

Lab Report

|  |  |
| --- | --- |
| Name: Alexis Steven Garcia | Date: 10/22/2018 |
| Course: EGCP-450 | Lab #: 5 |

Grading Criteria:

|  |  |  |
| --- | --- | --- |
| **Section** | **Earned Points** | **Possible Points** |
| Problem/Objective: |  | 10 |
| Background: |  | 15 |
| Questions/Deliverables: |  | 15 |
| Program Code: |  | 30 |
| Demo: |  | 30 |
| Total: | 0 | 100 |

**PLEASE UPLOAD YOUR REPORT IN TITANIUM. NO PAPER REPORTS.**

Professor Comments:

# Problem/Objective

State the problem statement and/or objective of the lab. This must be a complete paragraph (i.e., at least 5 sentences).

The objective of this lab was to become familiar with designing digital Finite Impulse Response (FIR) filters. The goal was to create a FIR filter using MATLAB’s filter designer tool and implement it on the MSP432. More specifically, the purpose of the lab was to create a lowpass filter that would filter out a frequency above a certain threshold. In other words, a low-pass filter is a filter that passes signals with a frequency lower than a certain cutoff frequency and attenuates with frequencies higher than the cutoff frequency. Also, despite the example we are implementing not being a real-time system, the intention of this lab is to gain experience in coding a digital filter and analyzing the output.

# Background

Briefly describe what you did in the lab including technical detail. It must be at least two **complete** paragraphs to receive full credit.

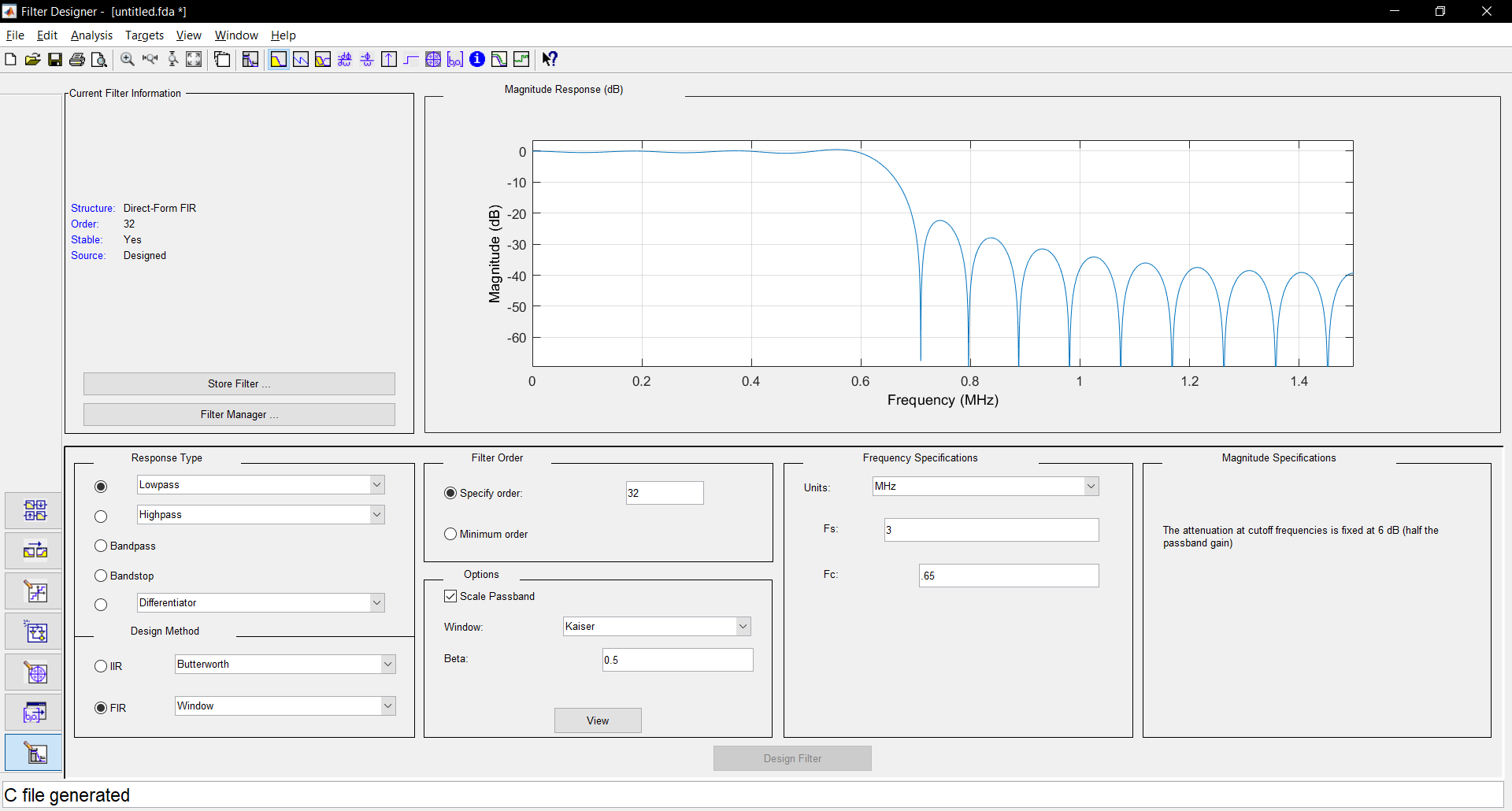
This lab assisted in understanding how to design a digital Finite Impulse Response (FIR) filter using MATLAB’s filter designer tool. Our lowpass filter was created with the idea of filtering out a frequency lower than a certain cutoff frequency and attenuating frequencies higher than the cutoff. In other words, the specifications for our filter were designed in a way that would remove a 1MHz sinewave from the signal and leave a 375kHz sinewave. For this to be accomplished, we should design a filter with a lowpass response type and a windowed design. Next, we set the filter order to 32 to increase our maximum delay. In other words, we increase N in the equation . Additionally, we specify our sampling frequency () to be the same frequency as the MSP432 default bus clock. Finally, with a sampling frequency of 3MHz, we set the cutoff frequency to be between 375kHz and 1MHz to filter out the 1MHz signal.

