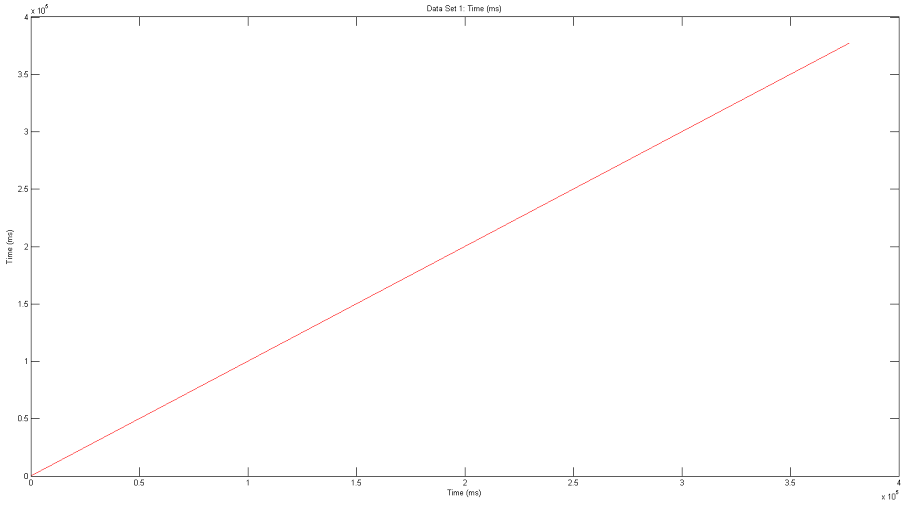
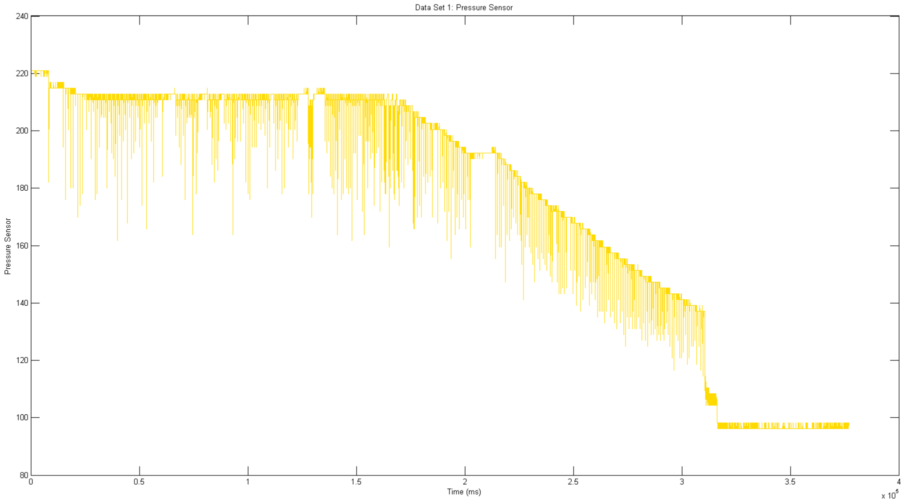
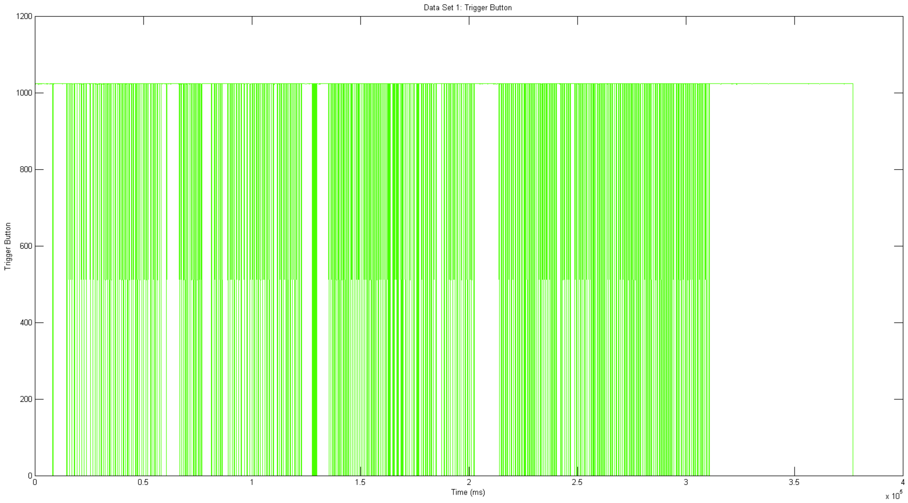
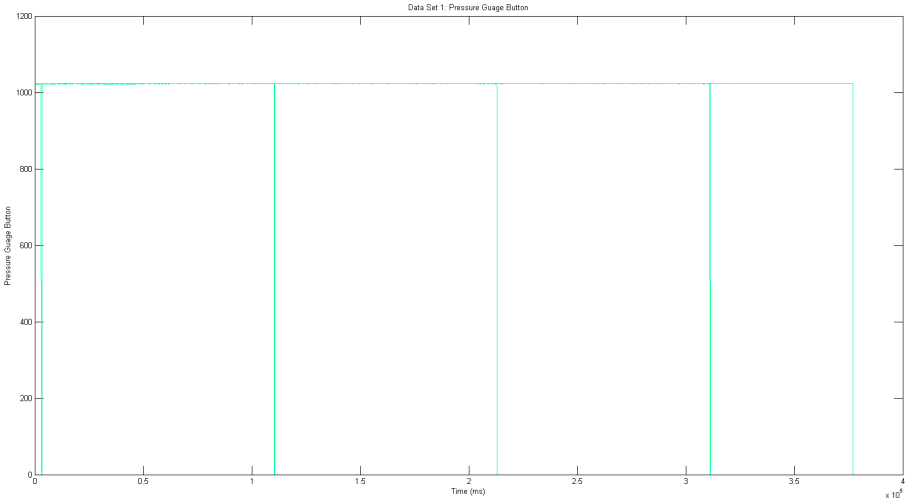
Plotting Experimental Data Set

