

# Extension and optimization of an interface to use multiple DVB-T USB receivers in a MIMO-OFDM transmission system

Yanfen Li

Supervised by

Prof. Dr. –Ing. Andreas Czylik

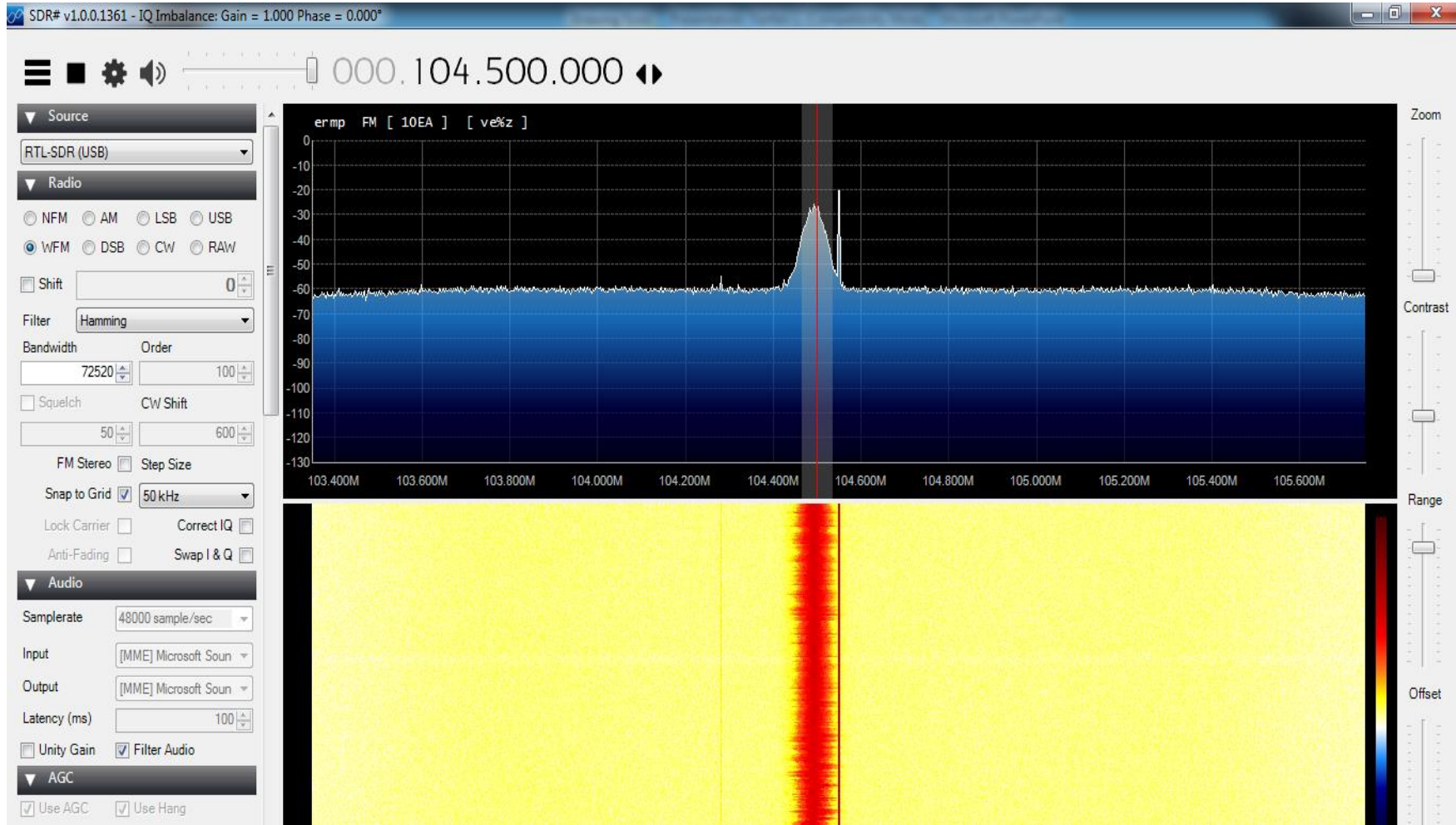


# Outline

- **Introduction**
- **Synchronization process**
- **Mex file**
- **Test results**
- **Conclusion**



# SDR-Software Defined Radio



# RTL-SDR: RTL2832U chipset



1. Tuner IC: Signal receiving
2. Data acquisition IC: Data sampling

# RTL-SDR

<b>Tuner</b>	<b>Frequency range</b>
Elonics E4000	52 - 2200 MHz with a gap from 1100 MHz to 1250 MHz
Rafael Micro R820T	24 - 1766 MHz
Rafael Micro R828D	24 - 1766 MHz
Fitipower FC0013	22 - 1100 MHz
Fitipower FC0012	22 - 948.6 MHz
FCI FC2580	146 - 308 MHz and 438 - 924 MHz