With our filter designed, the next steps are to generate samples for 375 kHz and 1 MHz sinusoids using MATLAB. To do this, we first obtain the sampling rate needed for the desired sinusoid. This is done by dividing the sampling frequency by the frequency of the desired sinusoid. Then we make a loop from 0 to , where N is the sampling rate, and take the angle every interval. After, we plug in each angle in the equation to receive our samples for the specified sinusoids.

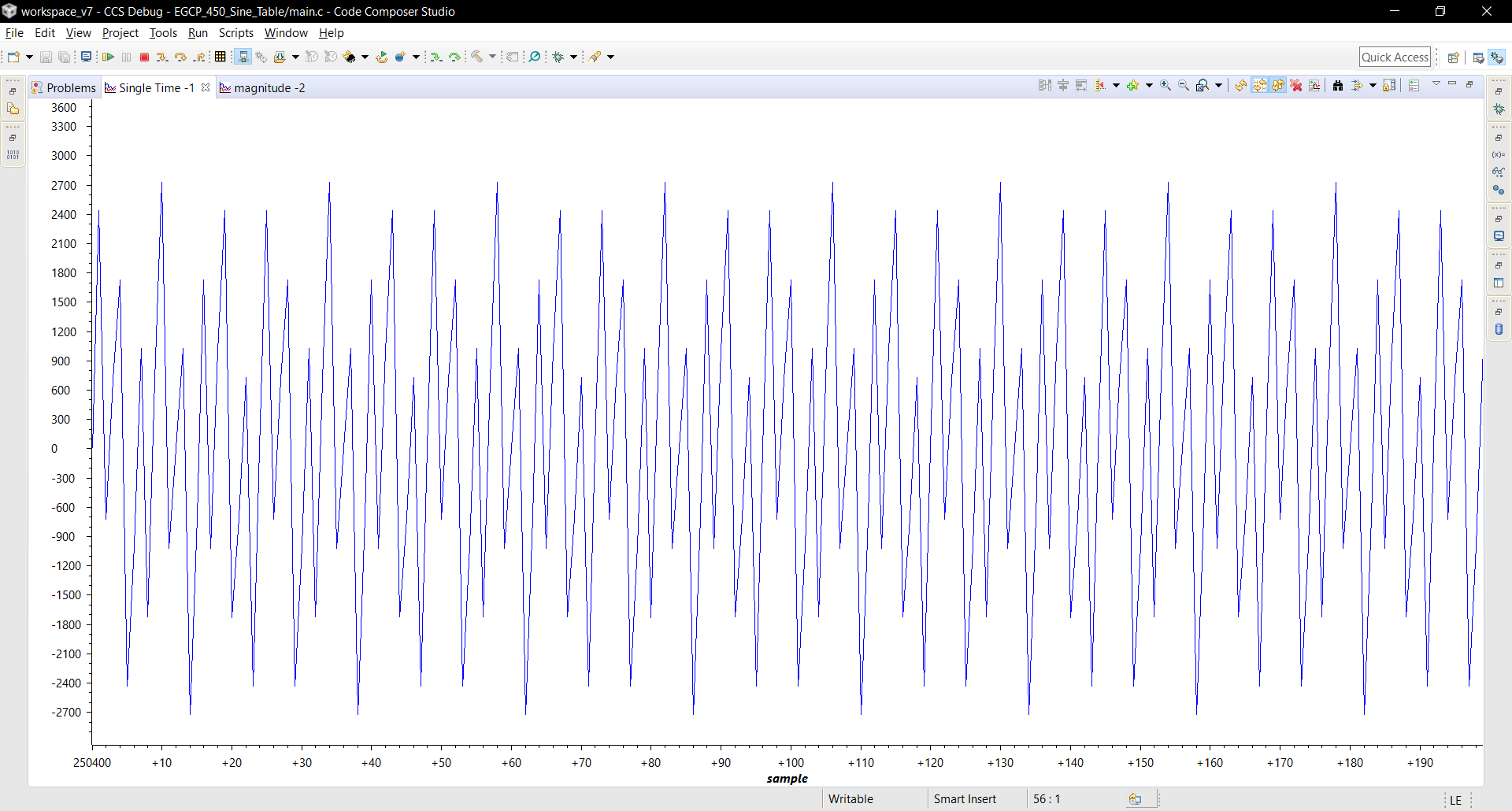
The final step is to use CCS to create an embedded program to filter a signal. We begin by exporting the coefficients created from the filter into a header file that can be accessed by our main program. Then, we add the samples received for the 375kHz and 1MHz sinewaves to two different arrays. Afterward, within our main program code, we create a loop to fill a buffer of 200 samples where each sample is the sum of the 375 kHz and 1 MHz sinusoids, but the gain of the 1MHz sinewave is twice the amplitude of the 375kHz sinewave. Finally, we create a function that takes a pointer to the buffer and the number of samples. This function will traverse through the buffer of samples, implement the convolution equation with N coefficients and N delay samples, and update delay samples to be used for calculating y(n) at y(n+1).

