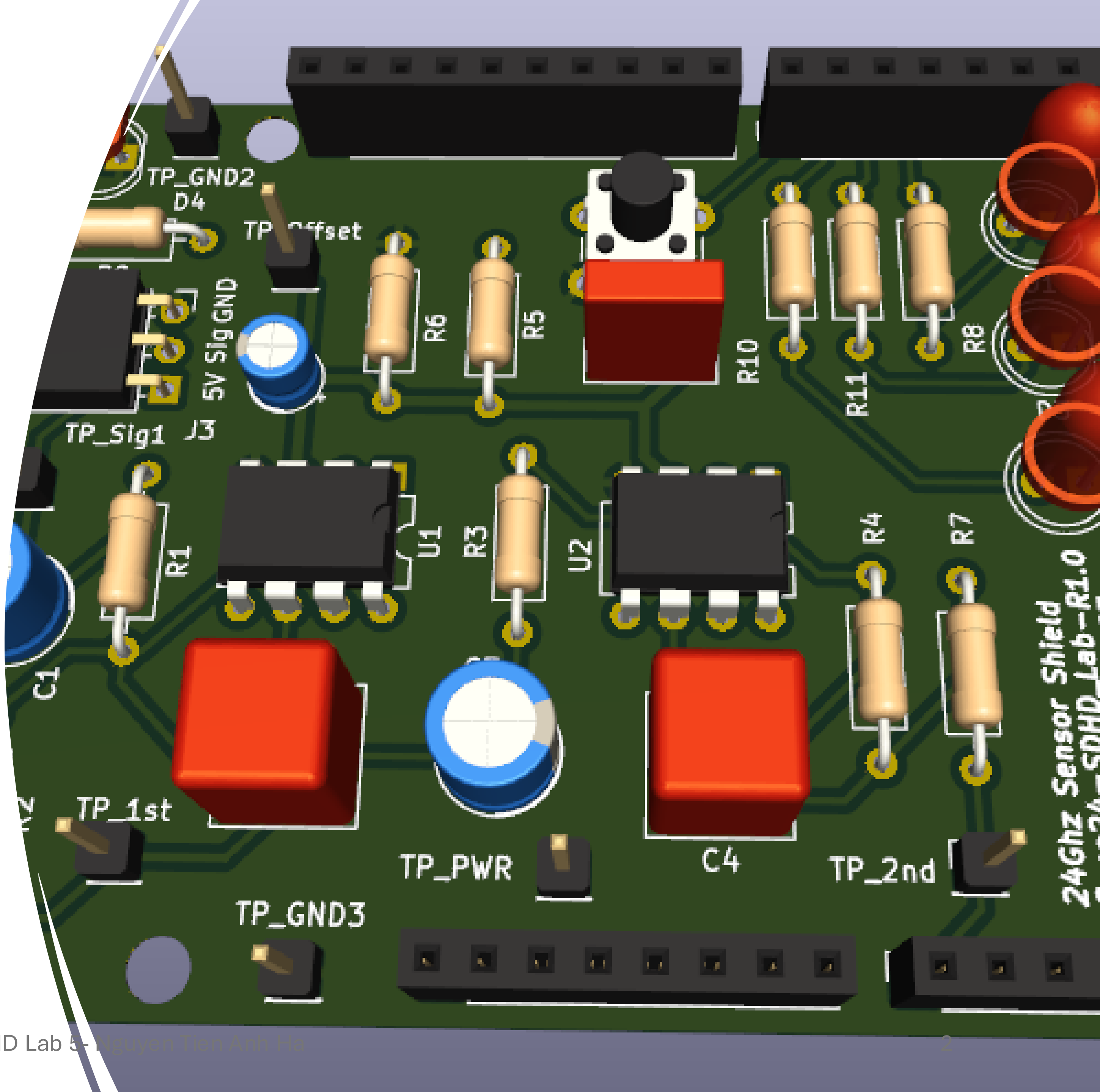
24Ghz Movement Detection

Signal Hardware Driven Design

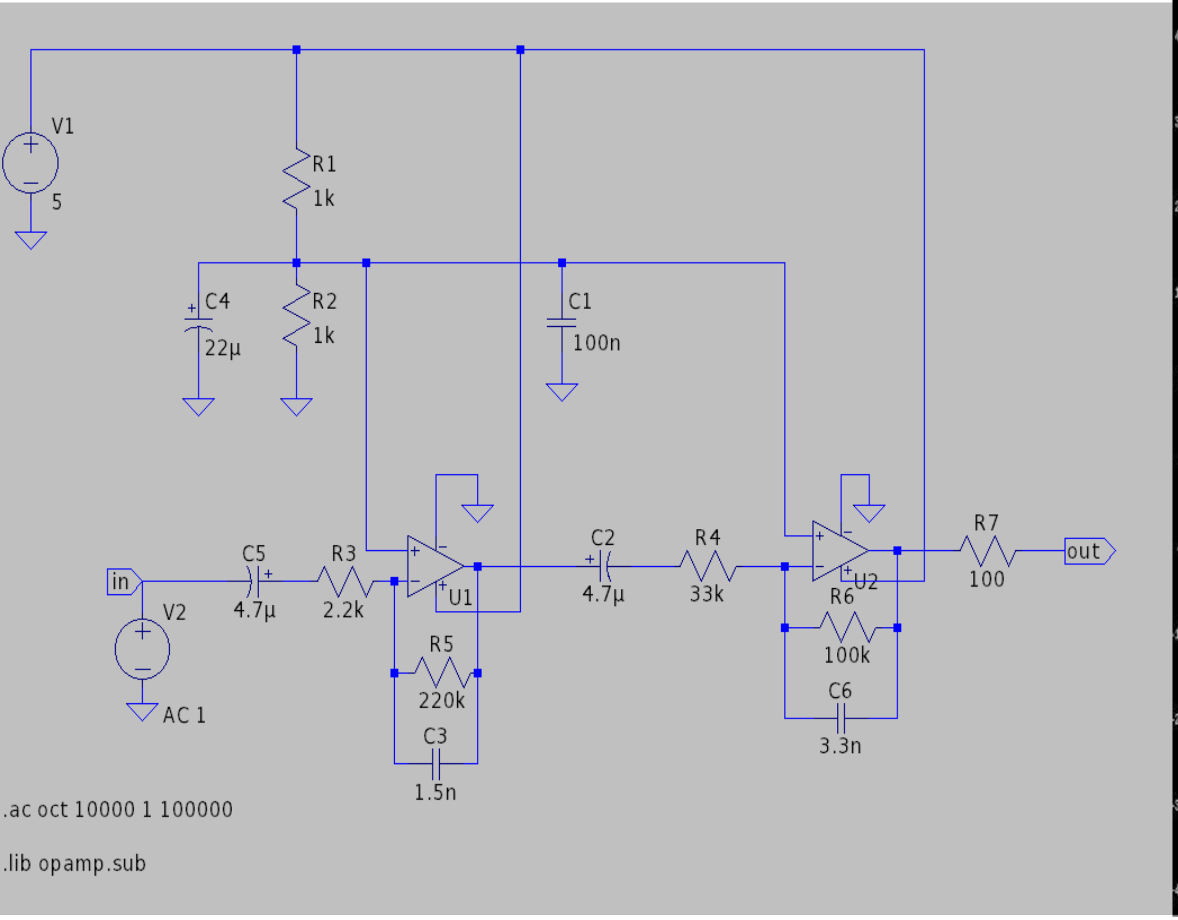
Nguyen Tien Anh Ha - 1127956

