Week 6 Research Report

Adam Kamrath

July 2023

1 Background

This report provides an update on the progress made during week 6 of the ongoing USRP research project. In the previous week, significant advancements were made in simplifying the signal received from the ISC-MR102-USB card reader using filters. However, despite our efforts, we encountered difficulties in receiving the signal transmitted by the RF-ID card.

2 Introduction

Throughout the week, we achieved a major breakthrough by successfully capturing the response signal from an RF-ID card. Additionally, significant progress was made in converting the baseband file data into a digital signal composed of distinct high and low signals. Moreover, we initiated the process of automating the extraction of binary sequences from these signals.

3 Methods

During the start of the week, the initial task was to successfully receive the signal from the RF-ID card. To ensure optimal reception, the card was placed in close proximity to the receiver antenna (almost touching) while the transmission antenna was positioned approximately 1.5 ft away from the receiver.

Upon examining the received data using both a high-pass filter and a band-pass filter, interesting observations were made concerning the RF-ID card signal. The analysis revealed that the card signal appeared within the band-pass data, but its strength was relatively weak. Conversely, when investigating the high-pass data, a stronger representation of the signal was evident. This can be seen from Figures 1 and 2.

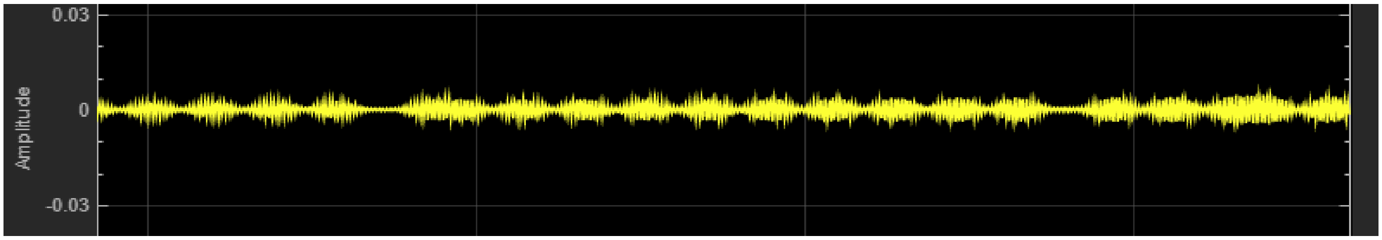

Figure 1: Card Signal in Band-pass

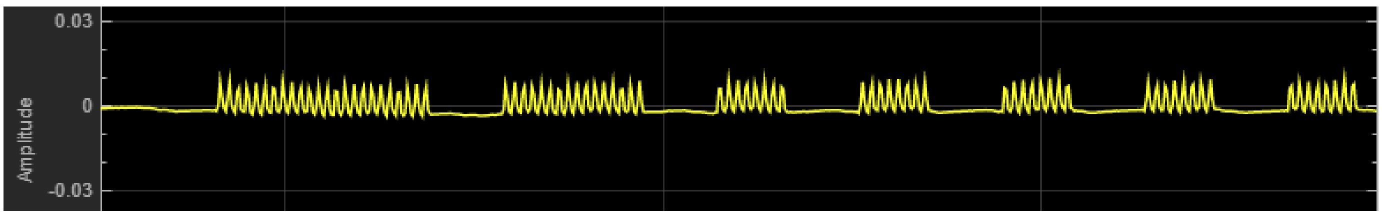

Figure 2: Card Signal in High-pass

It is also worth noting that by plotting the absolute value of the data leads to a much easier time understanding and analyzing the waveform.

The next step involves editing the baseband file to facilitate data observation. In MATLAB, baseband files are stored in an Nx1 matrix, with N representing the number of samples. This matrix-based structure simplifies data manipulation, as values can be easily accessed and modified through indexing.

The objective of editing the code is to convert the data into two possible values: 1 or 0. The method is straightforward: the program loops through all the samples in the baseband file and sets a threshold. If a sample value is above the threshold, it is changed to 1, and if it is below the threshold, it is changed to 0. The code (Code 1) below illustrates the program used to edit the baseband file.

Code 1: Baseband File Edit Code

```
function data = editData(data)
    for i = 1:size(data)
        if data(i) < .004
            data(i) = 0;
        else
            data(i) = 1;
        end
    end
end
```

After successfully receiving and editing the RF-ID card signal, the next step involved decoding the signal and extracting the corresponding binary sequence.

This process closely resembled the methodology used for obtaining the binary sequence from the ISC-MR102-USB card reader signal, albeit with different timings and signal pulses. As outline in the ISO15963 document, the RF-ID card emits the following signals, each carrying specific information:

- Logic 0
  - 8 pulses at 423.75Hz
  - Unmodulated time of 18.88us
- Logic 1
  - Unmodulated time of 18.88us
  - 8 pulses at 423.75Hz
- Start of Frame
  - Unmodulated time of 56.64us
  - 24 pulses at 423.75Hz
  - Logic 1
- End of Frame
  - Logic 0
  - 24 pulses at 423.75Hz
  - Unmodulated time of 56.64us

Using this documentation, the signal response from the card was initially decoded manually using the time scope in MATLAB. The signal response was a result of a system information request sent by the card reader device.

Following the manual decoding process, efforts were made to automate the decoding of these signals. The automation started with the automatic decoding of a system information request from the card reader. This was achieved using the 'decodeCardReader.m' file available on the research project GitHub page (https://github.com/adam-kamrath/USRP$_{R}esearch_{S}ummer$2023).

The program functions in the following manner: It begins at the first sample of the baseband file and identifies the first sample with a value of 1. Subsequently, it checks for the presence of a start of frame, as outlined in the ISO15963 document. If a start of frame is detected, the program proceeds to analyze the logic value of each pulse until it reaches the end of the frame.