Table 1. Frequency range of different kind of tuners

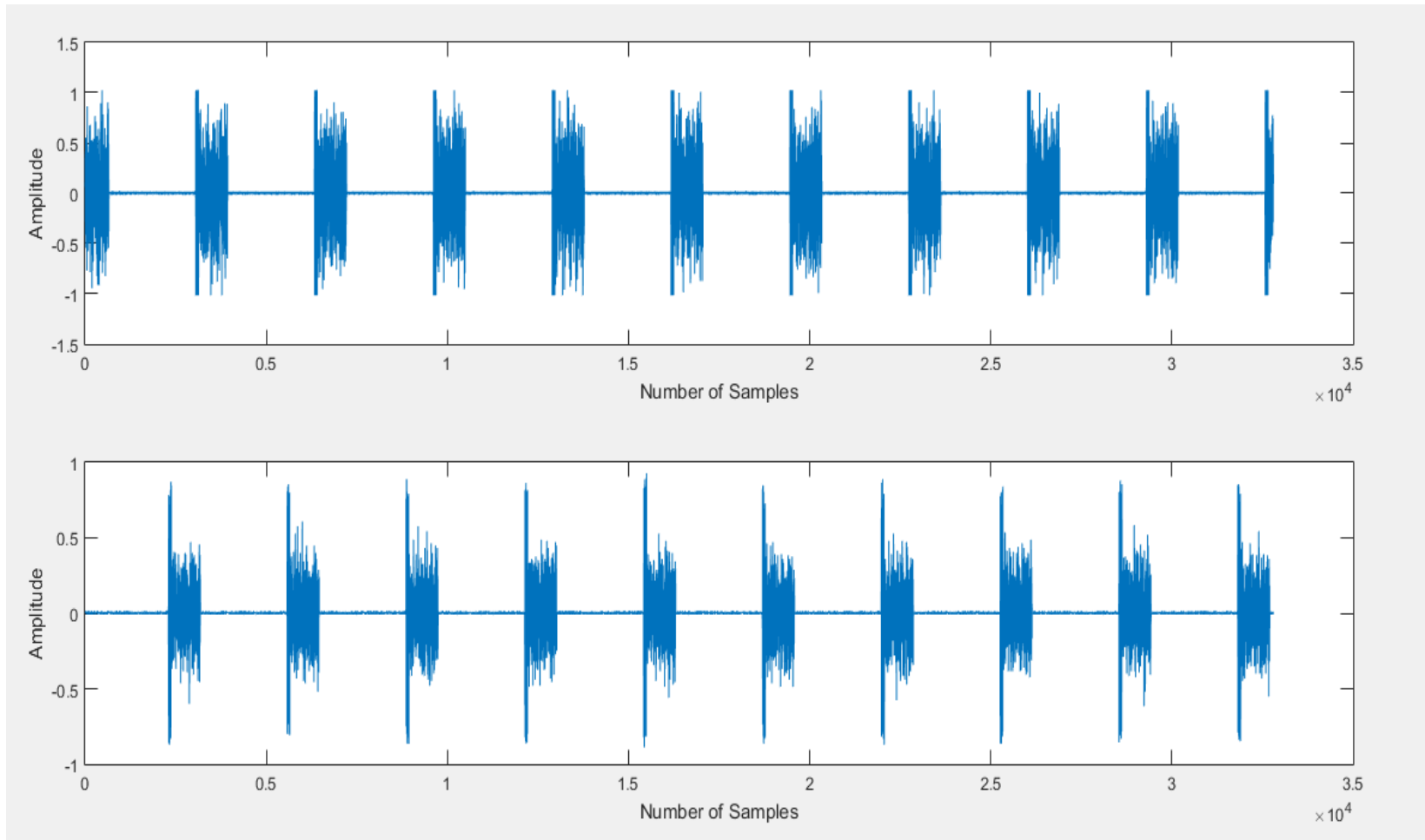
# Time synchronization

```
out_block_size: 3515
bytes_to_read: 70300
index: 0, freq: 800.000000, gainMode: 0, gain: 0, filename: 800.bin
index: 1, freq: 800.000000, gainMode: 0, gain: 0, filename: 801.bin
rtlsdr_get_device_count done: 2, opening 0
rtlsdr_get_device_count done: 2, opening 1
Found Rafael Micro R820T tuner
Found Rafael Micro R820T tuner
Exact sample rate is: 1000000.026491 Hz
Exact sample rate is: 1000000.026491 Hz
wait for signal
wait for signal
Device index 1 start at 44525477 ms since system boot.
Device index 0 start at 44525477 ms since system boot.

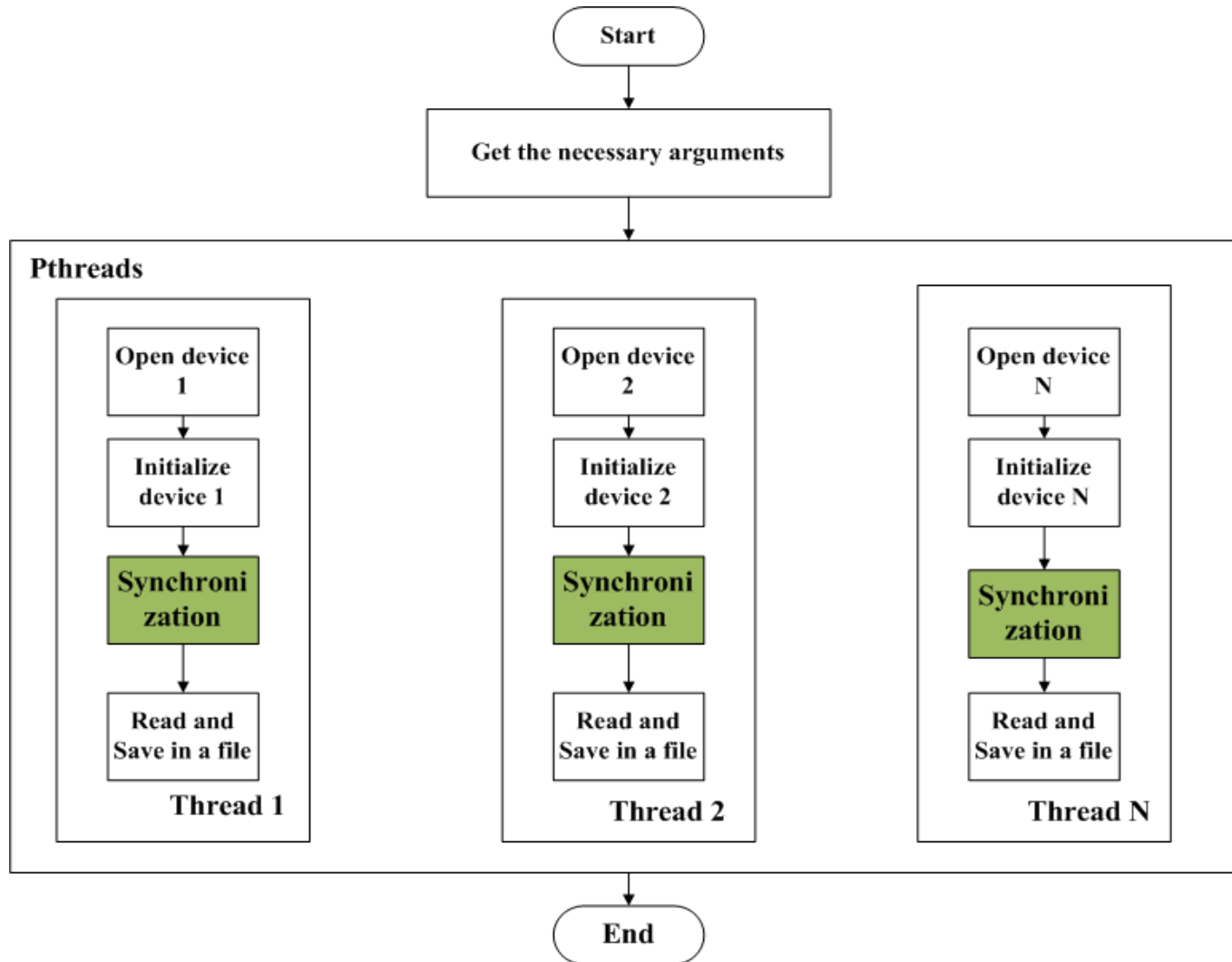
Process returned 0 (0x0)   execution time : 7.269 s
Press any key to continue.
```