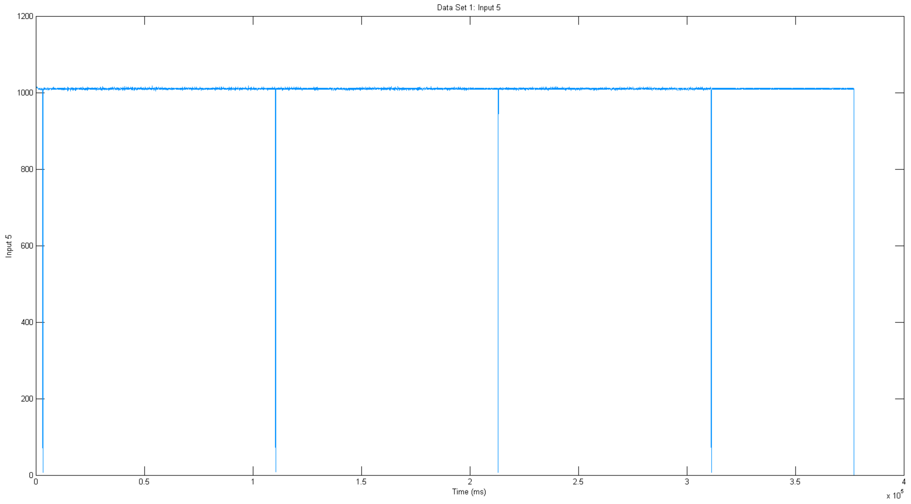
```
%close all;
clear all;
clc;
fig = 0;

files = {'data_take2.csv'}; %or {} or {file0.csv, file1.csv... etc.}
files = {};
if length(files) <= 0
    files = uigetfile('*.csv','Select Data Set(s)','MultiSelect','on');
end
if ~iscell(files) %force into iterable, if only 1 input file.
    files = {files};
end

for i = 1:length(files)

    col_data   = csvread(files{i}, 1);
    time_mills = col_data(:,1);
    titles = {'Time (ms)'; ...
        'Pressure Sensor'; 'Trigger Button'; 'Pressure Guage Button'};
    clear data;
    colors = hsv(min(size(col_data)));
    for j = 1:min(size(col_data))

        if j <= length(titles)
            data{j}.title = titles{j};
        else
            data{j}.title = sprintf('Input %.0d', j);
        end
        %data{j}.title = sprintf('Input %.0d', j);
        data{j}.y     = col_data(:,j);
        %fixing Arduino conversion factors... from first run.
        if mean(data{j}.y) < 5
            data{j}.y = data{j}.y .* 1023.0 ./ 5.0;%bad voltage conversion.
        end
        data{j}.x     = time_mills;
        % don't plot TIME vs TIME.
        if j ~= 0 %usually 1, debug 0
            fig = fig+1;
            figure(fig);
            plot(  data{j}.x, data{j}.y, 'color', colors(j,:));
            ylabel( data{j}.title);
            xlabel(titles(1));
            title(sprintf('Data Set %.0d: %s', i, data{j}.title));
        end
```