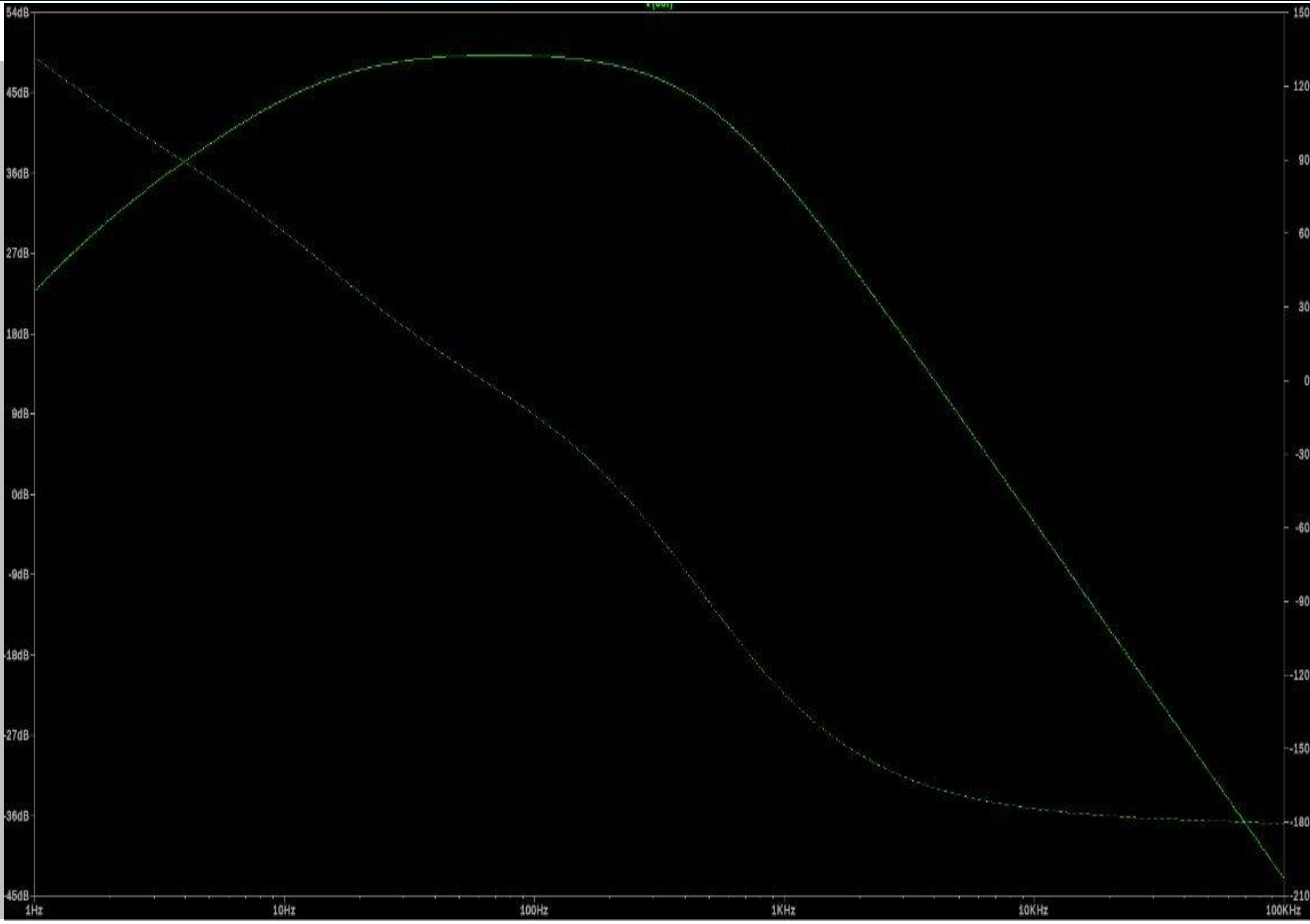
Table of content

- Hardware
- DFT Scaling
- Hanning Window
- CFAR Implementation
- Final Result

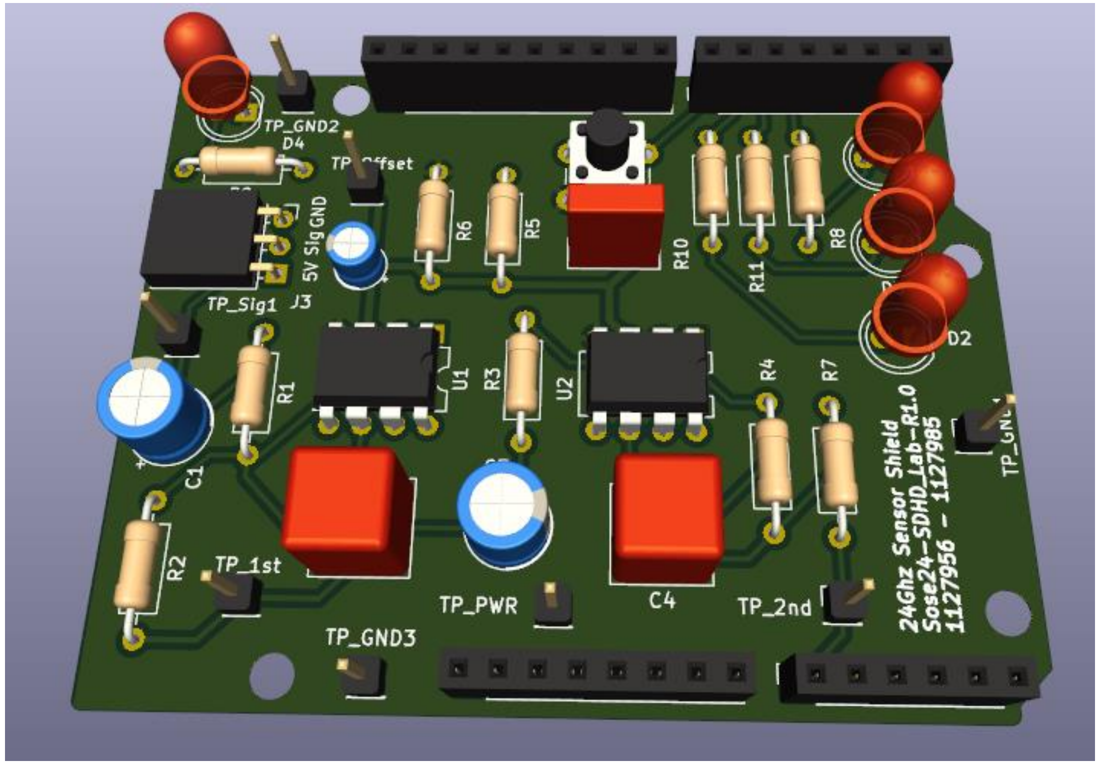


