**Lab Report #4:** Data Conversion: Analog-to-Digital and Digital-to-Analog

Brian Dawson

Joshua Annandsingh

Microcomputer Systems

10/25/19

The College of New Jersey

Department of Electrical and Computer Engineering (ECE)



**Introduction:**

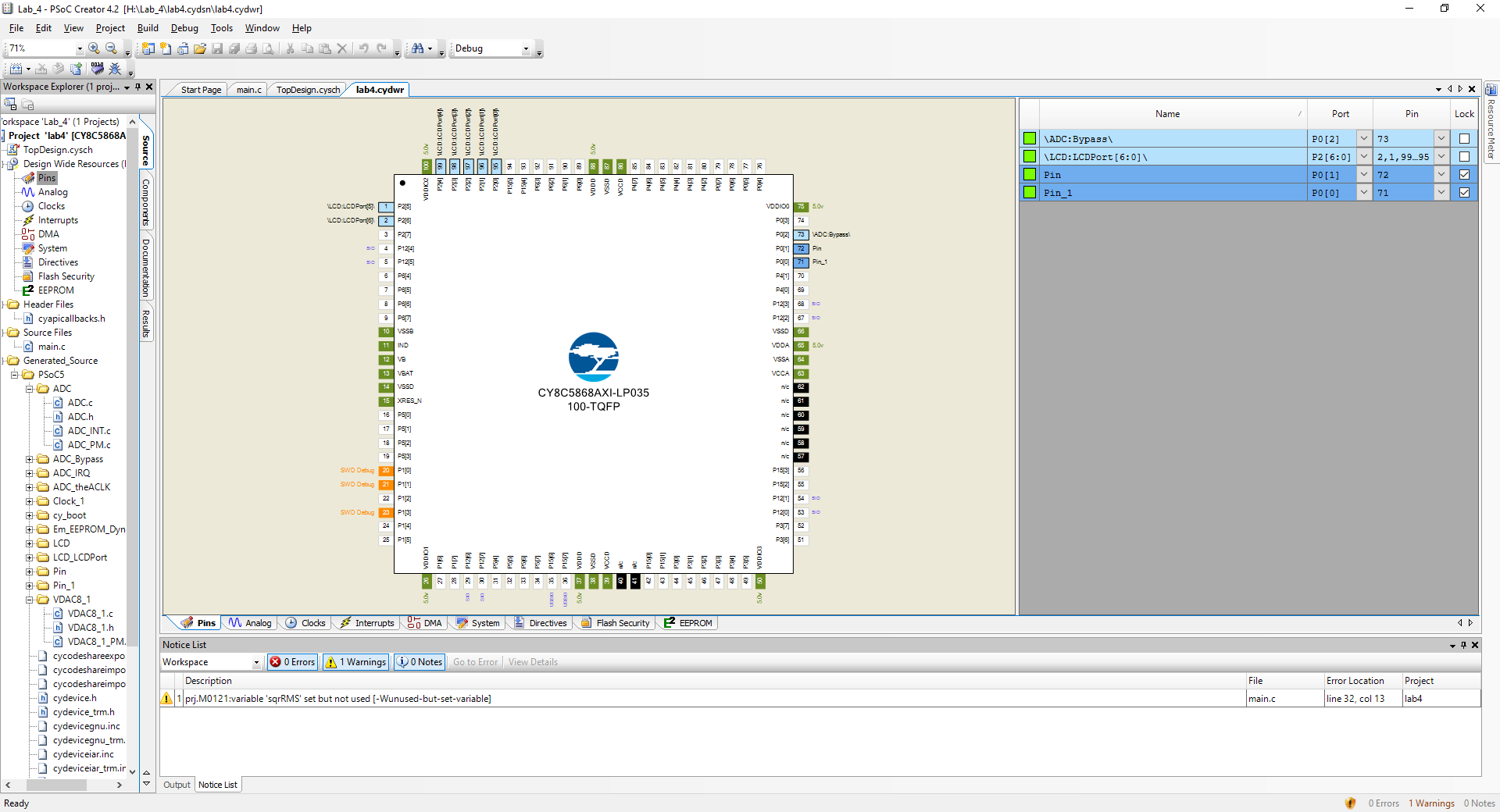
In this lab, the students had to use PSoC Creator to create a project using the CY8CKIT-050 kit. The project was to create a real time digital signal processing system using the PSoC. The project would achieve this using a SAR ADC and a VDAC.

