**Initialize HW and FFT**

Initialize Microphone

Initialize FFT table for a sample length of 1024 // initialize manually, bug in DSP library

Initialize RGB for LCD

Initialize LCD

Initialize fill screen

While(1) // we are sampling and processing continuously

**Take samples at 44.1 KHz and process**

For( i=0; i < 44.1k; i){

**Get raw voltage data from Mic (J1.6)**

**Store raw data in arrays**

SoundArray[time] // 16 bit integers, used to calculate RMS

SoundBufferIn[time] // 32 bit float, used for FFT

Update Sum of input voltages // will be used for RMS calculation

If ( time == sample length){

**Calculate RMS**

Reset sum

Reset time

}

}

**Process input voltage with Fast Fourier Transform**

**Call arm\_rfft\_f32 with pointers to SoundBufferIn and SoundBufferOut**

// SoundBufferOut holds real and complex values next to each other in array

**Calculate magnitude of each set of real and imaginary numbers in decibels**

Counter = 0; // magnitude array shall be half of sample length

For( int i = 2; i < sample length, i += 2){ //skip first set of transf. values

MagnitudeArray[counter]

= 20\*Log10f \* magnitude of SoundBuffOut[i] and SoundBufferOut[i+1]

// convert magnitude to decibels

Update sum of decibel values // used to calculate average dB value

Counter++;

}

**Calculate bin number**

ADC Sampling Rate / Sample Length // ADC set to 125 KHz

**Get index number of max value in magnitude array**

**Sound Frequency = maxIndex \* bin**

Adjust bin offset // we use integers so we can ignore remainders

SoundFrequ = (maxIndex – (SoundFrequency/(SampleLength/2)))\*bin

**Calculate average Decibel value**

Sum of decibel values / (SAMPLELENGTH/2)

**Write parameters on top of LCD screen**

dB

RMS

Frequency

Bin

**Plot graph on LCD**

**Draw axes** // magnitude over frequency

i=0;

While ( i < SampleLength/2){

**Plot point from MagnitudeArray[i] // a decibal value**

**Increment plot pointer on x axis**

I++

}

**Update numerical values on top of graph**

Average decibel value

Sound RMS

Input frequency

Bin frequency