Frequency range of tuner Rafael Mirco R820T: 24 - 1766 MHz

# Received data

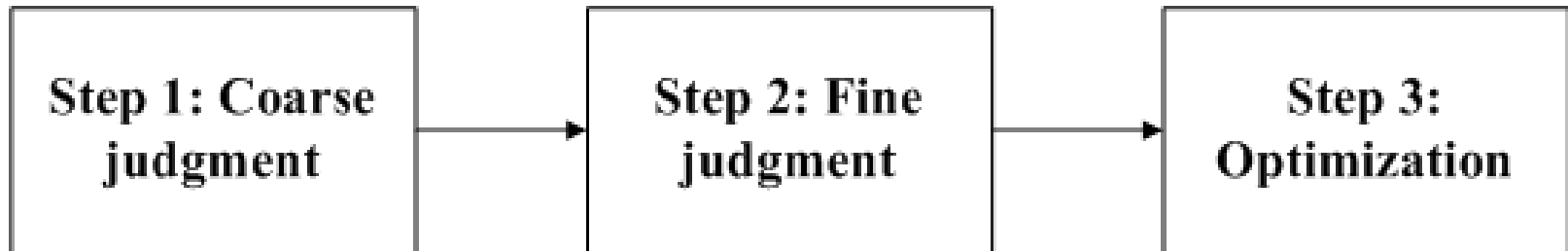


# Flowchart





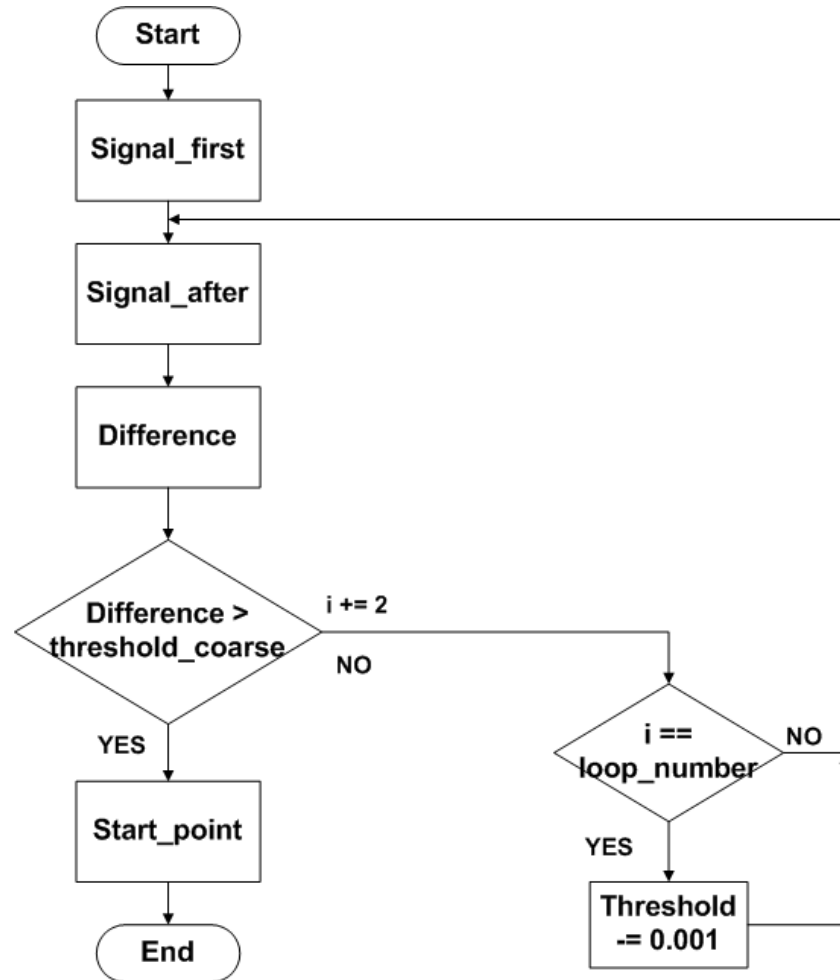
# Synchronization steps



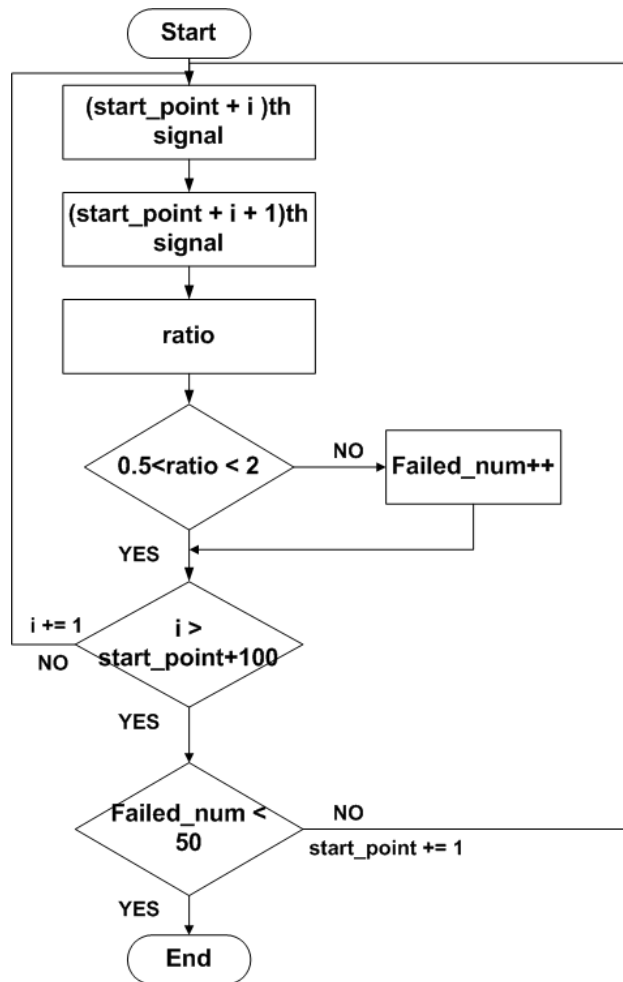
# Step 1 Coarse judgment

threshold\_coarse: 2

loop\_number: samples read  
in every loop



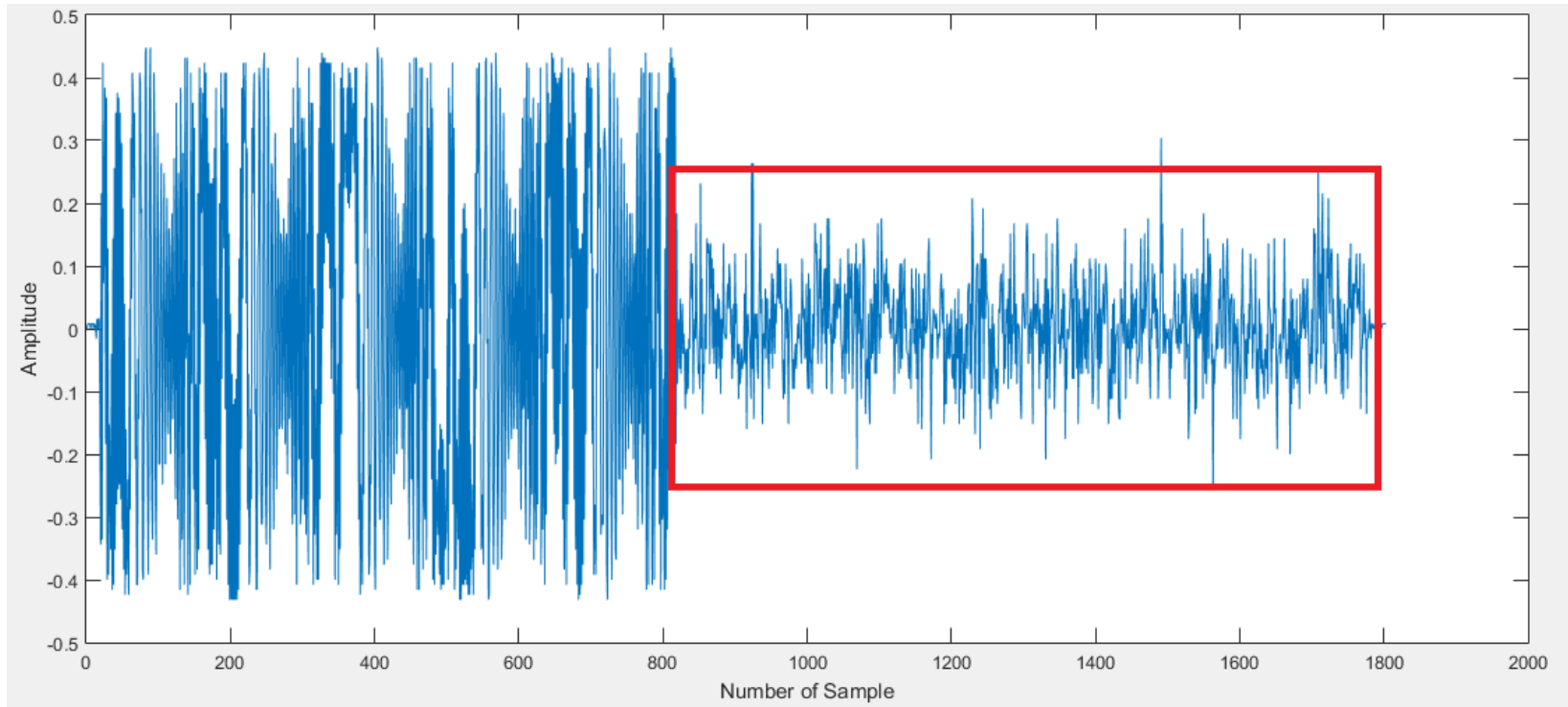