**Procedure:**

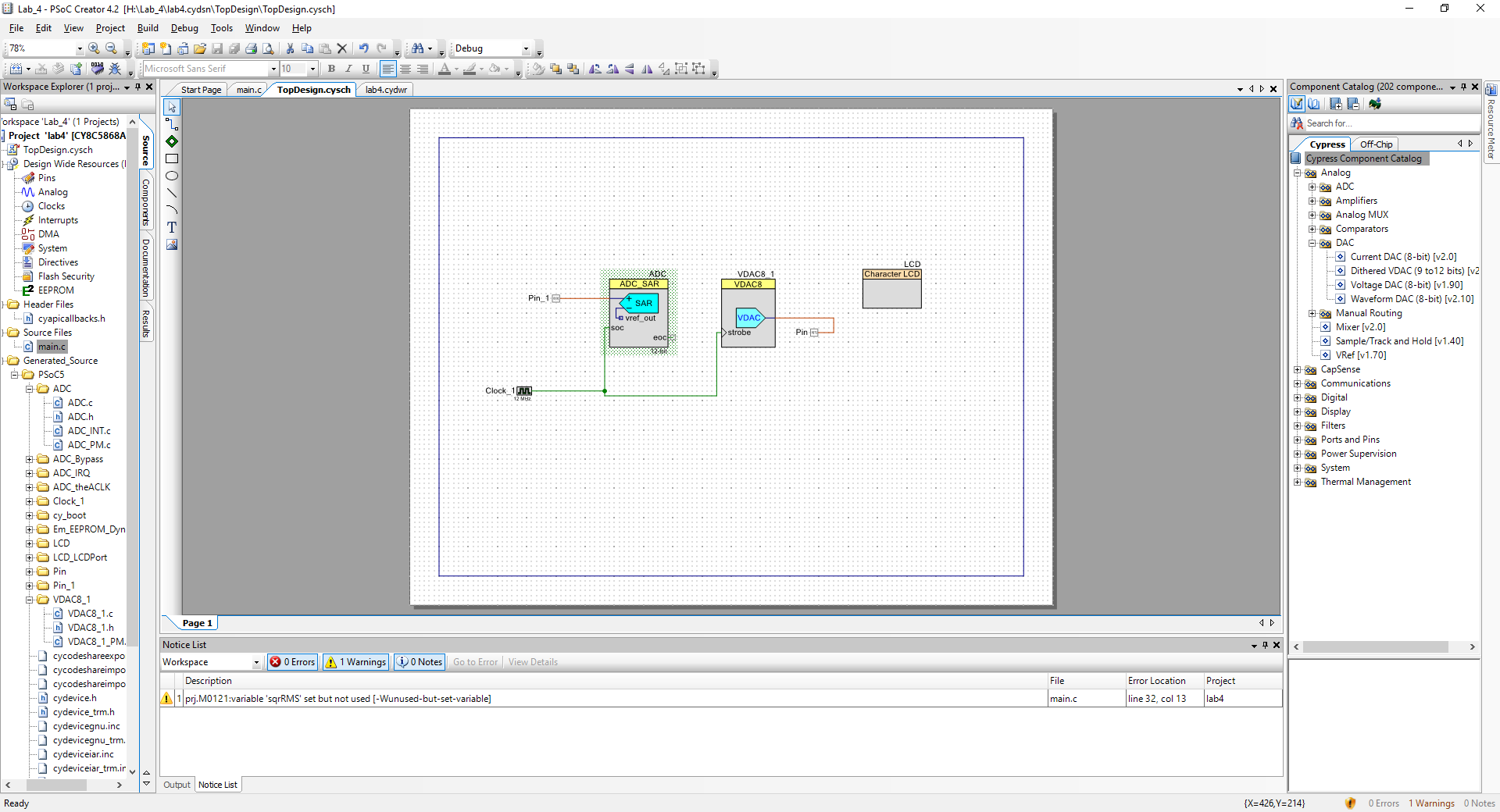
1. The students created a new PSoC project.
2. The students added the ADC and the DAC to the top design. They then connected a clock to both the ADC and the DAC.
3. The students then connected both the ADC and the DAC to pin creating a way to both input and output signals.
4. The students then made sure that the pins were mapped correctly to pin on the board.
5. The students then created code to receive an unknown incoming signal and determine the minimum and maximum values, the mean value, the RMS, and whether the signal was a sine, triangle, or square wave.
6. The students connected the LCD display to the board and then confirmed the code did the process correctly and displayed the correct information.

**Design:**

1. PSoC schematics



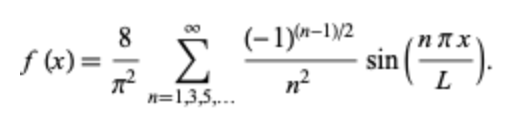
*Figure 1: PSoC Creator Pins*

**

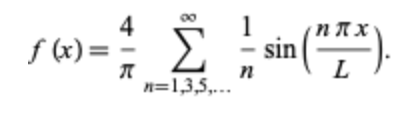
*Figure 2: PSoC Creator Top Design*

1. Equations

Fourier Series for Triangle wave



Fourier Series for Square wave



RMS Value for Sine Wave

= VMax/(√2)

= 2000/(√2)

= 1448mV

RMS Value for Square Wave

= VMax/(√3)

= 1.75/(√2)