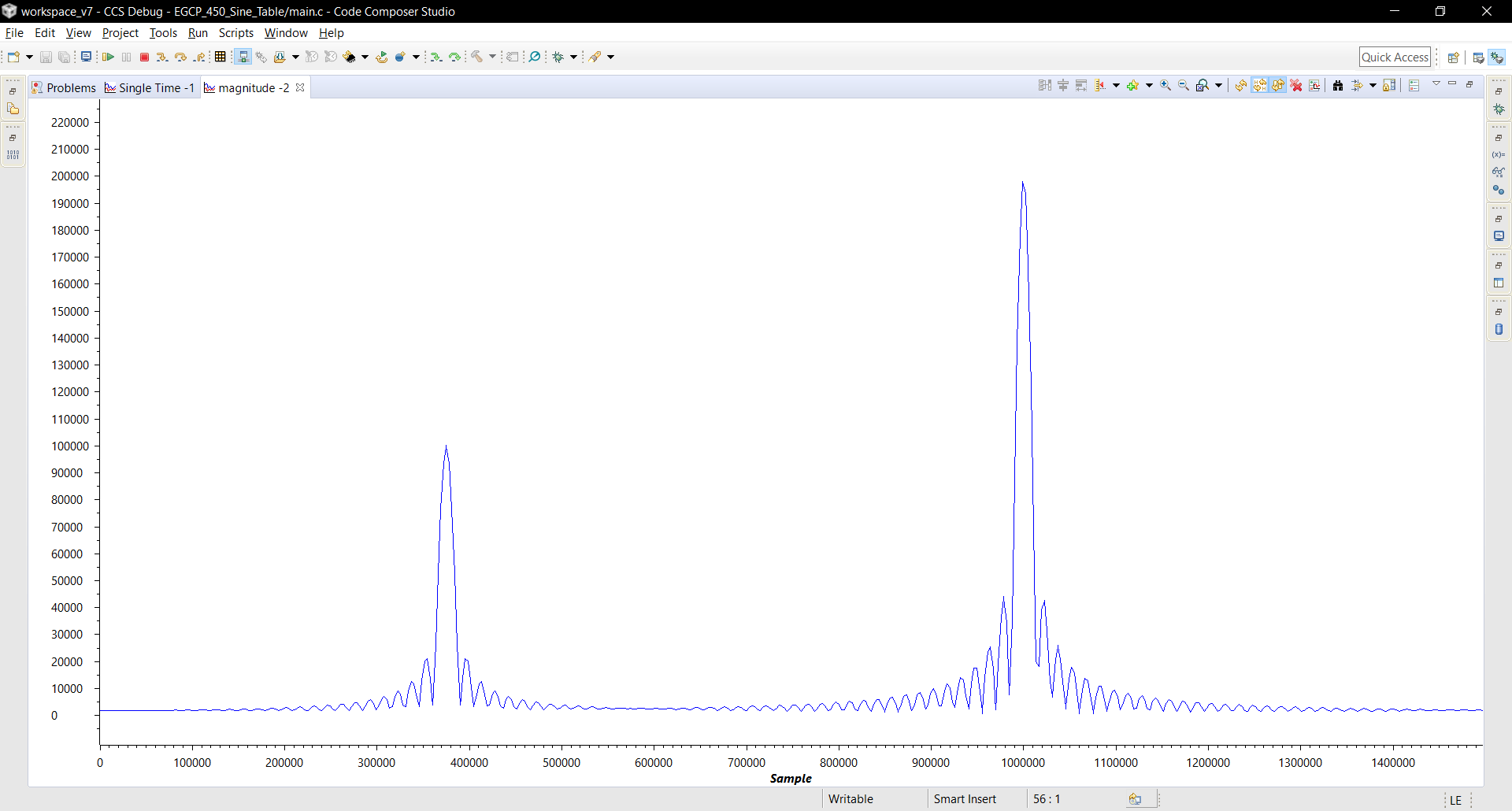
# Questions/Deliverables

1. Screenshot of FDATool with the parameters settings. That is, the FDATool GUI has the all the settings (e.g., sampling rate, response type, method, cutoff frequency, etc.) entered and are not left blank.