# Step 2 Fine judgment



100 signals are chosen start from  $start\_point$

$failed\_num$ : used to judge the chosen  $start\_point$

# Step 3 Optimization

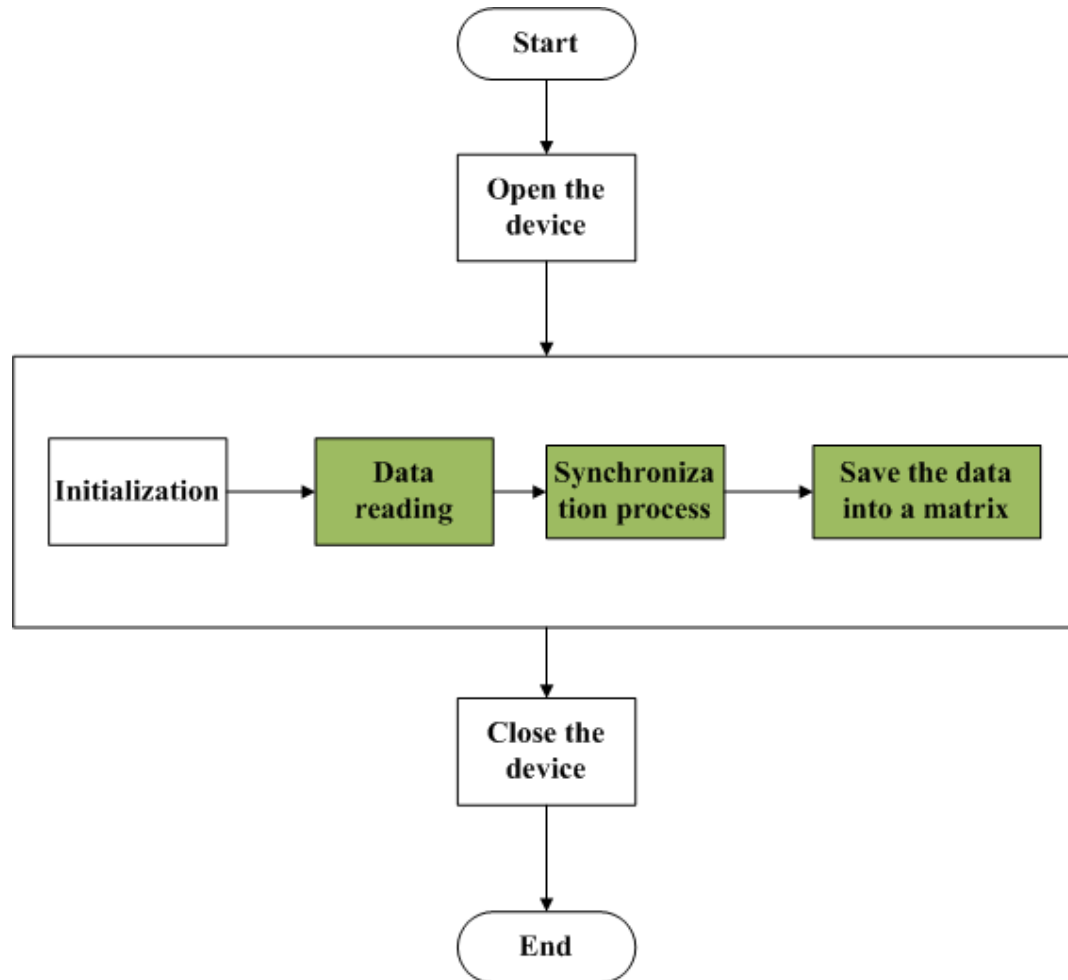


# Mex file

- Is an interface between C and MATLAB
- Can be called in MATLAB as a built-in function
- Gateway function: `void mexFunction()`