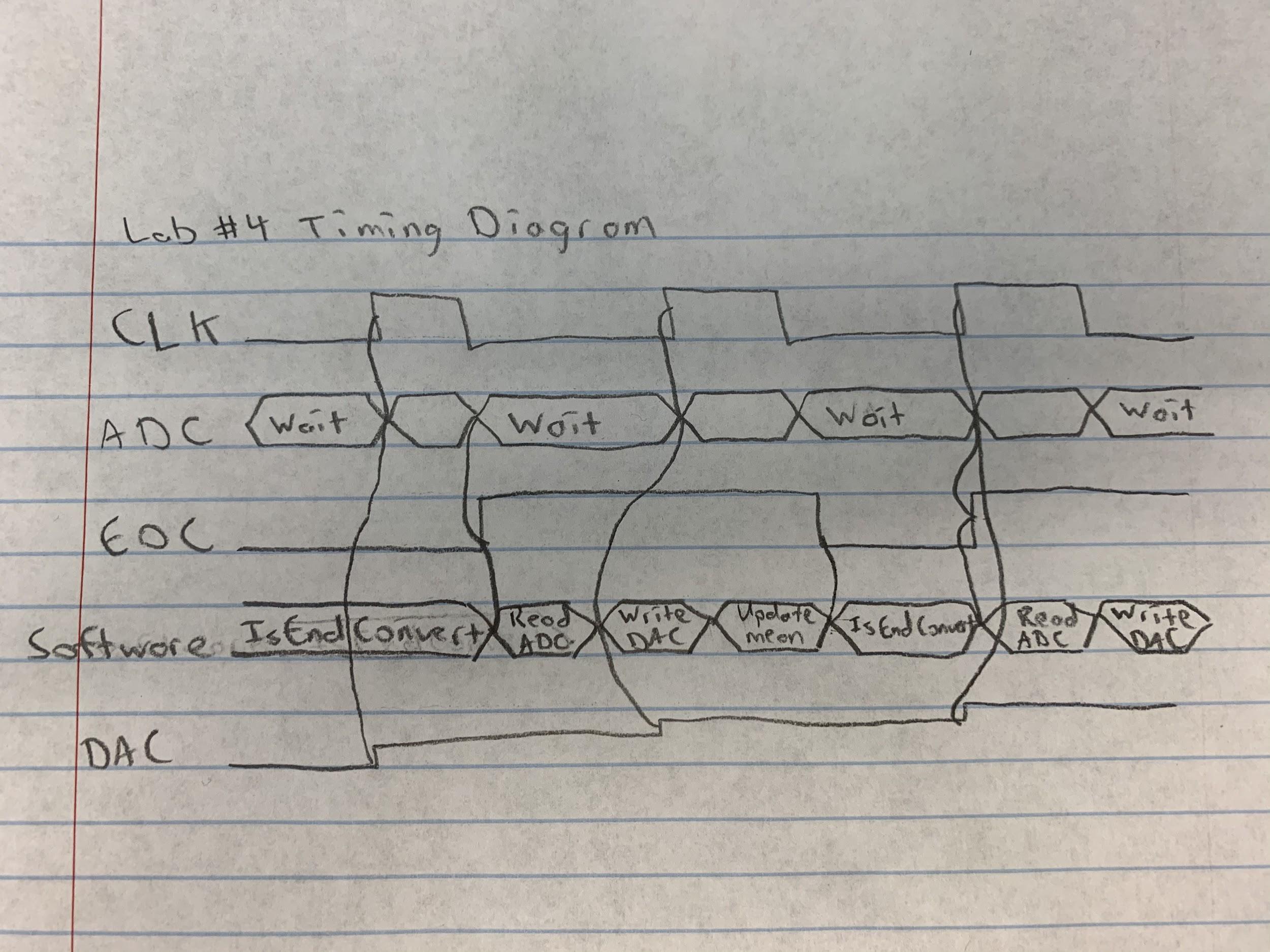
= 1011mV

RMS Value for Triangle Wave

= VMax

= 1500mV

1. Timing Diagram



1. Code

/\* ========================================

\*

\* Copyright YOUR COMPANY, THE YEAR

\* All Rights Reserved

\* UNPUBLISHED, LICENSED SOFTWARE.

\*

\* CONFIDENTIAL AND PROPRIETARY INFORMATION

\* WHICH IS THE PROPERTY OF your company.

\*

\* ========================================

\*/

#include "project.h"

#include <stdio.h>

#include <string.h>

#include <stdint.h>

#include <math.h>

int main(void)

{

CyGlobalIntEnable; /\* Enable global interrupts. \*/

/\* Place your initialization/startup code here (e.g. MyInst\_Start()) \*/

ADC\_Start(); //Starts ADC

LCD\_Start(); //Starts LCD

VDAC8\_1\_Start(); //Starts VDAC

int vMin = 0;

int vMax = 0;

int vAvg = 0;

int sinRMS = 0;

int triRMS = 0;

int sqrRMS = 0;

int sum = 0;

Clock\_1\_Start(); //Starts the clock

char output[4096]; //Sets output variable to be used later

for(int i = 0; i <5000; i++) // For loop for collecting data

{

ADC\_IsEndConversion(ADC\_WAIT\_FOR\_RESULT); //Waits for conversion of data in ADC

int v1= ADC\_GetResult16(); //Sets 16 bit ADC result to equal an integer

v1 = ADC\_CountsTo\_mVolts(v1); //Converts to millivolts

sum += v1; //sum = sum + v1

VDAC8\_1\_SetValue(v1); //Sets the value of the VDAC to v1

if(v1<vMin) // If v1 is less than min set min equal to v1

{

vMin = v1;

}

if(v1>vMax) //If v1 is greater than max set max equal to v1

{

vMax = v1;

}

vAvg= sum/i; // Calculates average from sum and number of inputs

sinRMS= (vMax/sqrt(2)); //Calculates the rms of a sine wave

triRMS = (vMax/sqrt(3)); //Calculates the rms of a triangle wave

sqrRMS = vMax; // Calculates the rms of a square wave

}

if (sinRMS > 1400 && sinRMS < 1500) //If statement to check if wave is a sine wave

{

for(;;)

{

//Outputs the type of wave, the rms, the maximum, and the minimum

LCD\_ClearDisplay(); //Clears the LCD

sprintf(output,"Wave : Sine" ); //Prints text

LCD\_PrintString(output); //Prints output

LCD\_Position(1,0); //Move to second position on LCD

sprintf(output,"sinRms = %dmV", sinRMS); //Prints text

LCD\_PrintString(output);//Prints output

CyDelay(2000);//Waits for 2 seconds

LCD\_ClearDisplay();

sprintf(output,"vMax = %dmV", vMax );//Prints text

LCD\_PrintString(output);//Prints output

LCD\_Position(1,0);//Moves to second position on LCD

sprintf(output,"vMin = %dmV", vMin);//Prints text

LCD\_PrintString(output);//Prints output

CyDelay(2000);//Waits for 2 seconds

LCD\_ClearDisplay();

sprintf(output,"vAvg = %dmV", vAvg);//Prints text

LCD\_PrintString(output);//Prints output

CyDelay(2000);//Waits for 2 seconds

}

}

else if (triRMS > 950 && triRMS < 1100) //Tests wave to see if it is a triangle wave

{

for(;;)

{

//Outputs the type of wave, the rms, the maximum, and the minimum

LCD\_ClearDisplay();

sprintf(output,"Wave : Triangle" );//Prints text

LCD\_PrintString(output);//Prints output

LCD\_Position(1,0);//Moves to second position on LCD

sprintf(output,"triRms = %dmV", triRMS);//Prints text

LCD\_PrintString(output);//Prints output

CyDelay(2000);//Waits for 2 seconds

LCD\_ClearDisplay();

sprintf(output,"vMax = %dmV", vMax );//Prints text

LCD\_PrintString(output);//Prints output

LCD\_Position(1,0);//Moves to second position on LCD

sprintf(output,"vMin = %dmV", vMin);//Prints text

LCD\_PrintString(output);//Prints output

CyDelay(2000);//Waits for 2 seconds

LCD\_ClearDisplay();

sprintf(output,"vAvg = %dmV", vAvg);//Prints text

LCD\_PrintString(output);//Prints output

CyDelay(2000);//Waits for 2 seconds

}

}

else

{

for(;;)