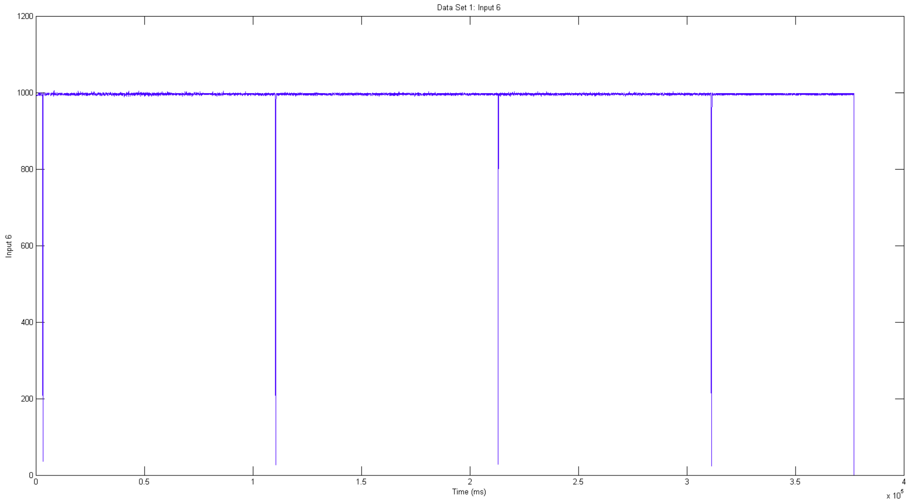

Data Set 1: Time (ms)
x 10^5
4
3.5
3
2.5
2
1.5
1
0.5
0
Time (ms)
0
0.5
1
1.5
2
2.5
3
3.5
4
Time (ms)
x 10^5

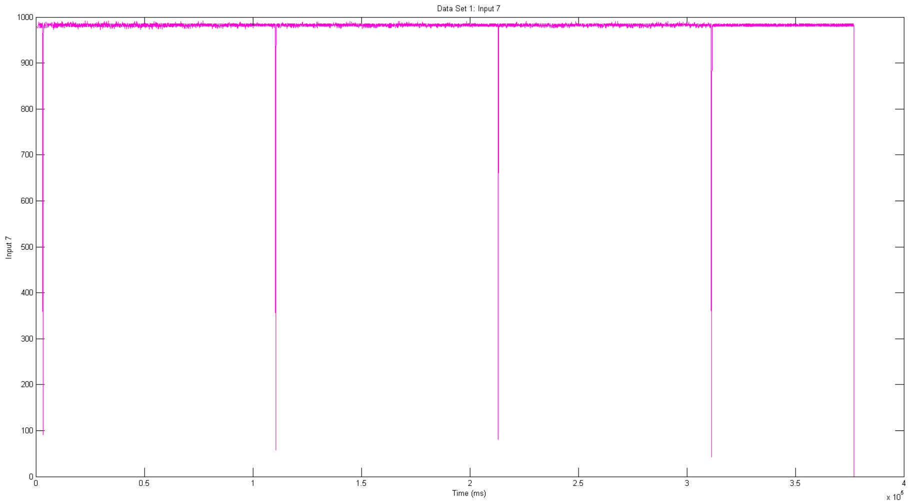

Data Set 1: Pressure Sensor
240
220
200
180
160
140
120
100
80
Pressure Sensor
0
0.5
1
1.5
2
2.5
3
3.5
4
Time (ms)
x 10^5


Data Set 1: Trigger Button
Trigger Button
1200
1000
800
600
400
200
0
0
0.5
1
1.5
2
2.5
3
3.5
4
Time (ms)
x 10^5