# Flowchart of mex file



# Program test

start trigger time

whole seconds  
2  
fractional seconds  
0

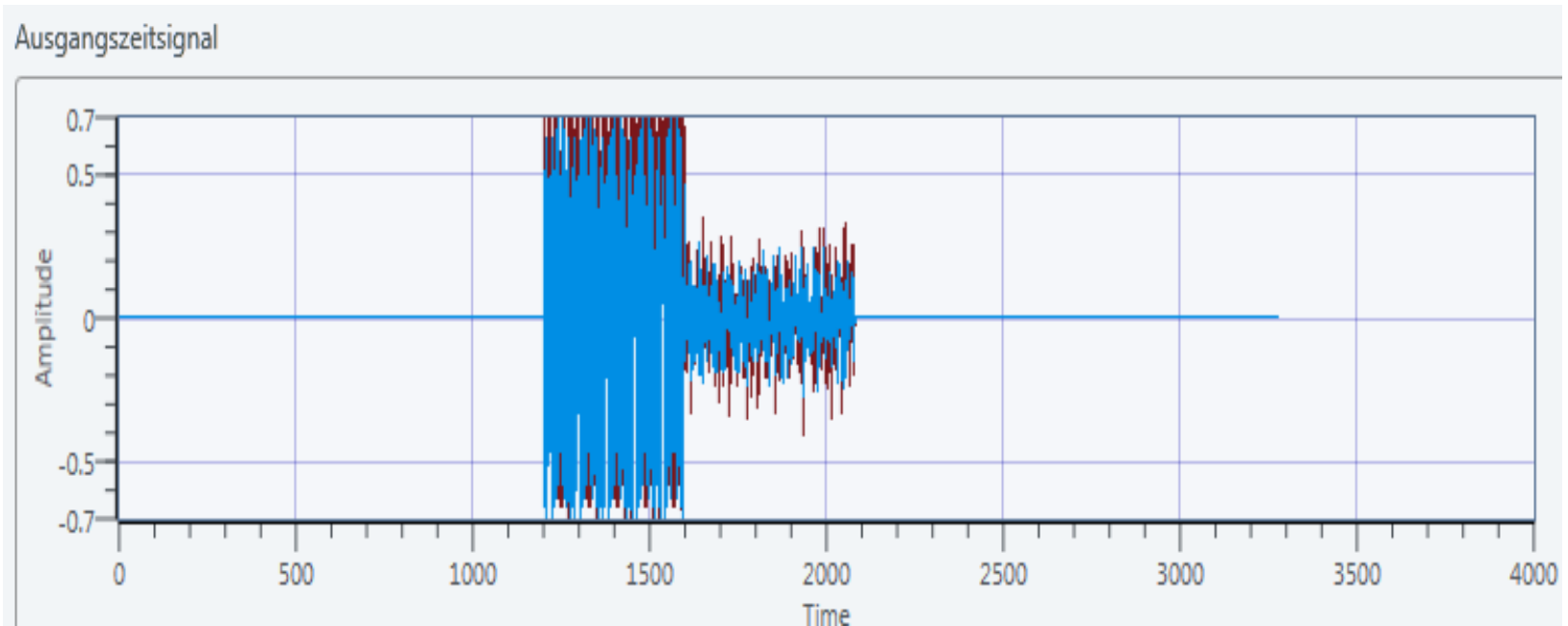
Settings Transceiver 1

active antenna  
TX/RX  
ip  
192.168.20.2

Common Transceiver Settings

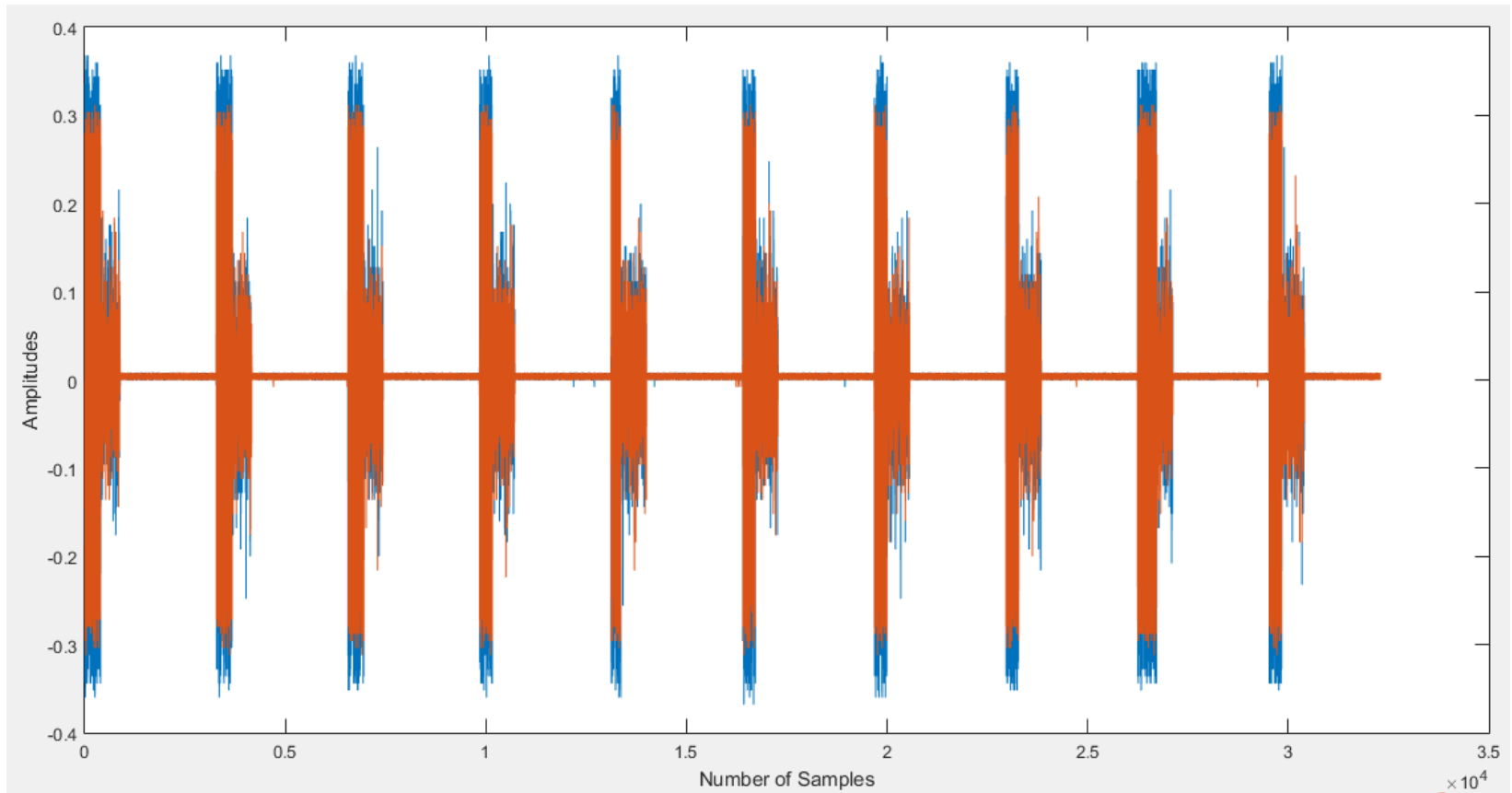
sampling rate [1/s] 1M	carrier frequency tx [Hz] 800M	carrier frequency rx [Hz] 600M	
gain tx [dB] 20	gain rx [dB] 10	initialisation waiting time whole seconds 0 fractional seconds 0.5	triggering Interval whole seconds 2 fractional seconds 0
no. of rx samples 3280	timeout [s] 10		
number of splits 1			