{

//Outputs the type of wave, the rms, the maximum, and the minimum

LCD\_ClearDisplay();

sprintf(output,"Wave : Square" );//Prints text

LCD\_PrintString(output);//Prints output

LCD\_Position(1,0);//Moves to second position on LCD

sprintf(output,"sqrRms = %dmV", vMax);//Prints text

LCD\_PrintString(output);//Prints output

CyDelay(2000);//Waits for 2 seconds

LCD\_ClearDisplay();

sprintf(output,"vMax = %dmV", vMax );//Prints text

LCD\_PrintString(output);//Prints output

LCD\_Position(1,0);//Moves to second position on LCD

sprintf(output,"vMin = %dmV", vMin);//Prints text

LCD\_PrintString(output);//Prints output

CyDelay(2000);//Waits for 2 seconds

LCD\_ClearDisplay();

sprintf(output,"vAvg = %dmV", vAvg);//Prints text

LCD\_PrintString(output);//Prints output

CyDelay(2000);//Waits for 2 seconds

}

}

}

**Results:**



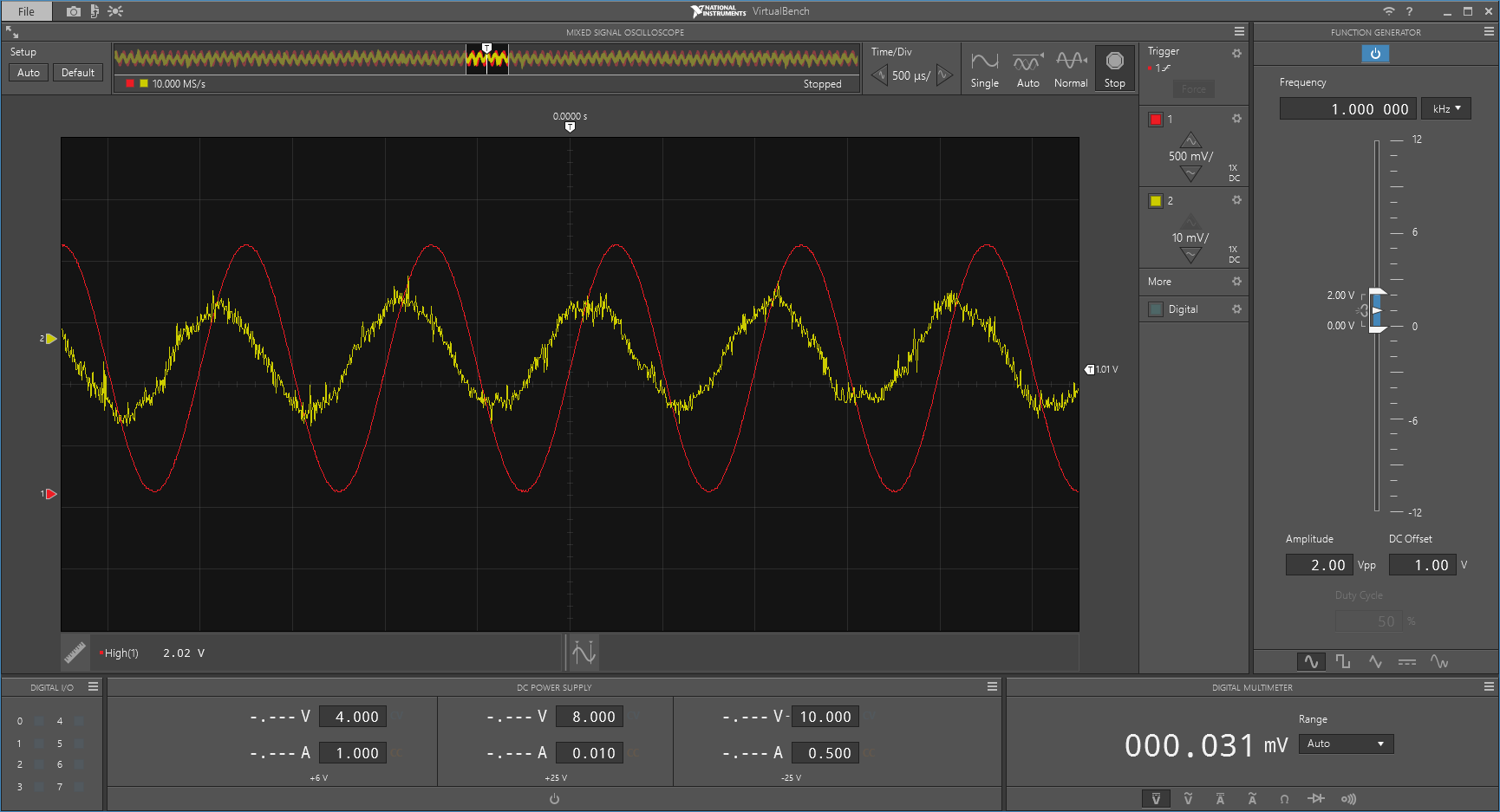
*Figure 1: Sine Wave and Sine Wave RMS*



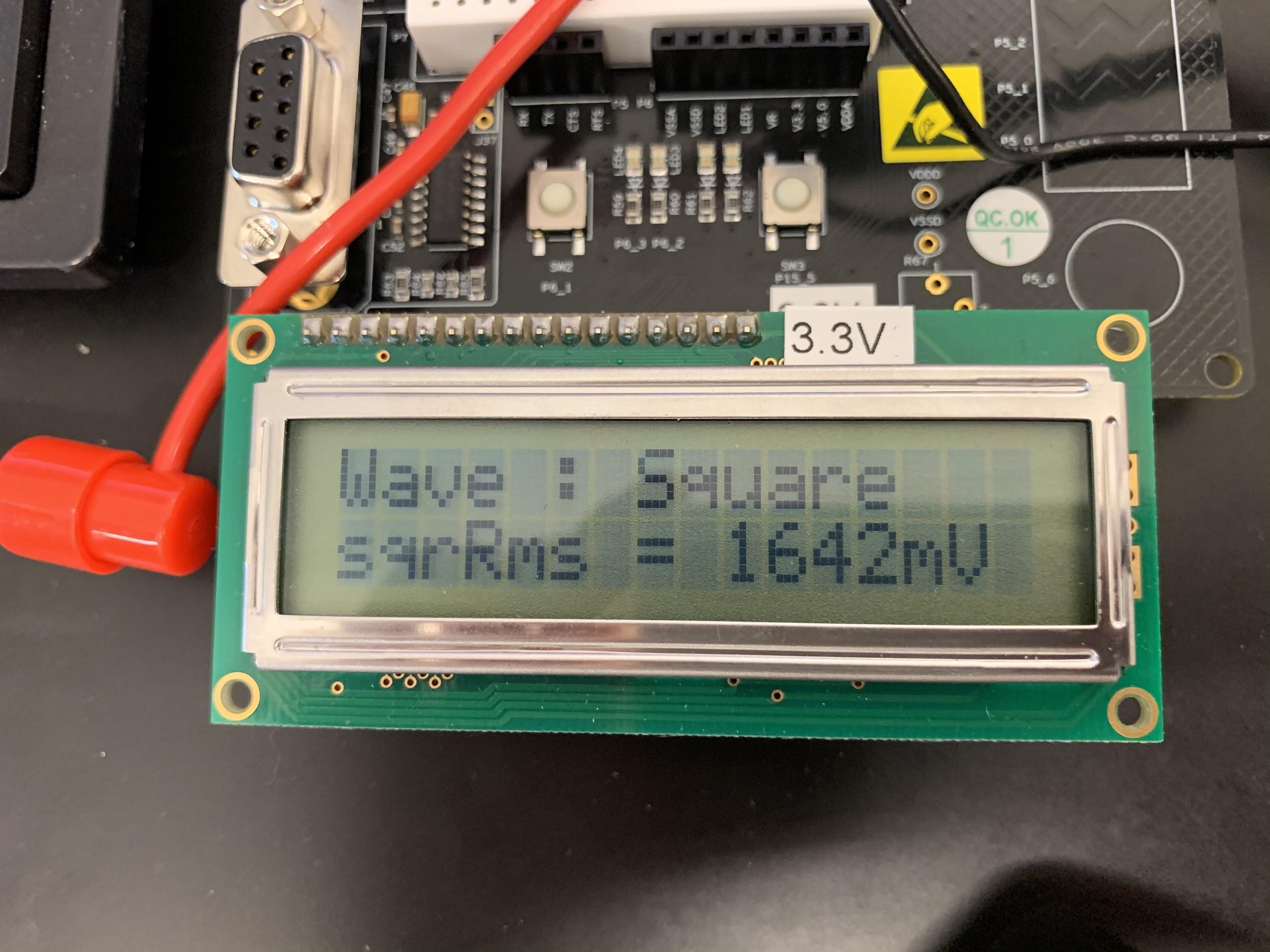
*Figure 2: Sine Wave Maximum and Sine Wave Minimum*