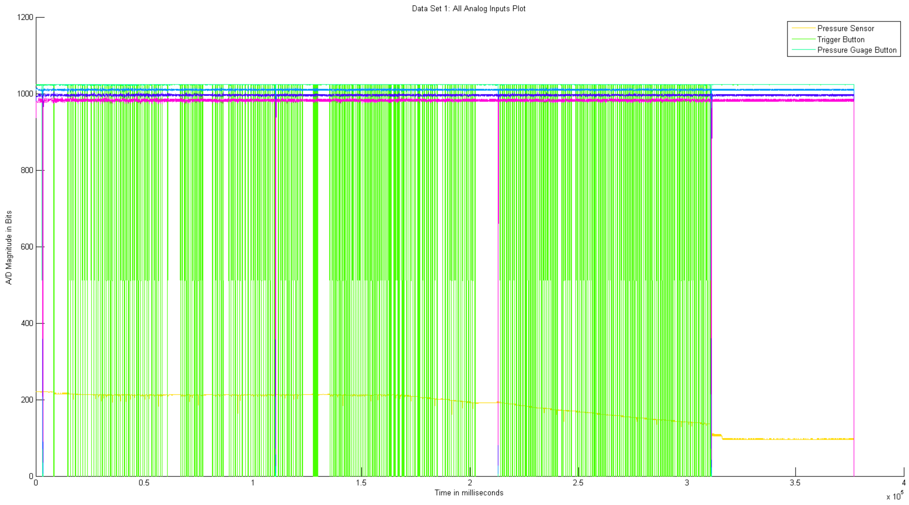

Data Set 1: Pressure Guage Button
Pressure Guage Button
1200
1000
800
600
400
200
0
0
0.5
1
1.5
2
2.5
3
3.5
4
Time (ms)
x 10^5


Data Set 1: Input 5
1200
1000
800
600
400
200
0
Input 5
0
0.5
1
1.5
2
2.5
3
3.5
4
Time (ms)
x 10^5


Data Set 1: Input 6
1200
1000
800
600
400
200
0
Input 6
0
0.5
1
1.5
2
2.5
3
3.5
4
Time (ms)
x 10^5

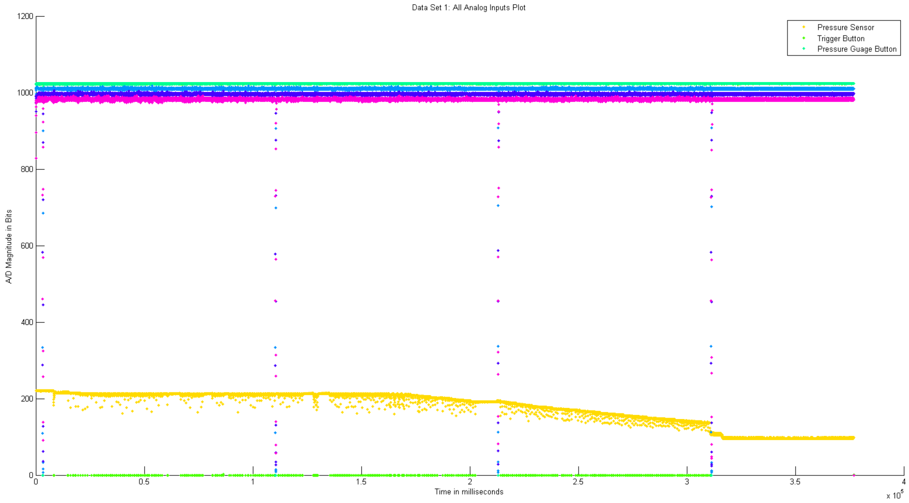


```
end

fig = fig+1;
figure(fig);
hold on;
for j = 2:length(data)
    plot(data{j}.x, data{j}.y, 'Color', colors(j,:));
end
ylabel('A/D Magnitude in Bits'  );
xlabel('Time in milliseconds'   );
legend(titles(2:length(titles)) );
title(sprintf('Data Set %.0d: %s', i, 'All Analog Inputs Plot') );
hold off;
```



```
fig = fig+1;
figure(fig);
```