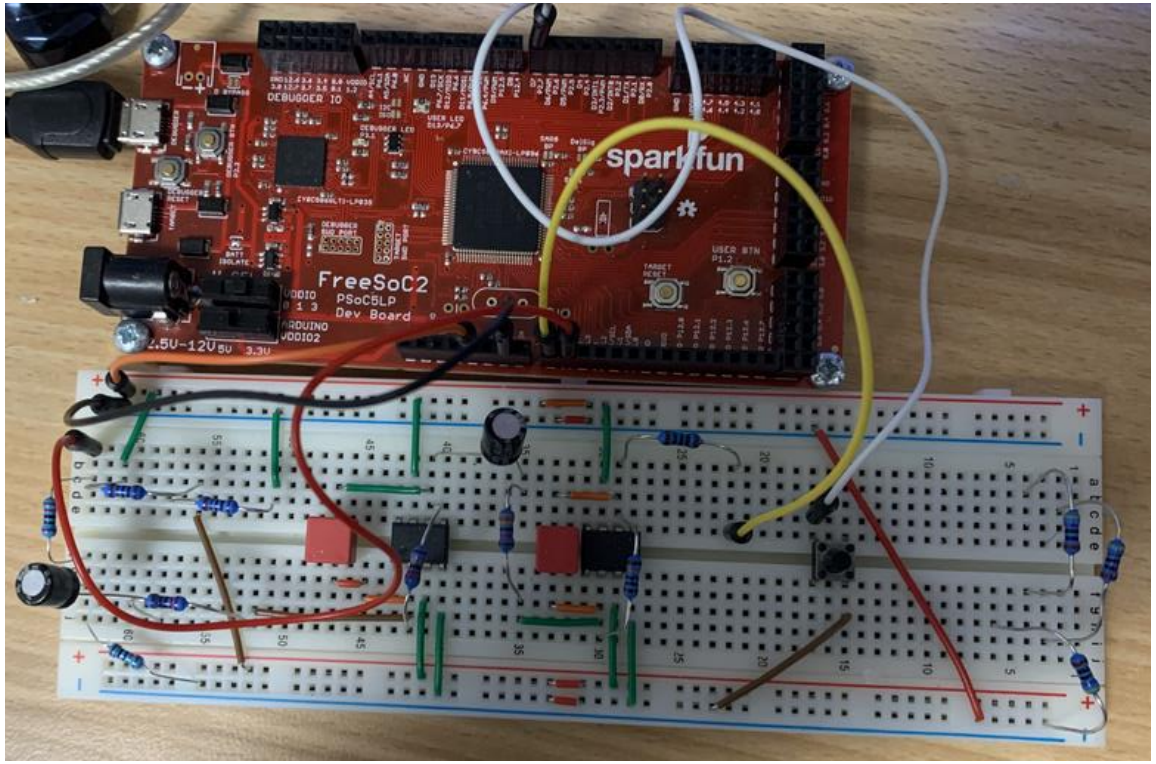
Simulation


V1
5
R1
1k
C4
22µ
R2
1k
C1
100n
in
V2
AC 1
C5
4.7µ
R3
2.2k
U1
R5
220k
C3
1.5n
C2
4.7µ
R4
33k
U2
R6
100k
C6
3.3n
R7
100
out