# Program test



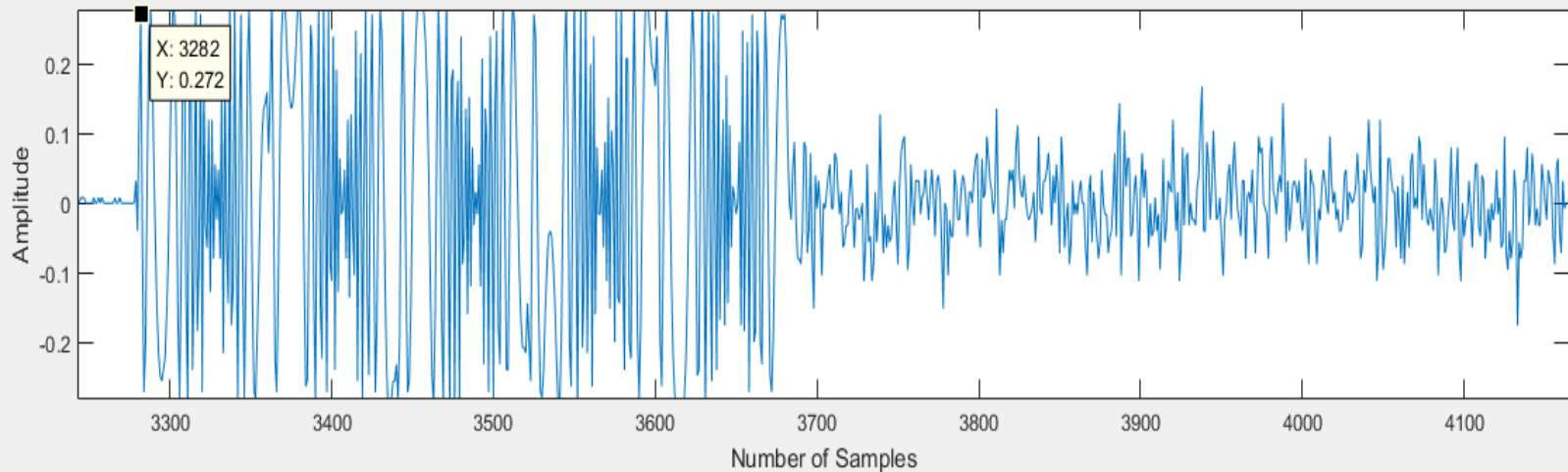
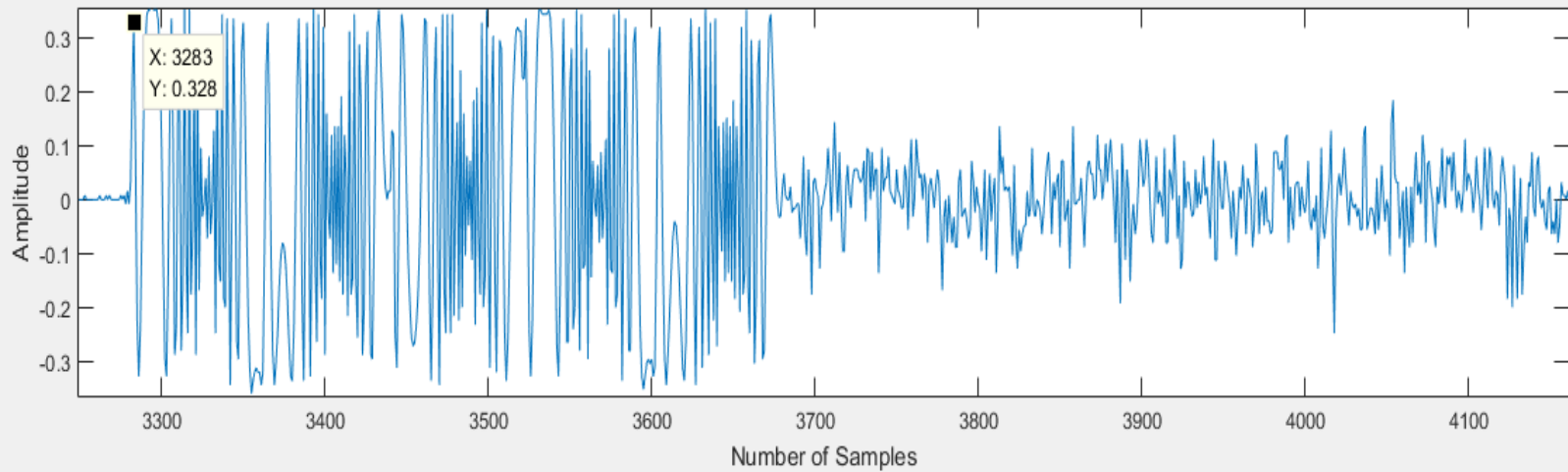