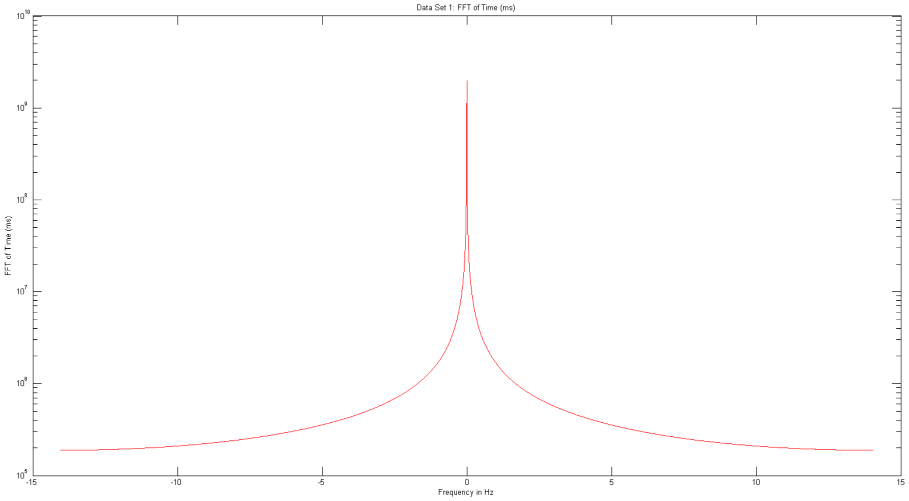
```
hold on;
for j = 2:length(data)
    plot(data{j}.x, data{j}.y, '.', 'Color', colors(j,:));
end
ylabel('A/D Magnitude in Bits'  );
xlabel('Time in milliseconds'   );
legend(titles(2:length(titles)) );
title(sprintf('Data Set %.0d: %s', i, 'All Analog Inputs Plot') );
hold off;
```

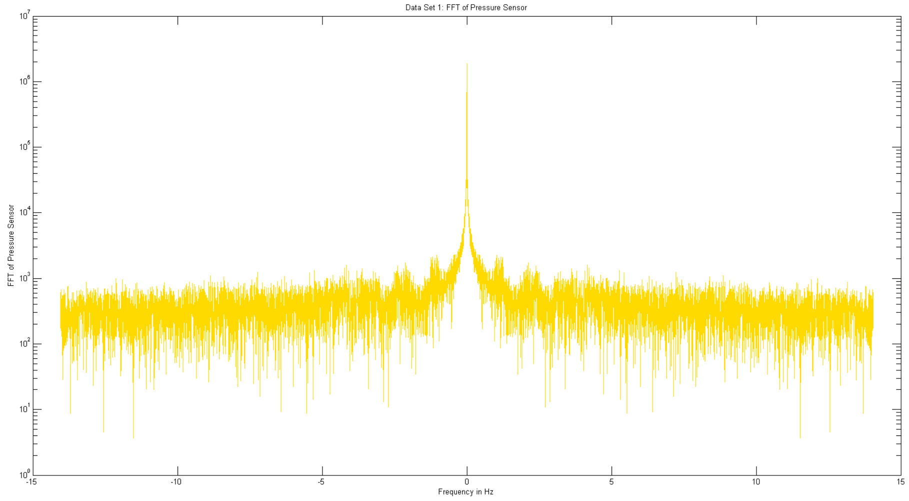


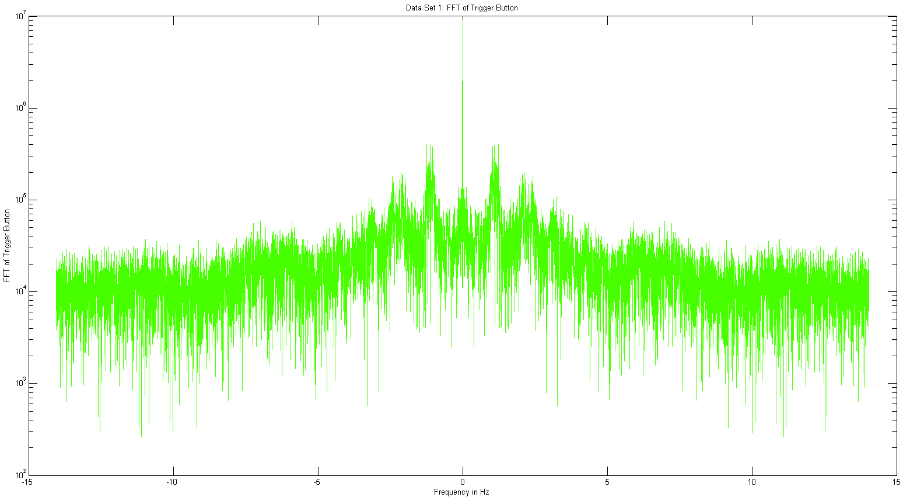
```
fit = polyfit(transpose(time_mills), 1:length(time_mills),1);
avg_period = 1/fit(1);
fprintf('\nData Set %s (%.0d) ', files{i}, i);
fprintf('has an Average Sampling Rate of %.3f ', avg_period);
fprintf('milliseconds.\n');
```