.ac oct 10000 1 100000
.lib opamp.sub

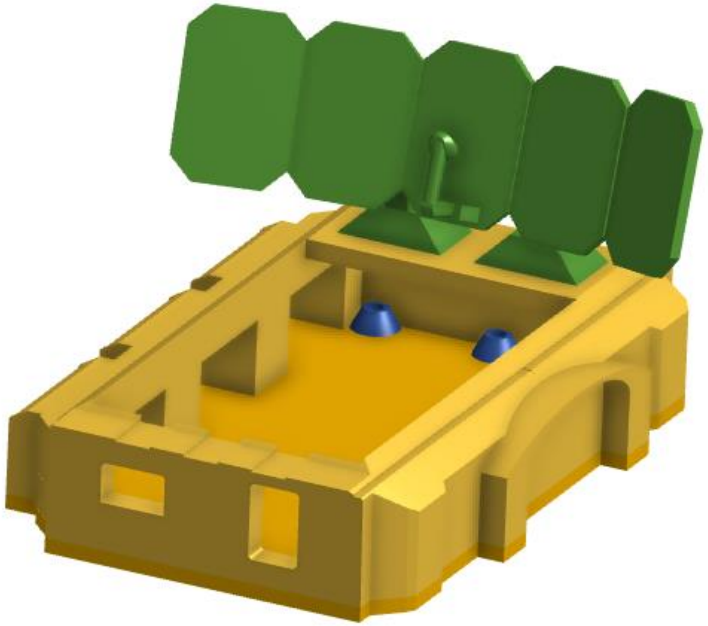

54dB
45dB
36dB
27dB
18dB
9dB
0dB
-9dB
-18dB
-27dB
-36dB
-45dB
1Hz
10Hz
100Hz
1KHz
10KHz
100KHz