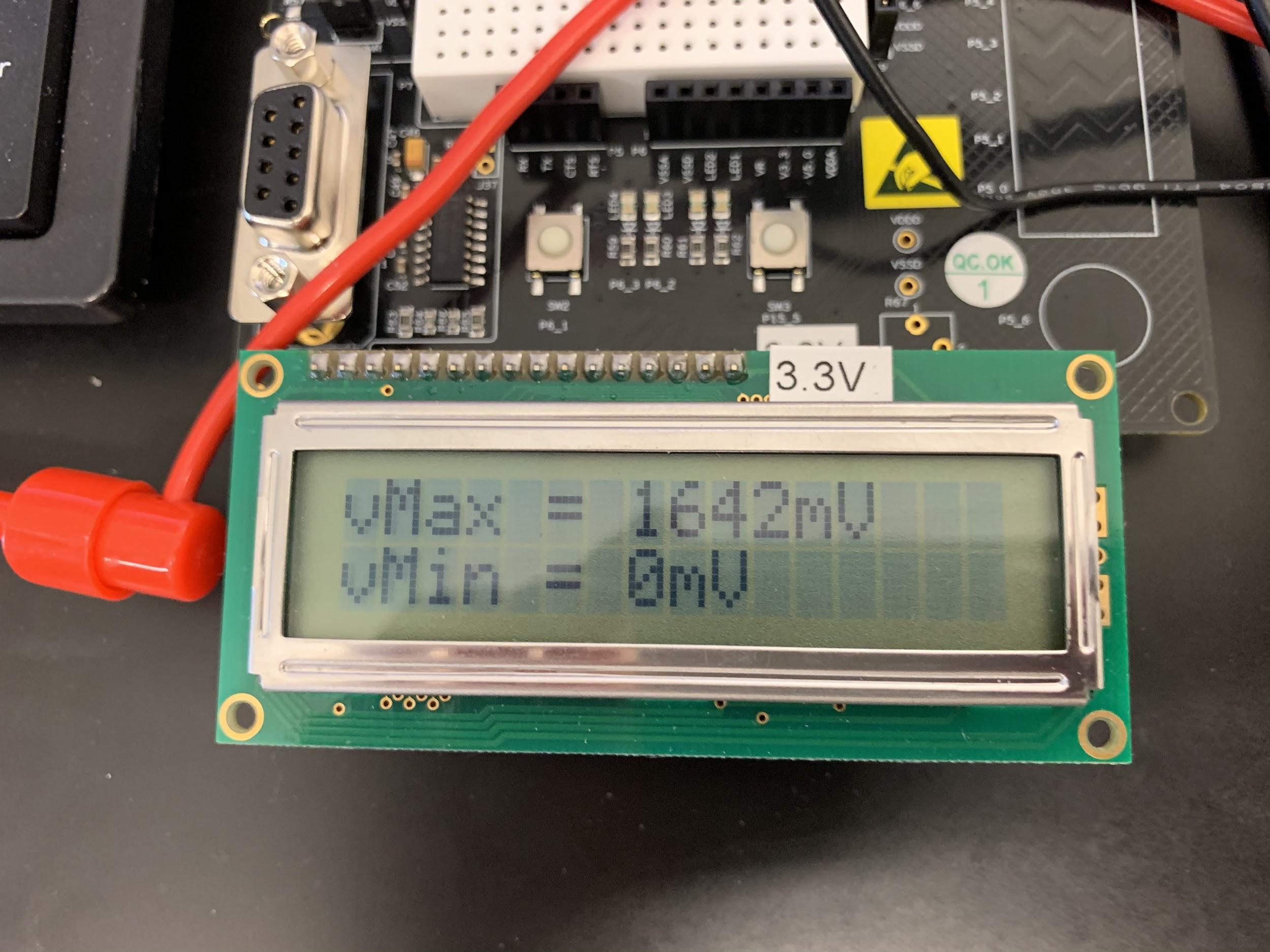
*Figure 3: Sine Wave Average*



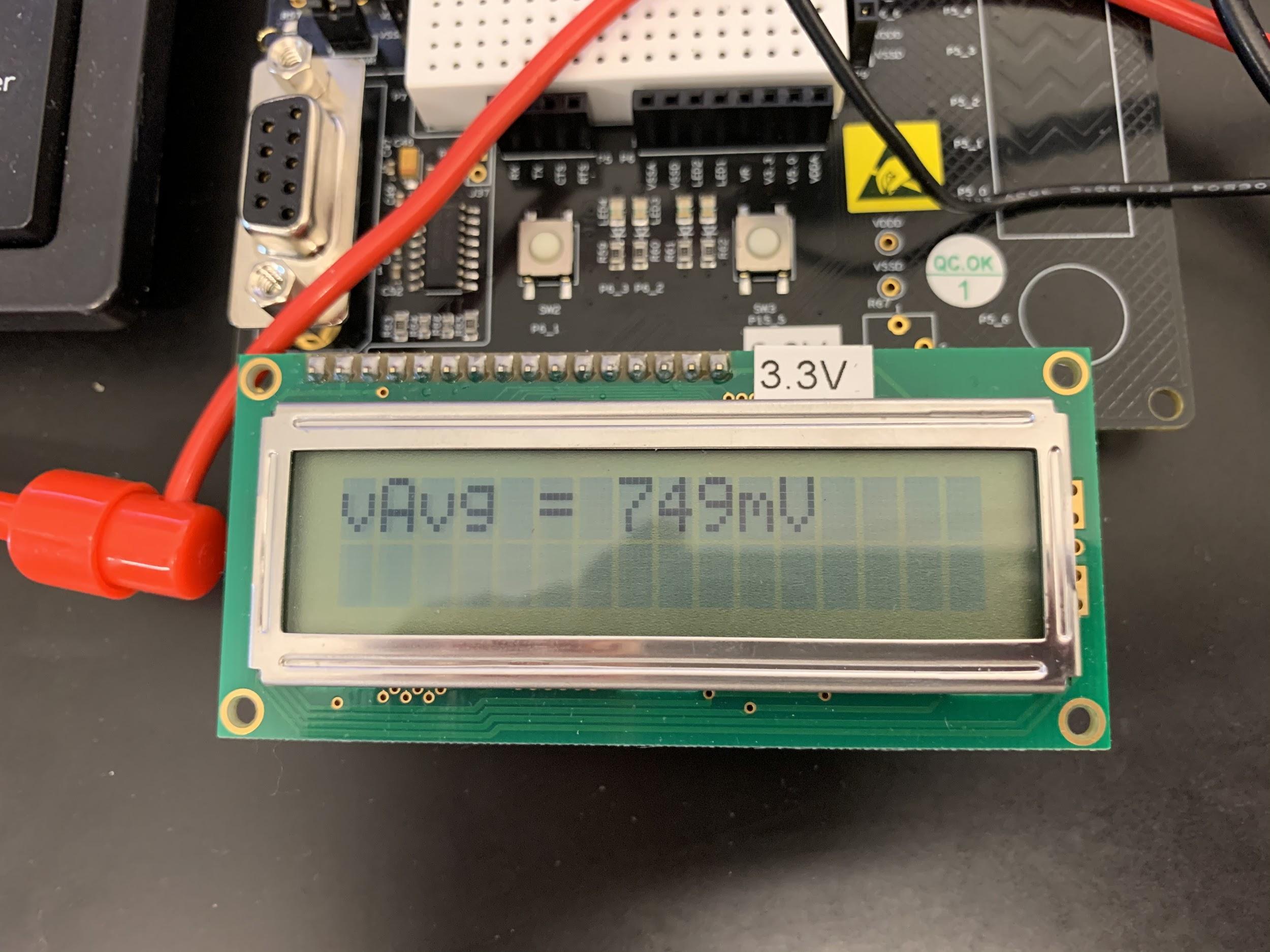
*Figure 4: Sine Wave Waveform and Output*