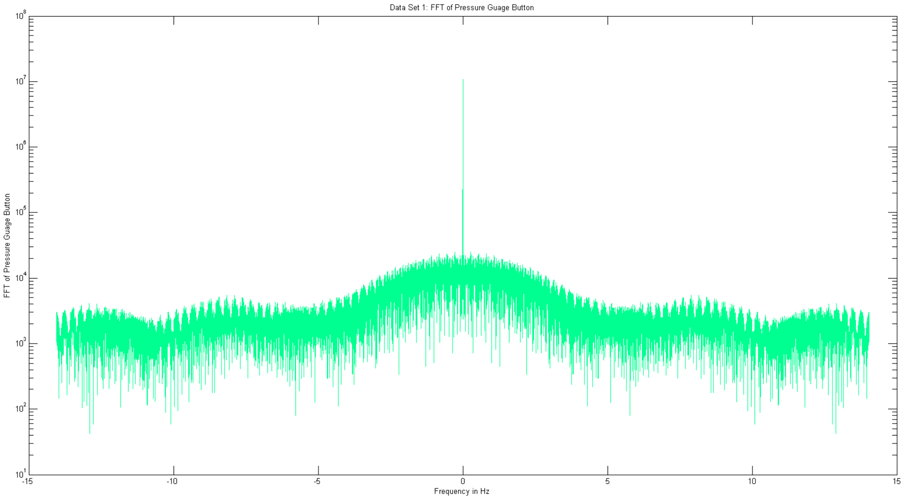
```
    Data Set data_take2.csv (1) has an Average Sampling Rate of 35.584 millise
```

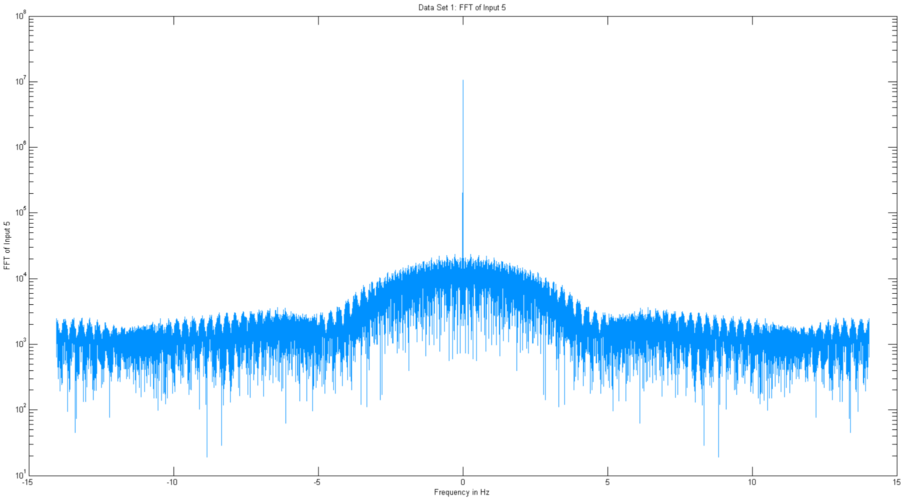
```
fs_rate   = 1000/(avg_period); %ms to Hz
N_samples = length(time_mills);
f_axis    = linspace(-fs_rate/2,fs_rate/2,N_samples);
% dft with length N, from frequency [-fs/2:fs/2], divided over N bins.
clear data;
for j = 1:min(size(col_data))

    if j <= length(titles)
        data{j}.title = titles{j};
    else
        data{j}.title = sprintf('Input %.0d', j);
    end
    data{j}.title = sprintf('FFT of %s', data{j}.title);
    data{j}.y = col_data(:,j);
    %fixing bad Arduino conversion factors... from first run.
    if mean(data{j}.y) < 5
        data{j}.y = data{j}.y .* 1023.0 ./ 5.0;%return volts to bits.
    end
    data{j}.y     = fftshift(fft(data{j}.y));
    data{j}.x     = f_axis;
    % don't plot TIME vs TIME.
    if j ~= 0 %usually 1, debug 0
        fig = fig+1;
        figure(fig);
        semilogy(  data{j}.x, abs(data{j}.y), 'color', colors(j,:));
        ylabel( data{j}.title);
        xlabel('Frequency in Hz');
        title(sprintf('Data Set %.0d: %s', i, data{j}.title));
    end
```