Once the initial program was developed, the automation of the card signal decoding process was initiated. This new program shares a similar approach to the previous one but places greater emphasis on the number of consecutive pulses rather than the timing between them. The decodeCardSignal.m file on the research GitHub page represents the ongoing work on this program, although it is not fully complete at this stage.

4 Results and Discussion

Regarding the data editing process, the comparison between Figures 3 and 4 clearly illustrates the difference before and after editing the file. The edited file transforms into a digital signal, where all samples are represented as either 1 or 0. However, it is crucial to address a potential concern with the current setup. The editing program has only been tested on a single baseband file, which means that if the card signal is weaker or exhibits variations, the editing program may not have the correct threshold for accurate conversion. Consequently, some samples could be erroneously converted to 0s when they should be 1s, leading to an incorrect binary sequence during the decoding process.

To address this issue, it is essential to further validate and fine-tune the editing program on various baseband files, encompassing a range of signal strengths and characteristics. This validation process will help ensure the program's reliability and accuracy in handling different scenarios. By conducting comprehensive testing, the program can be optimized to handle a broader range of card signal variations and deliver precise binary sequences upon decoding.

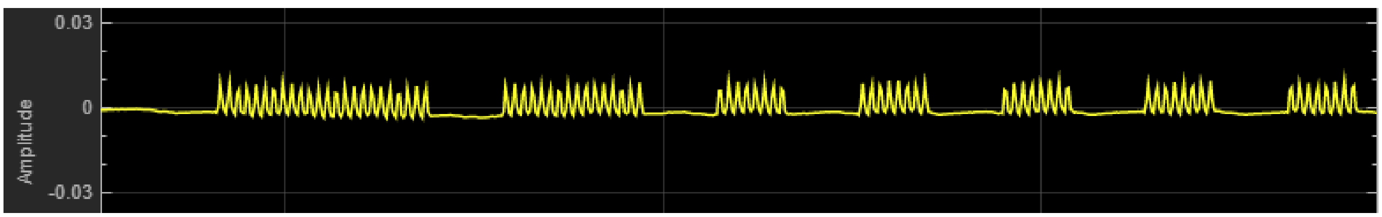

Figure 3: Unedited Signal

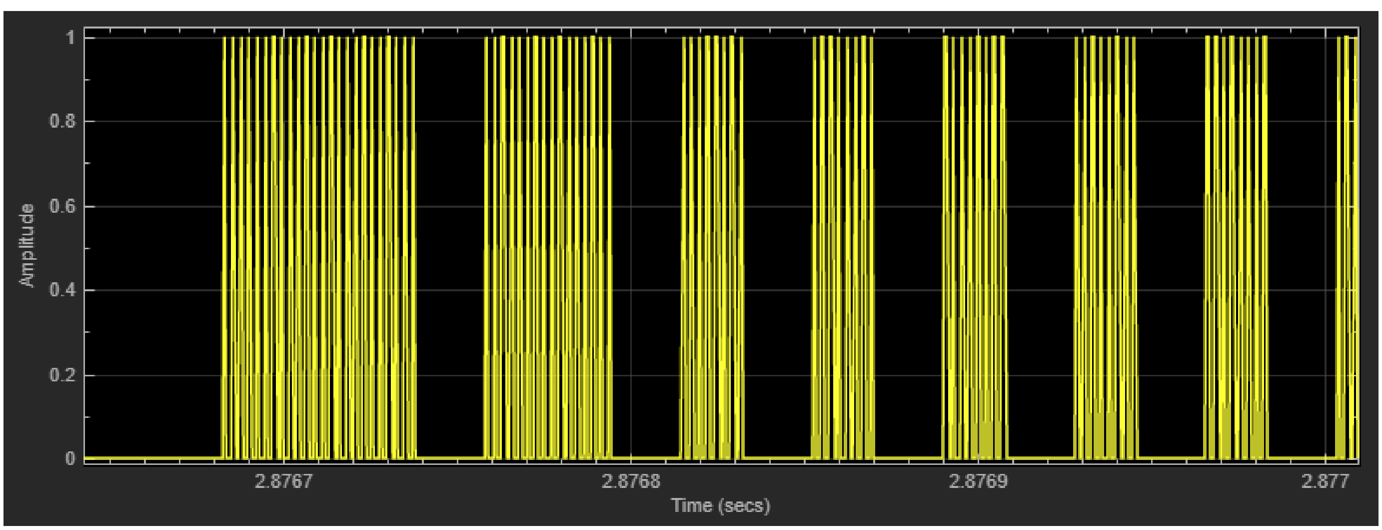

Figure 4: Edited Signal

Regarding the automated decoding process for the data signals, the program designed to decode the reader signal works flawlessly with the specific baseband file used for testing. It accurately displays the binary sequence transmitted by the card reader. However, it's important to reiterate that this program has only been tested on one baseband file, which leaves room for potential bugs to emerge when dealing with other files.

As for the card signal decoder programs, they are still in their early stages of development. The detection of the start of the frame in the signal can be successfully achieved, but currently, the process is relatively slow. The program's efficiency is being worked on to achieve real-time decoding of the card signal instead of relying on a baseband file.

5 Conclusion

Overall, this week witnessed notable achievements in capturing the RF-ID card's response signal, converting baseband file data into digital signals, and initiating the automation of binary sequence extraction. These advancements pave the way for further exploration and analysis in our ongoing research project.

6 References

GitHub Page: https://www.mathworks.com/help/supportpkg/usrpradio/ug/frequency-offset-calibration-with-usrp-tm-hardware.html