**

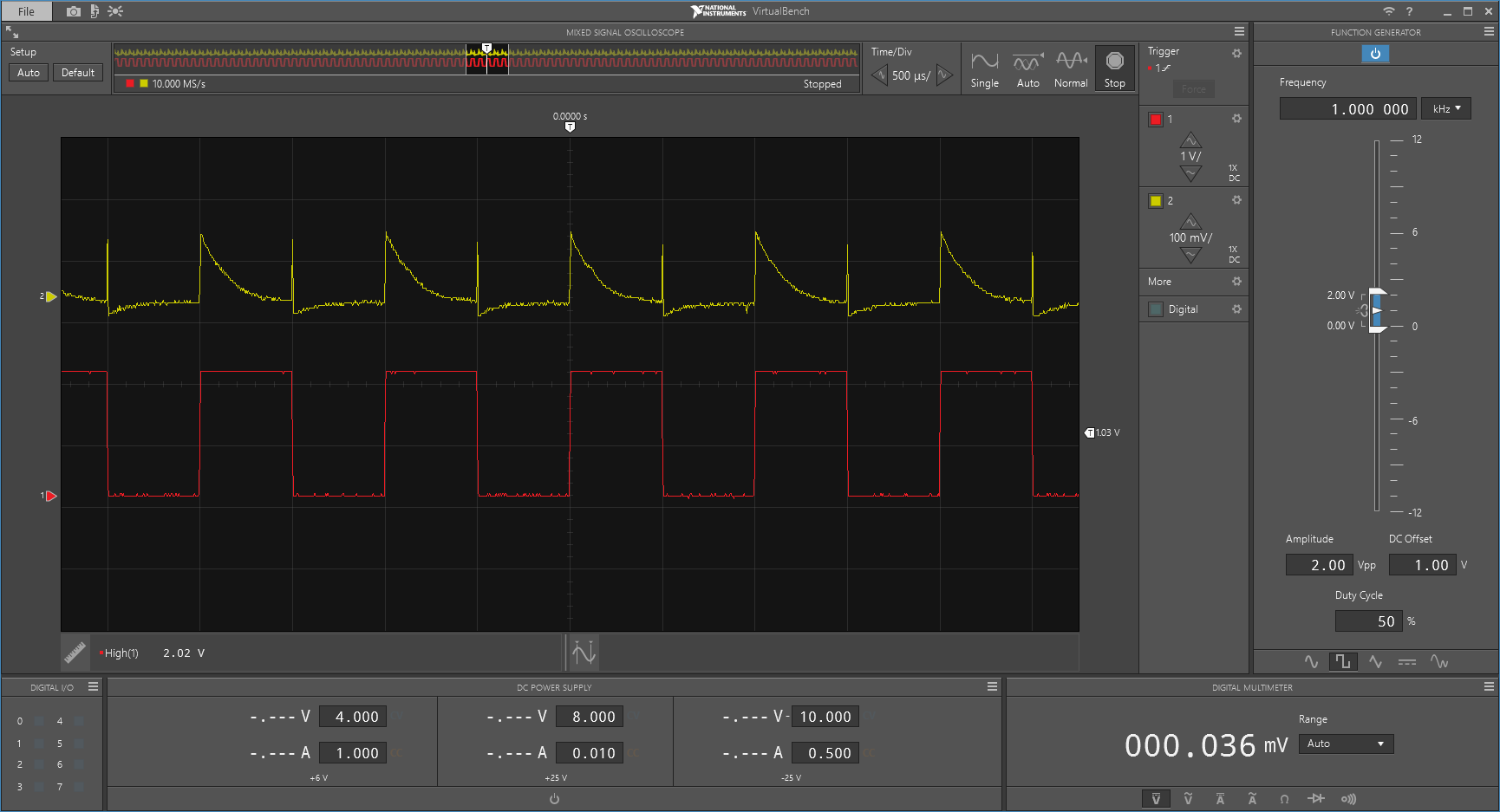
*Figure 4: Square Wave and Square Wave RMS*

**

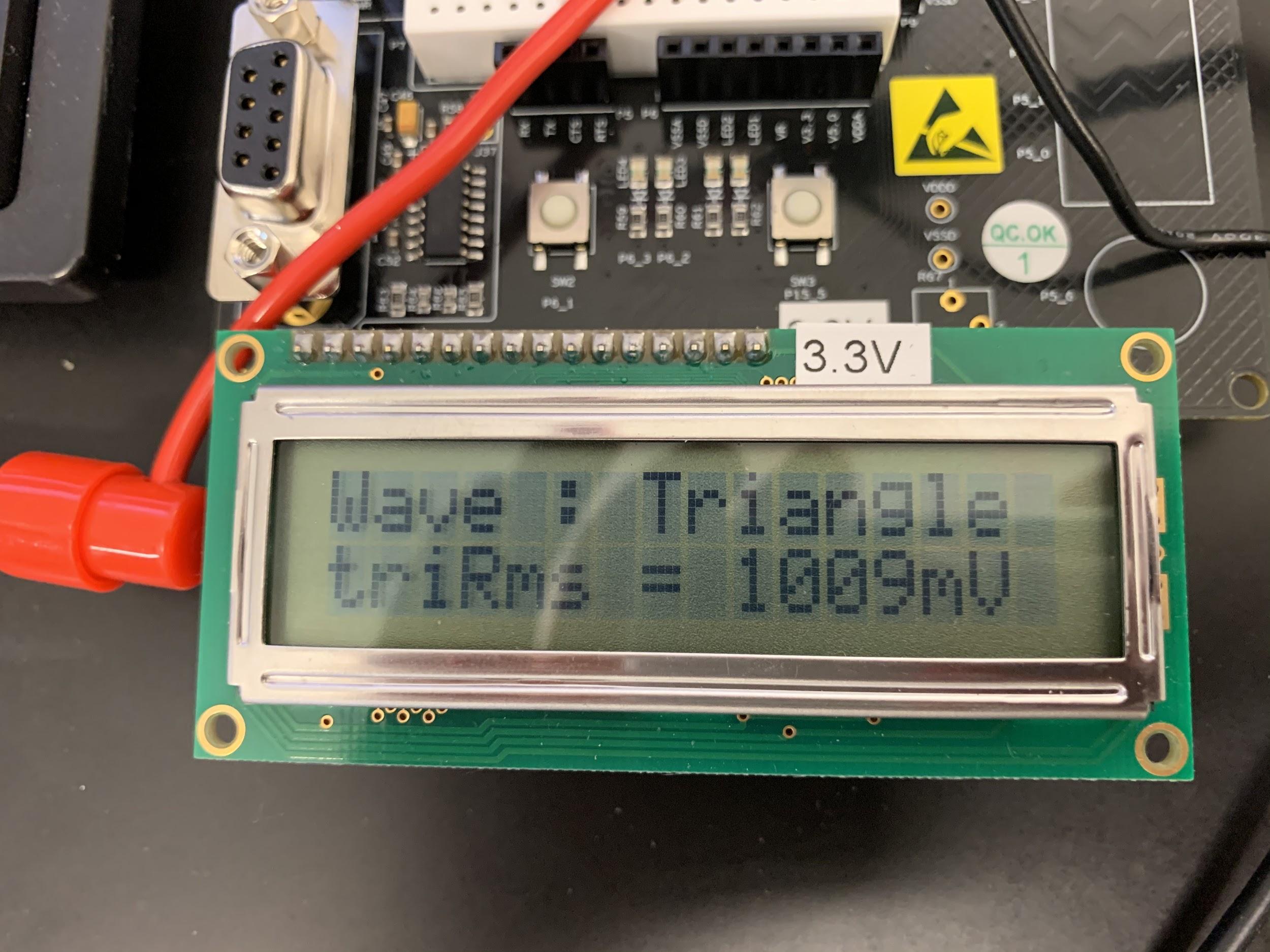
*Figure 5: Square Wave Maximum and Minimum*