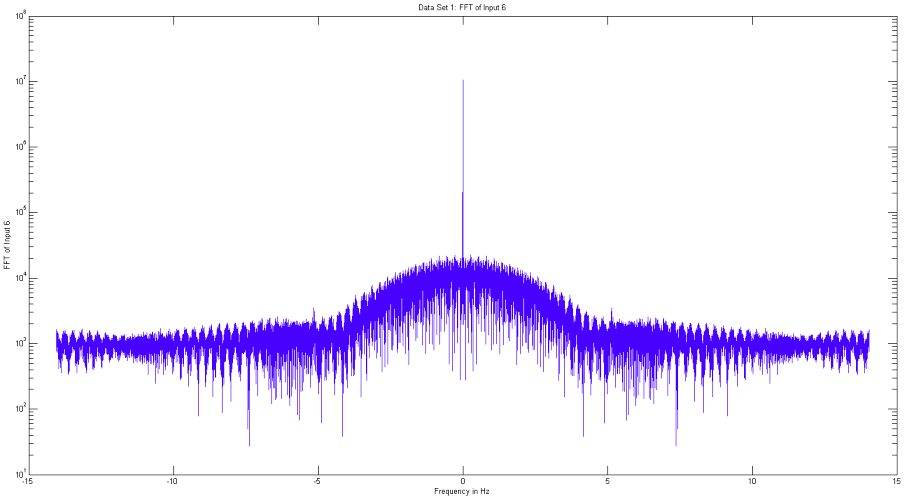


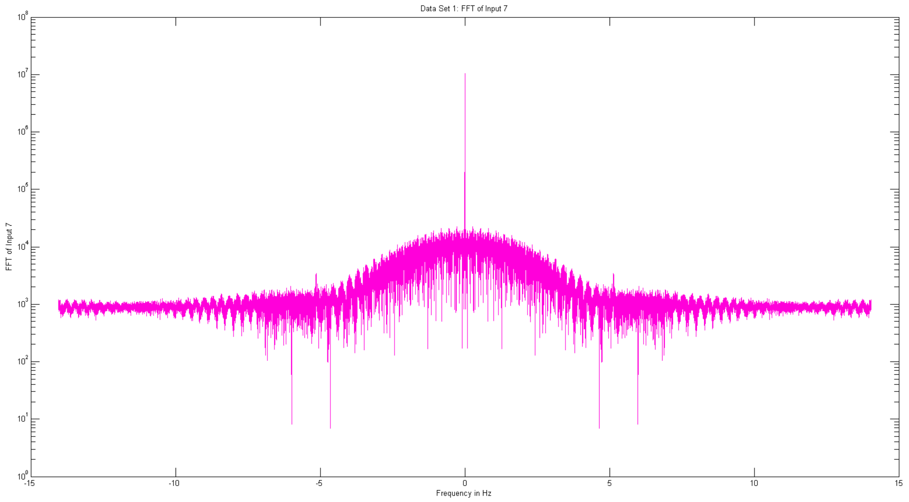



Data Set 1: FFT of Trigger Button
FFT of Trigger Button
Frequency in Hz
10^7
10^6
10^5
10^4
10^3
10^2
-15
-10
-5
0
5
10
15