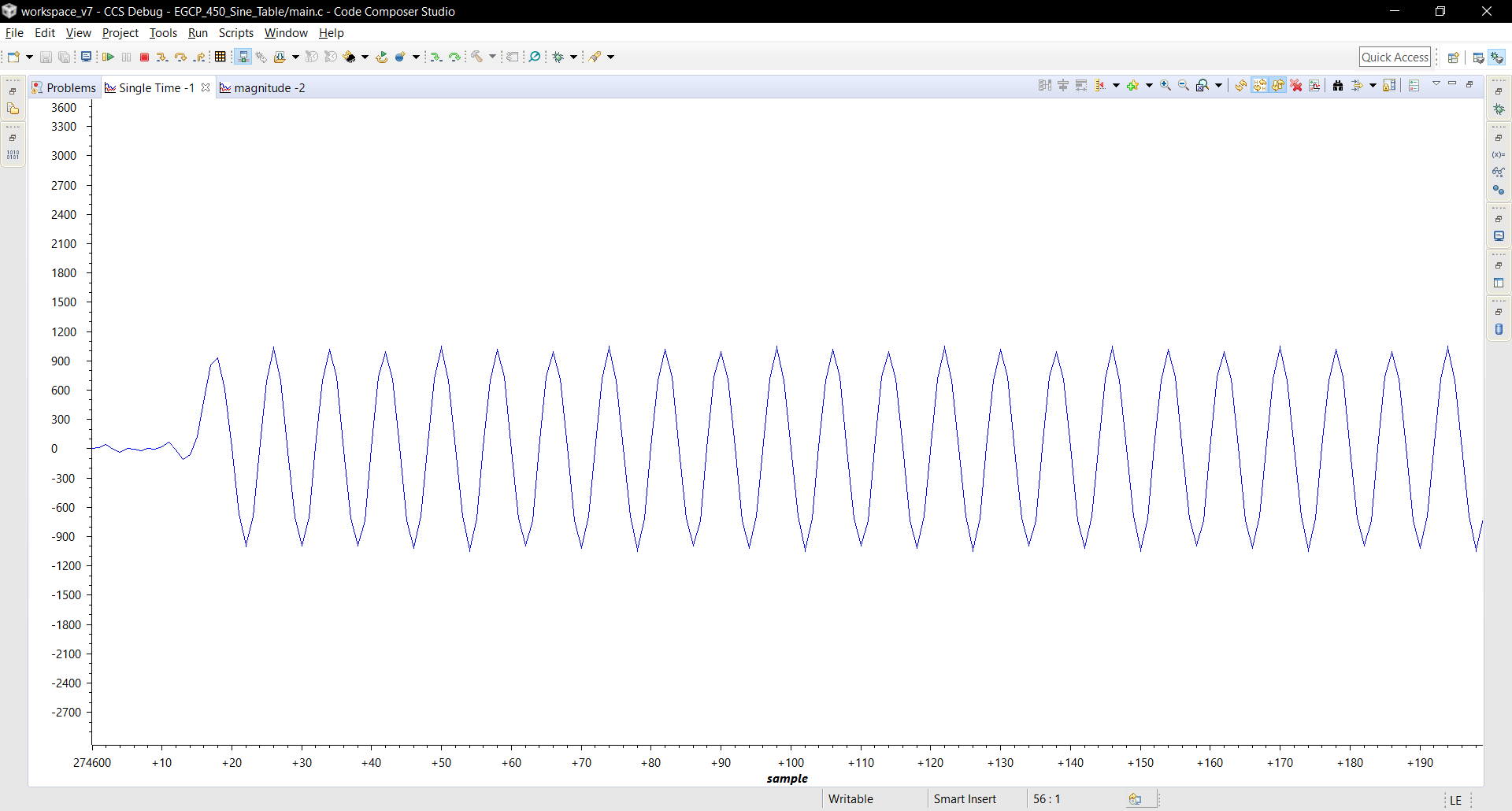


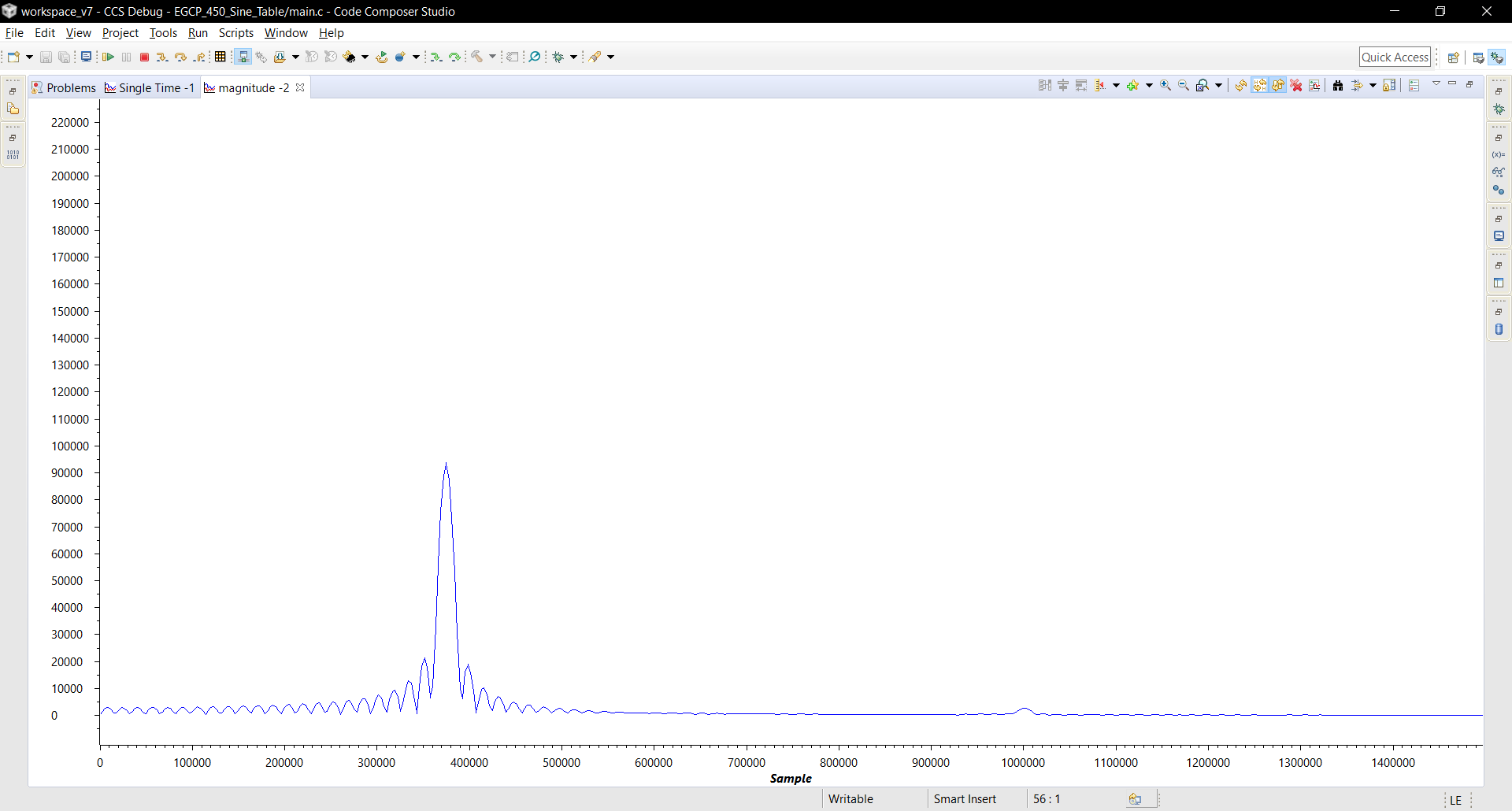
1. Screenshots of the time and magnitude plots at the first breakpoint (pre-filter).





