**ECE304**

**Lab3&4 Memo**

**Bruce Jiang, Siwei Cai**

**Introduction**

The objective of this week’s experiment is to transfer a simulation of nulling filter from MATLAB to Arduino. Within the limited storage size on Arduino, applied algorithm should operate on fixed point format instead of floating points. To understand better, experimenting on MATLAB to see what the graph would looks like is the first step.

**Plots Results**

Chart, histogram

Description automatically generated

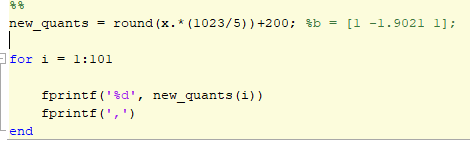
***Figure 1 – MATLAB Input & Output Discretized Signals***

Chart, line chart, histogram

Description automatically generated

***Figure 2 – Arduino Nulling Filter Simulation From MATLAB***

From MATLAB, to get figure 1 and 2, we first convert the x value into A/D by multiplying 1023 over 5, then rounded and shifted it. Using the new quantized samples and applied to y function to get nulling filter process done.



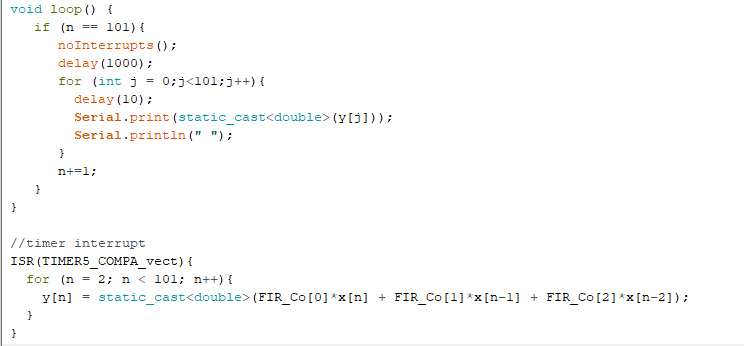
***Figure 3 – MATLAB Code Quantization***

Next step implements the new quantization table from above into Arduino and convert them and FIR coefficients properly into Q format.

Below are the Arduino functions to convert a given integer or float number into the fixed format numbers.



***Figure 4 – Arduino Code***



***Figure 5 – Arduino Code***

***Figure 6 – Arduino FIR Outputs in Excel***

**Conclusion**

In this lab, the nulling coefficient we implemented into Arduino used 15 decimal bits and 16 fractional bits for all the conversions include integer and float points. And in the compelling display the Arduino board have used up 1024 bytes of Global variables which is 8192 bits, 12% of 8192 bytes. Thus, the integer part in x values we implemented from MATLAB could reduce the fractional bits size at the end to save more memory of Global variables. Finally, the Figure 5 looks identical to figure 2, indicating the success of the filter simulating transfer.

**Appendix**

<https://github.com/Pharap/FixedPointsArduino>