**

*Figure 6: Square Wave Average*



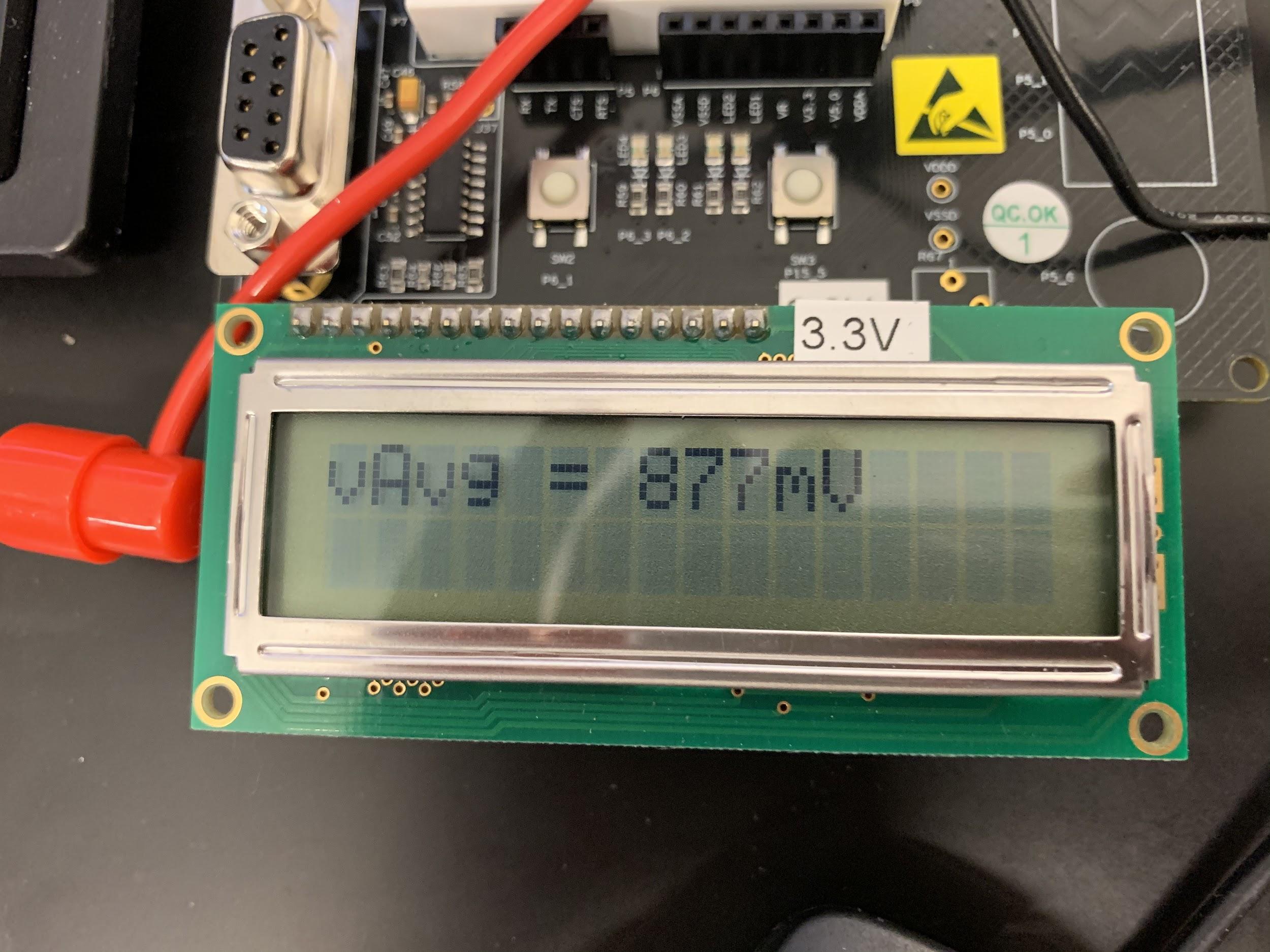
*Figure 7: Square Wave Waveform and Output*

**

*Figure 7: Triangle Wave and Triangle Wave RMS*

**

*Figure 8: Triangle Wave Maximum and Minimum*

**

*Figure 9: Triangle Wave Average*



*Figure 10: Triangle Wave Waveform and Output*

**Conclusion:**

The lab worked correctly and as expected. The code was able to differentiate the different waves and then provide the requisite information to the LCD display. The students found the way to use RMS to determine the different kinds of waves interesting and liked the idea of using this to be able to find out what kind of signal they were looking at without seeing the waveform.