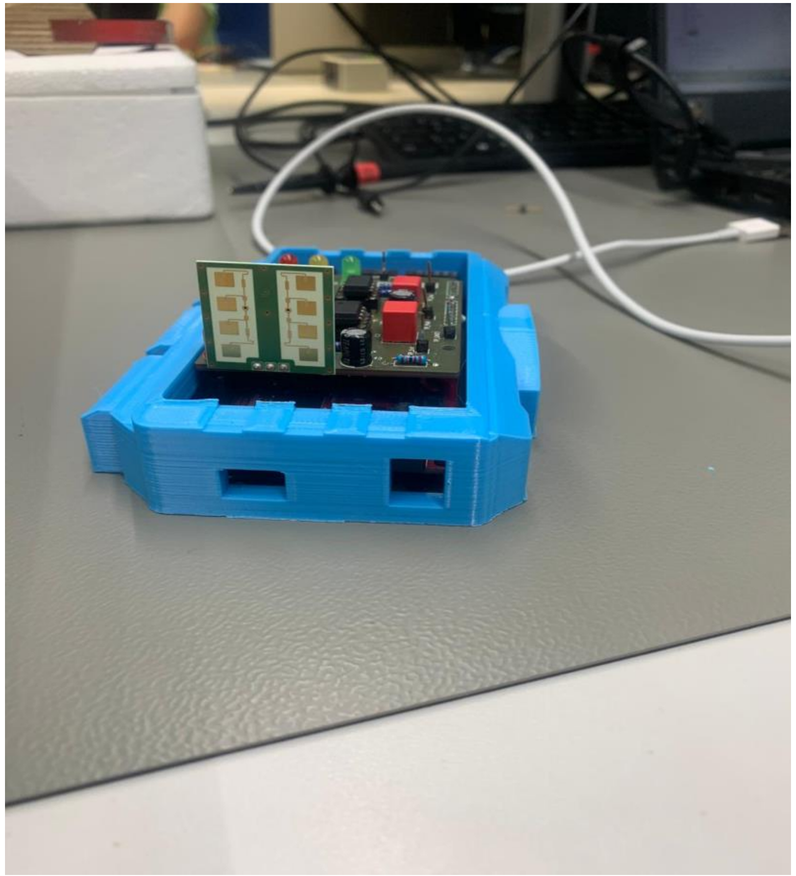
Hardware


24Ghz Sensor Shield
Sose24-SDHD_Lab-R1.0
1127956 - 1127985
TP_GND2
TP_Sig1
TP_1st
TP_PWR
TP_2nd
TP_GND3

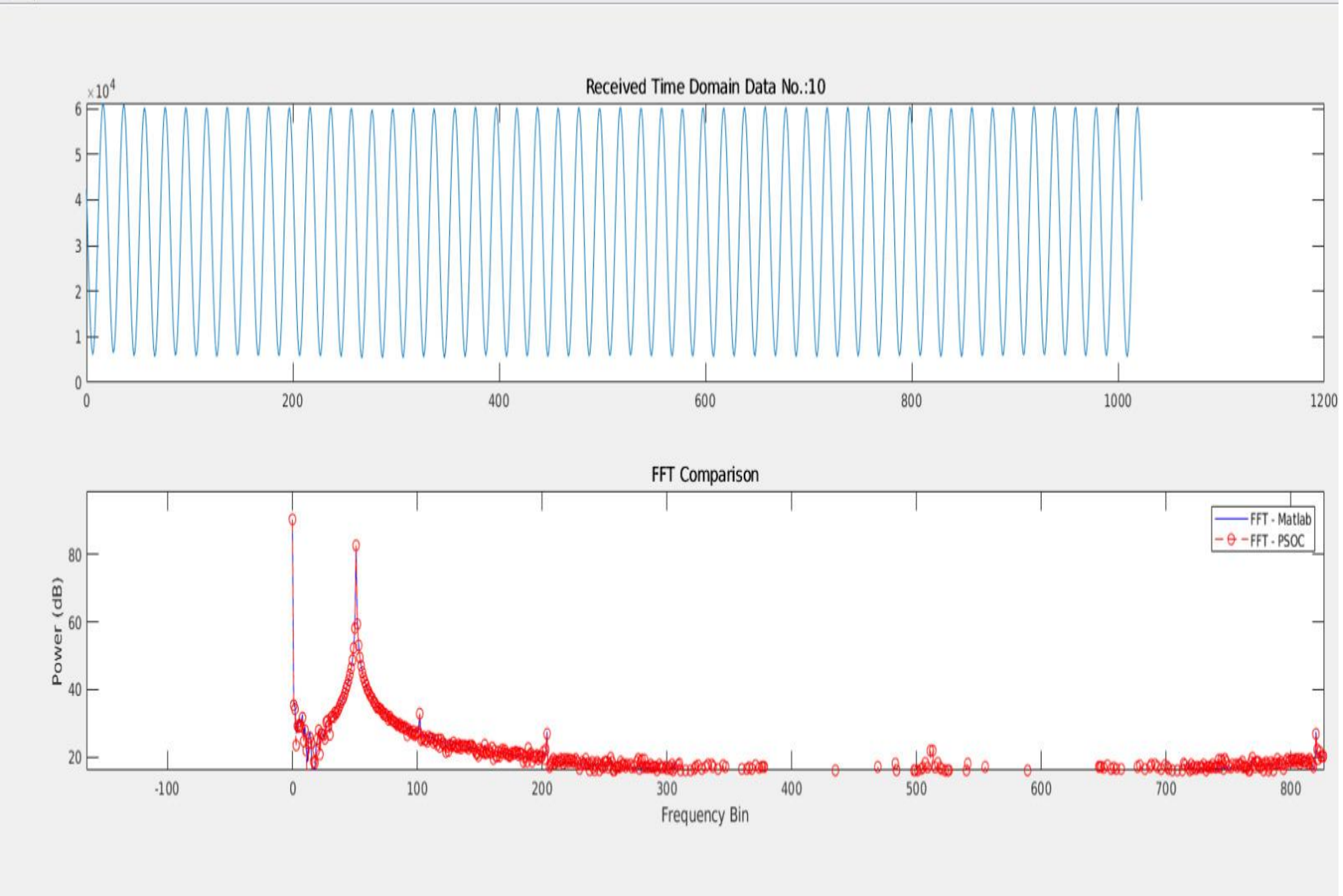

sparkfun
FreeSoC2
PSoC5LP
Dev Board

PCB and Enclosure





DFT Scaling Verification



- FFT libraries downsample by N samples by default
- Divided the computed MATLAB amplitude by N to match the power spectrum from PSOC