# Test result in C



Test result of C program

# Test result in C

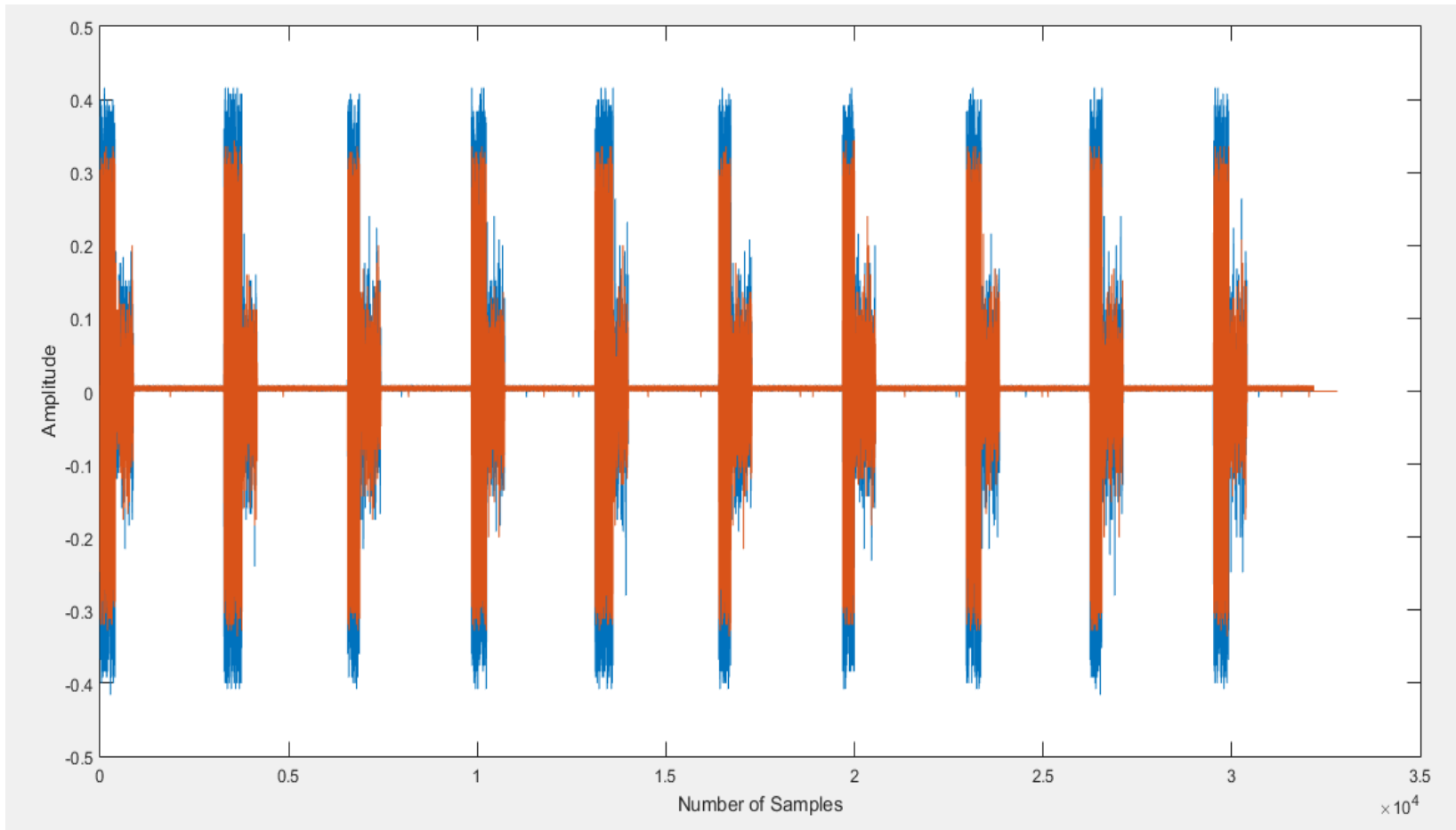


# Test result in MATLAB

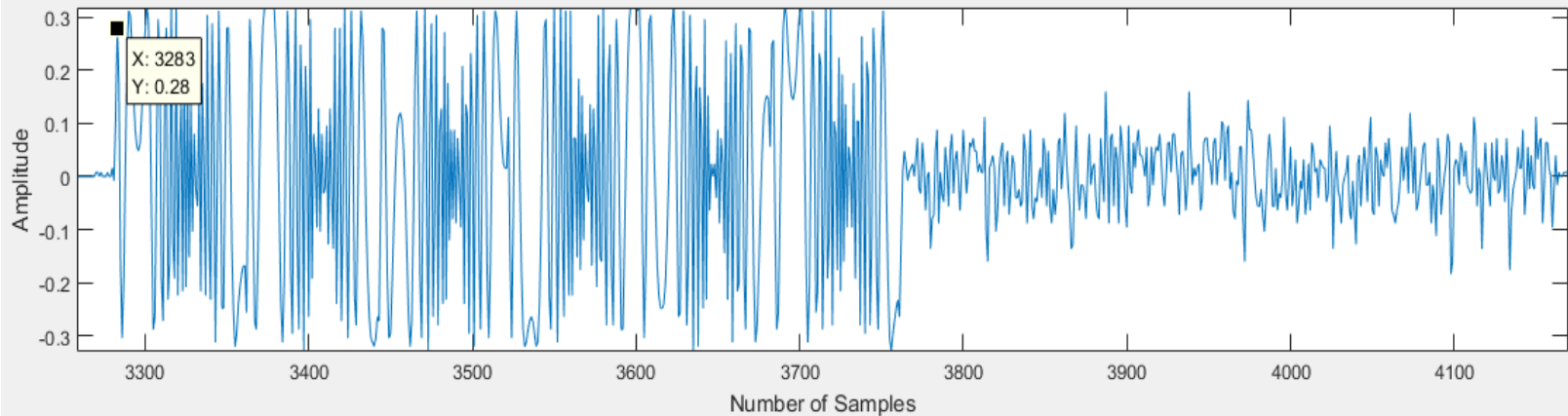
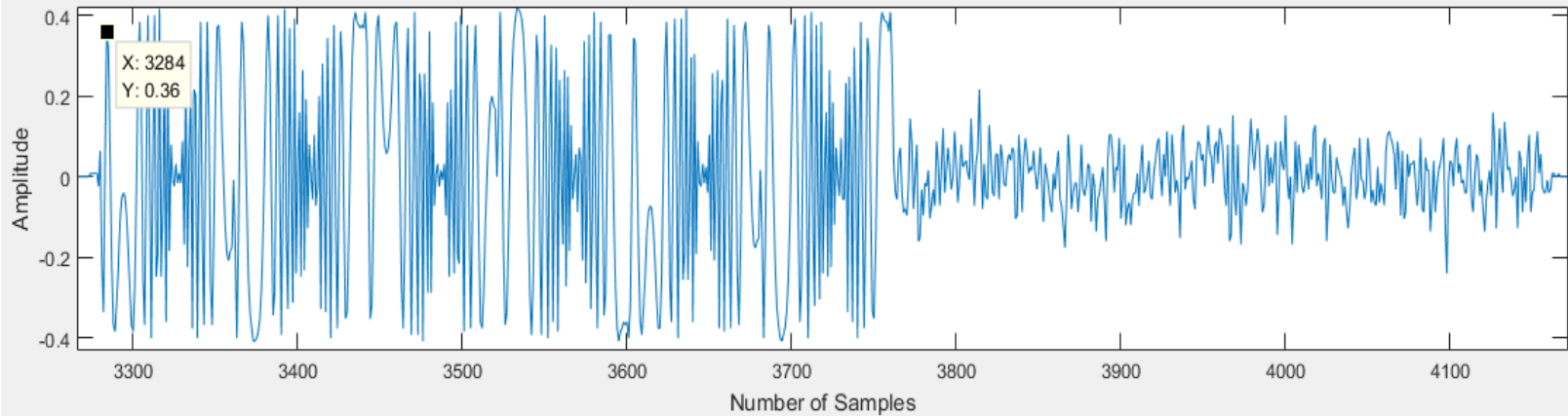
```
>> [A,B] = masterthesis('masterthesis','-b','3280','-n','32800','-s','1','-g','800','20','file1','800','20','file2');  
index: 0, freq: 800.000000, gainMode: 1, gain: 20, filename: file1  
index: 1, freq: 800.000000, gainMode: 1, gain: 20, filename: file2  
File created successfully!  
File created successfully!  
rtlsdr_get_device_count done: 2, opening 0  
rtlsdr_get_device_count done: 2, opening 1  
index 0: set gain to 20  
wait for signal  
index 1: set gain to 20  
wait for signal  
Device index 0 start at 11570094 ms since system boot.  
Device index 1 start at 11570094 ms since system boot.  
start_position = 1246  
start_position = 3094
```



# Test result in MATLAB



# Test result in MATLAB



# Conclusion

1. **Save** the data read in different loops into a whole memory
2. Find the **start point** of the data streams separately but in parallel during the data read in the first loop
3. **Shift** the two data streams and save into a file or into a matrix



# Thank you for your interest!