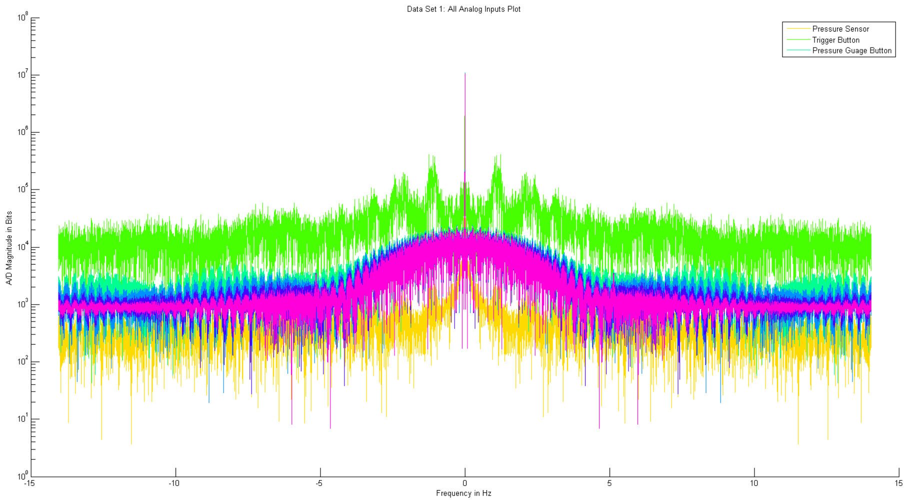

Data Set 1: FFT of Pressure Guage Button
FFT of Pressure Guage Button
Frequency in Hz
10^8
10^7
10^6
10^5
10^4
10^3
10^2
10^1
-15
-10
-5
0
5
10
15


Data Set 1: FFT of Input 5
FFT of Input 5
10^8
10^7
10^6
10^5
10^4
10^3
10^2
10^1
-15
-10
-5
0
5
10
15
Frequency in Hz


Data Set 1: FFT of Input 6
FFT of Input 6
10^8
10^7
10^6
10^5
10^4
10^3
10^2
10^1
-15
-10
-5
0
5
10
15
Frequency in Hz

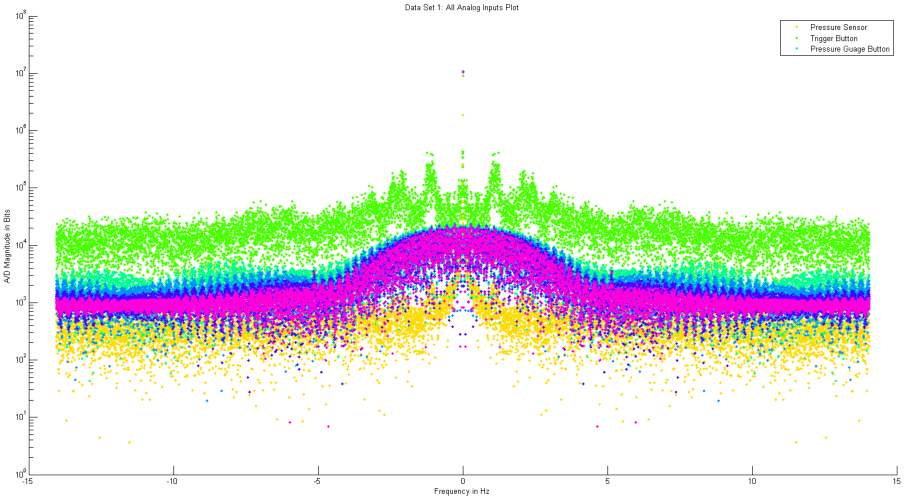


```
end

fig = fig+1;
figure(fig);
hold on;
for j = 2:length(data)
    semilogy(data{j}.x, abs(data{j}.y), 'Color', colors(j,:));
end
ylabel('A/D Magnitude in Bits'  );
xlabel('Frequency in Hz'        );
legend(titles(2:length(titles)) );
title(sprintf('Data Set %.0d: %s', i, 'All Analog Inputs Plot') );
set(gca, 'yscale','log');
hold off;
```



```
fig = fig+1;
figure(fig);
```

```
    hold on;
    for j = 2:length(data)
        semilogy(data{j}.x, abs(data{j}.y), '.', 'Color', colors(j,:));
    end
    ylabel('A/D Magnitude in Bits'  );
    xlabel('Frequency in Hz'        );
    legend(titles(2:length(titles)) );
    title(sprintf('Data Set %.0d: %s', i, 'All Analog Inputs Plot') );
    set(gca, 'yscale','log');
    hold off;
```



```
end
```

Published with MATLAB® R2014a