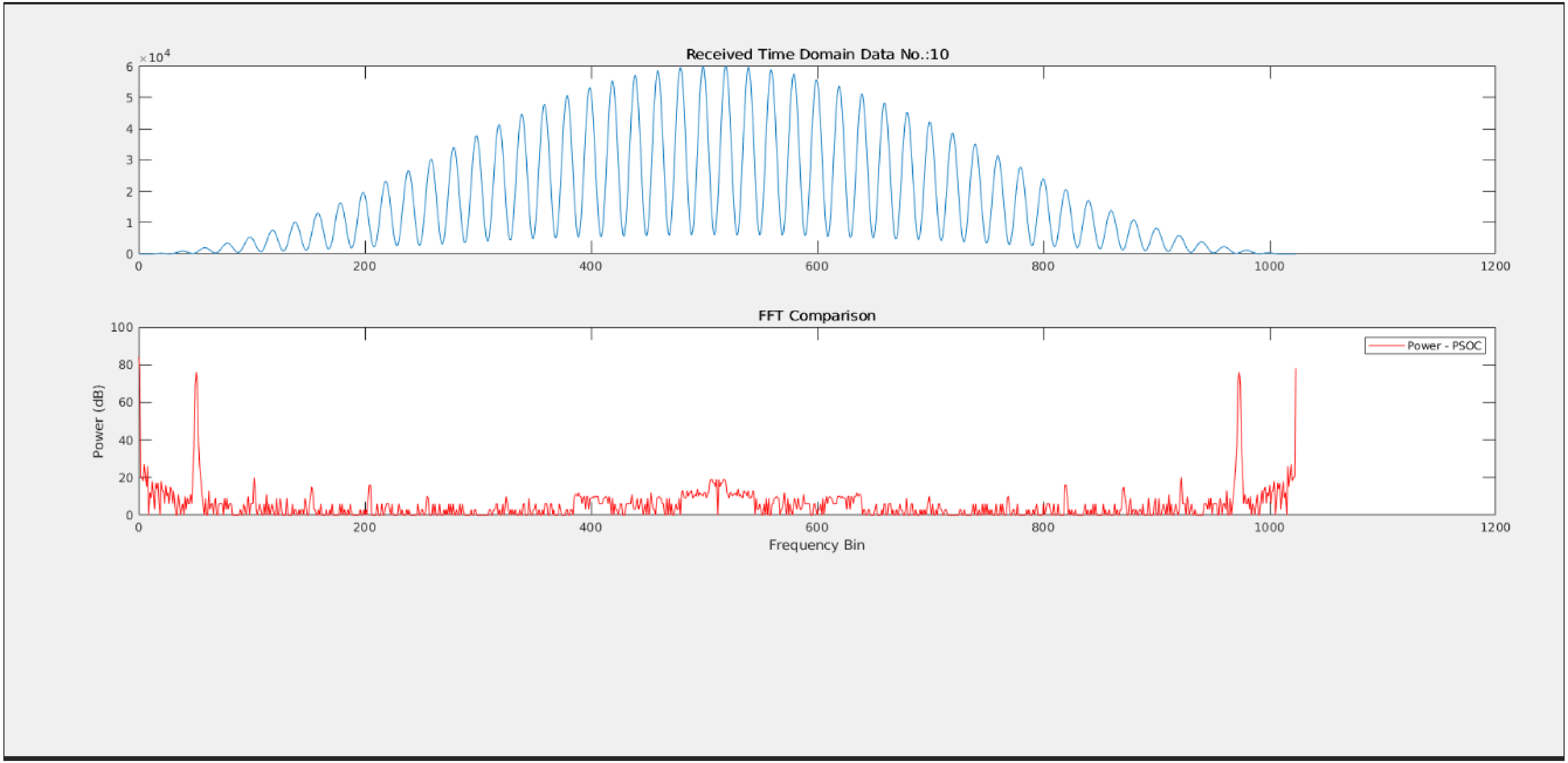
Hanning Window

```c
#define MAX_LENGTH 1024

static const double hanning_table[MAX_LENGTH] = {
    0.000000, 0.000009, 0.000038, 0.000085, 0.000151, 0.000236, 0.000339, 0.000462,
    0.000603, 0.000764, 0.000943, 0.001141, 0.001357, 0.001593, 0.001847, 0.002120,
    0.002412, 0.002723, 0.003052, 0.003401, 0.003768, 0.004153, 0.004558, 0.004981,
    0.005422, 0.005883, 0.006362, 0.006859, 0.007376, 0.007910, 0.008464, 0.009036,
    0.009626, 0.010235, 0.010862, 0.011508, 0.012173, 0.012855, 0.013556, 0.014276,
    0.015014, 0.015770, 0.016544, 0.017336, 0.018147, 0.018976, 0.019823, 0.020688,
    0.021572, 0.022473, 0.023392, 0.024330, 0.025285, 0.026258, 0.027249, 0.028258,
    0.029285, 0.030329, 0.031391, 0.032471, 0.033568, 0.034683, 0.035816, 0.036966,
    0.038134, 0.039319, 0.040521, 0.041741, 0.042978, 0.044232, 0.045503, 0.046792,
    0.048098, 0.049420, 0.050760, 0.052117, 0.053490, 0.054881, 0.056288, 0.057712,
    0.059153, 0.060610, 0.062084, 0.063574, 0.065081, 0.066604, 0.068143, 0.069699,
    0.071271, 0.072860, 0.074464, 0.076084, 0.077721, 0.079373, 0.081041, 0.082725,
    0.084425, 0.086141, 0.087872, 0.089618, 0.091380, 0.093158, 0.094951, 0.096759,
    0.098582, 0.100421, 0.102274, 0.104143, 0.106026, 0.107924, 0.109838, 0.111765,
    0.113708, 0.115665, 0.117637, 0.119623, 0.121623, 0.123638, 0.125666, 0.127709,
    0.129766, 0.131837, 0.133922, 0.136021, 0.138133, 0.140259, 0.142399, 0.144552,
    0.146718, 0.148898, 0.151091, 0.153297, 0.155517, 0.157749, 0.159994, 0.162252,
    0.164523, 0.166806, 0.169102, 0.171411, 0.173732, 0.176065, 0.178410, 0.180768,
    0.183137, 0.185519, 0.187912, 0.190317, 0.192734, 0.195163, 0.197603, 0.200054,
    0.202517, 0.204990, 0.207476, 0.209972, 0.212479, 0.214996, 0.217525, 0.220064,
    0.222614, 0.225174, 0.227745, 0.230326, 0.232917, 0.235518, 0.238129, 0.240750,
    0.243381, 0.246021, 0.248671, 0.251331, 0.254000, 0.256678, 0.259365, 0.262062,
    0.264767, 0.267482, 0.270205, 0.272936, 0.275677, 0.278425, 0.281183, 0.283948,
    0.286721, 0.289503, 0.292292, 0.295090, 0.297895, 0.300708, 0.303528, 0.306355,
    0.309190, 0.312033, 0.314882, 0.317738, 0.320601, 0.323471, 0.326347, 0.329230,
    0.332120, 0.335016, 0.337918, 0.340826, 0.343740, 0.346660, 0.349586, 0.352518,
    0.355455, 0.358397, 0.361345, 0.364298, 0.367256, 0.370220, 0.373188, 0.376161,
    0.379138, 0.382121, 0.385107, 0.388098, 0.391093, 0.394092, 0.397096, 0.400103,
```