1. Screenshots of the time and magnitude plots at the second breakpoint (post-filter).





# Program Code

Copy your code here. Please provide comments in your code. This will help me analyze your code and remove any ambiguity. **Provide your code as text, not as a screenshot/image**.

**Main.c:**

**#include** "fir.cof"

// Sine values f=375kHz @ fs=3MHZ

**short** sine\_table\_ind = 0;

**short** sine\_table\_gain = 1;

**short** sine\_table[8] = {0,707,1000,707,0,-707,-1000,-707};

// Sine values f=1MHz @ fs=3MHZ

**short** sine\_table\_ind2 = 0;

**short** sine\_table\_gain\_1M = 2;

**short** sine\_table\_1M[3] = {0,866,-866};

// Buffer

**short** buffer\_ind = 0;

**short** buffer[200];

//Filter params

**int** yn;

**short** dly[N];

**void** fir\_filt(**short**\* buffer\_ptr, **short** buffer\_n){

**short** ind1, ind2;

**for**(ind1=0; ind1<N; ind1++){

dly[ind1] = 0;

}

**for**(ind1=0; ind1<buffer\_n; ind1++){

dly[0] = \*(buffer\_ptr+ind1);

yn = 0;

**for**(ind2=0; ind2<N; ind2++){

yn += (h[ind2] \* dly[ind2]);

}

**for**(ind2=N-1; ind2>0; ind2--){

dly[ind2] = dly[ind2-1];

}

\*(buffer\_ptr+ind1) = yn >> 15;

}

}

**void** main(**void**){

**short** sine\_table\_n = **sizeof**(sine\_table)/**sizeof**(sine\_table[0]);

**short** sine\_table\_s = **sizeof**(sine\_table\_1M)/**sizeof**(sine\_table\_1M[0]);

**short** buffer\_n = **sizeof**(buffer)/**sizeof**(buffer[0]);

// Main Loop

**while**(buffer\_ind < buffer\_n){

buffer[buffer\_ind] = sine\_table[sine\_table\_ind]\*sine\_table\_gain +

sine\_table\_1M[sine\_table\_ind2]\*sine\_table\_gain\_1M;

buffer\_ind++;

**if** (++sine\_table\_ind2 > sine\_table\_s-1)

sine\_table\_ind2 = 0;

**if** (++sine\_table\_ind > sine\_table\_n-1)

sine\_table\_ind = 0;

}

fir\_filt(&buffer[0], buffer\_n);

**while**(1){}

}

**FIR.cof:**

#define N 33

short h[33] = {

124, 642, 144, -686, -481, 601, 860, -342, -1245,

-150, 1598, 1011, -1882, -2736, 2065, 9941, 13839, 9941,

2065, -2736, -1882, 1011, 1598, -150, -1245, -342, 860,

601, -481, -686, 144, 642, 124

};

**MATLAB Code:**

function EGCP450\_Lab5\_Main

fs = 3e6;

f = input('Frequency:\n');

N = round(fs/f);

ang = 0:360/N:360-(360/N);

x = 1000\*sin(deg2rad(ang));

y = int16(x);

BaseName = 'Lab\_4\_';

FileType = '.csv';

FileName=[BaseName,num2str(f),FileType];

dlmwrite(FileName, y, 'delimiter', ',');

csvread(FileName);

end