveform. Below is a sequence demonstrating three consecutive inventory requests that were sent out:



5 CONCLUSION

This week has been highly valuable in gaining insights into the waveform requirements for transmitting commands to the transponder cards. In the upcoming week, the focus will shift towards applying a filter to the signal in an attempt to enhance the decoding process. Additionally, attention will be directed towards analyzing the response received from the card, further contributing to a comprehensive understanding of the system..

6 REFERENCES

comm.SDRuTransmitter System Object - MATLAB & Simulink." MathWorks. Accessed June 28, 2023. URL: https://www.mathworks.com/help/supportpkg/usrpradio/ug/comm.sdrutransmitter-system-object.html.

comm.SDRuReceiver System Object - MATLAB & Simulink." MathWorks. Accessed June 28, 2023. URL: https://www.mathworks.com/help/supportpkg/usrpradio/ug/comm.sdrureceiver-system-object.html.

comm.BasebandFileWriter System Object - MATLAB & Simulink." MathWorks. Accessed June 28, 2023. URL: https://www.mathworks.com/help/comm/ref/comm.basebandfilewriter-system-object.html.

timescope - MATLAB & Simulink." MathWorks. Accessed June 28, 2023. URL: https://www.mathworks.com/help/dsp/ref/timescope.html.

comm.BasebandFileReader System Object - MATLAB & Simulink." MathWorks. Accessed June 28, 2023. URL: https://www.mathworks.com/help/comm/ref/comm.basebandfilereader-system-object.html