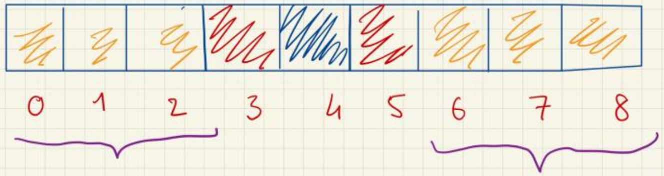
Advantages:

- Reduce spectral leakage and still provide good frequency resolution
- Easy to apply and fast to calculate.

$$w(n) = 0.5 - 0.5\cos\left(\frac{2\pi n}{M-1}\right) \qquad 0 \le n \le M-1$$


Received Time Domain Data No.:10
×10^4
0
1
2
3
4
5
6
0
200
400
600
800
1000
1200
FFT Comparison
Power (dB)
0
20
40
60
80
100
Power - PSOC
Frequency Bin

CA- CFAR Parameter



Trailing cell [0:2]

Leading cell [6:8]

Guard : 1

Reference : 3

Number of guard: 28

Number of reference: 30

Probability of false alarm: 10^{-6}

$$\hat{\sigma}^2_w = \frac{1}{N}\sum_{k=1}^{N} x_{ref}$$

$$x_{ref}(k) = |DFT(k)|^2$$

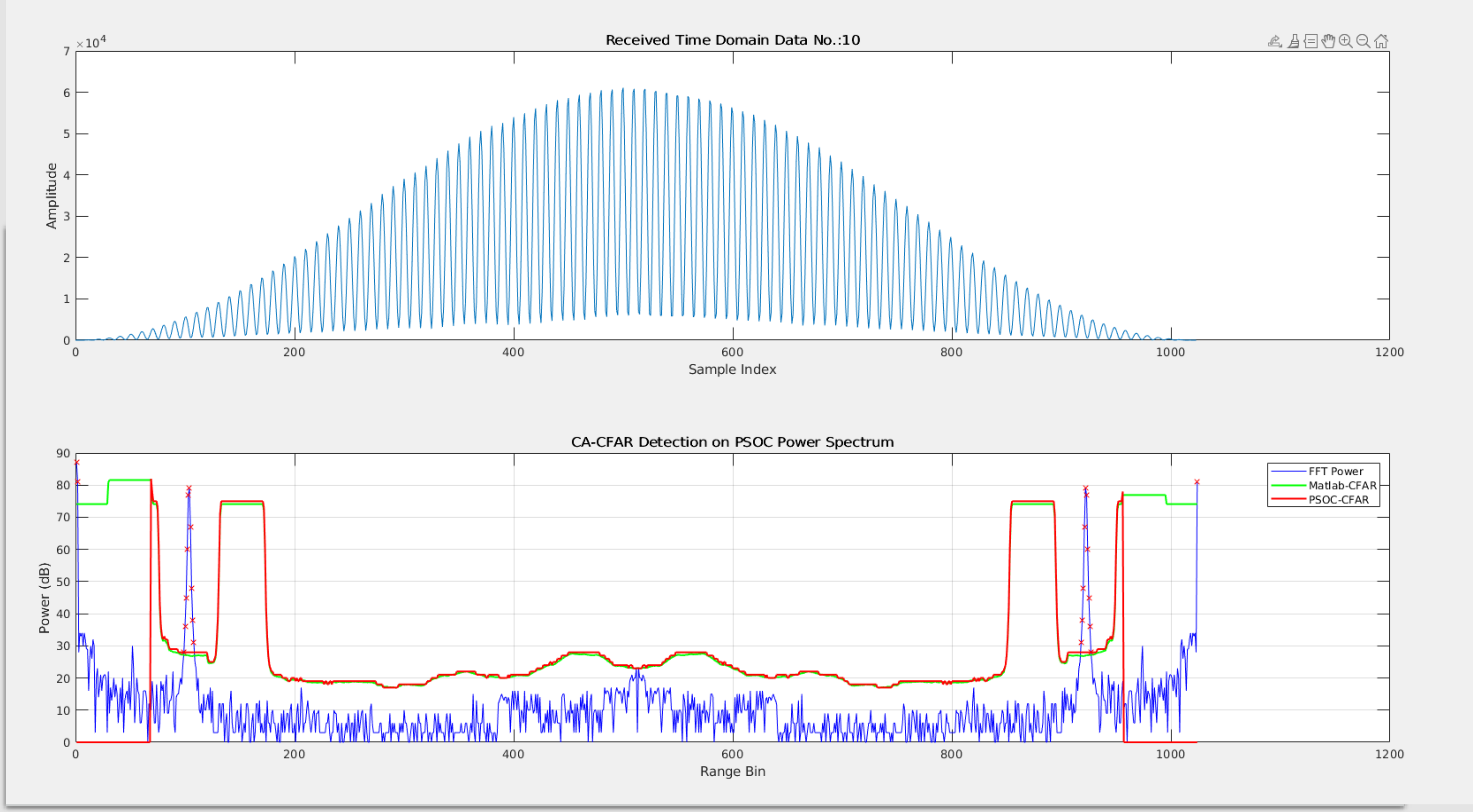
Required threshold:

$$\hat{T} = \alpha \hat{\sigma}^2_w$$

$$\alpha = N(P_{FA}^{-1/N} - 1)$$

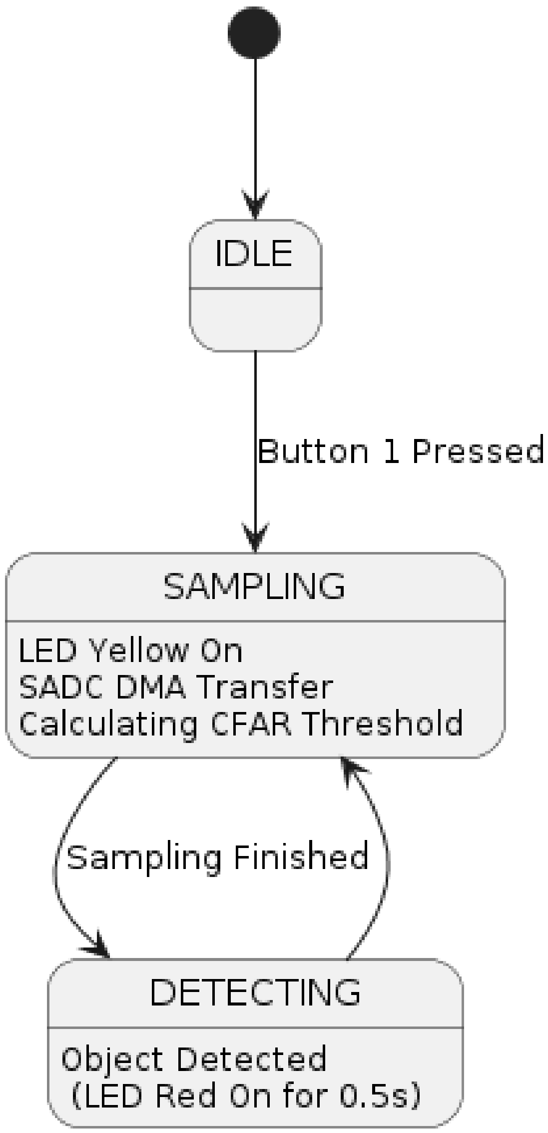
CA – CFAR Algorithm

```
if(CUT >= *first_CUT_Index && CUT <= *last_CUT_index){

    float64_t sumLeading = 0.0;
    float64_t sumTrailing = 0.0;
    // Calculate sum of leading and trailing window
    for (uint8_t i =0; i < NR; ++i){
        sumLeading += fftInput[CUT - NG - NR + i];
        sumTrailing += fftInput[CUT + NG + i +1];
    }

    float64_t noiseLevel = (sumLeading + sumTrailing) / (2* NR);


    float64_t threshold = alpha * noiseLevel;

    CAFRThreshold[CUT] = (sint32_t)(10*log10(threshold));


}
```



Received Time Domain Data No.:10
Amplitude
Sample Index
CA-CFAR Detection on PSOC Power Spectrum
Power (dB)
Range Bin
FFT Power
Matlab-CFAR
PSOC-CFAR

Updated state machine



Demonstration

Thank you for listening