Final Project: MSP430 Controlled Theremin

Skylar Adams
Rowan University

December 20, 2018

1 Design Overview

For the Final project, students were required to desing code for any MSP430 board of their choice which interfaces with sensors and utilizes one of the communication protocols supported by the board. In the case of this project, the MSP430F5529 was used as the brain of a Theremin controlled by distance sensors. One distance sensor controlled the frequency of the waveform, and the other would control the volume. For communication, UART was used to select between different pre-determined waveforms.

1.1 Design Features

The key design features include:

- MSP430F5529 used to create waveforms, analog to digital conversion, and UART communication
- Sharp GP2Y0A21YK0F IR distance sensor
- OPA548T op amp breakout board to drive output
- 4-bit R-2R ladder digital to analog converter

1.2 Featured Applications

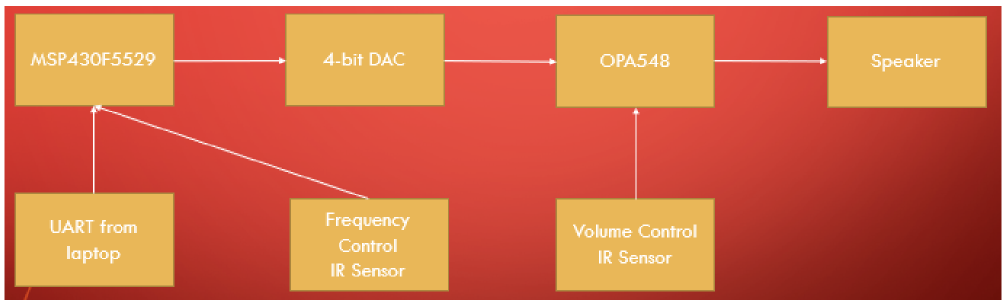
- Creating 4 different pre-set waveforms by converting digital signals to analog signals
- Changing frequency of waveforms via distance
- Changing volume output of a speaker via distance
- Converting analog signals into digital signals to use sensors

1.3 Design Resources

The github link for the design folders and code can be found here:
https://github.com/RU09342-F18/intro-to-embedded-final-project-theradac.git

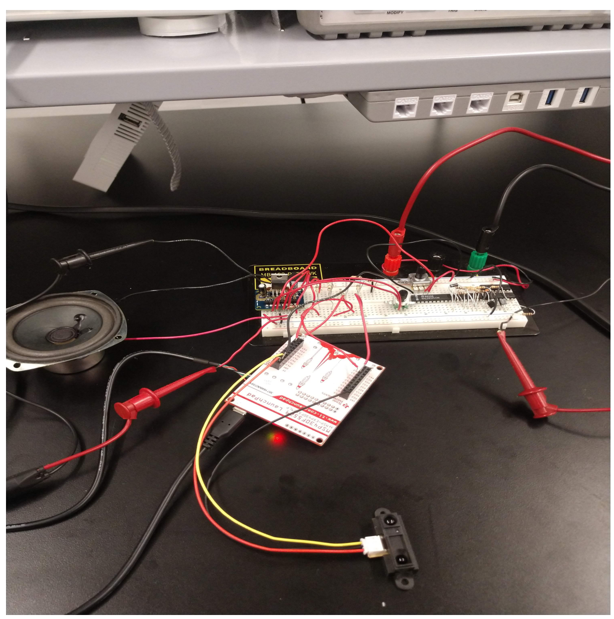
1.4 Simple Block Diagram

Figure 1: Simple Block Diagram



1.5 Board Image

Figure 2: Image of Board

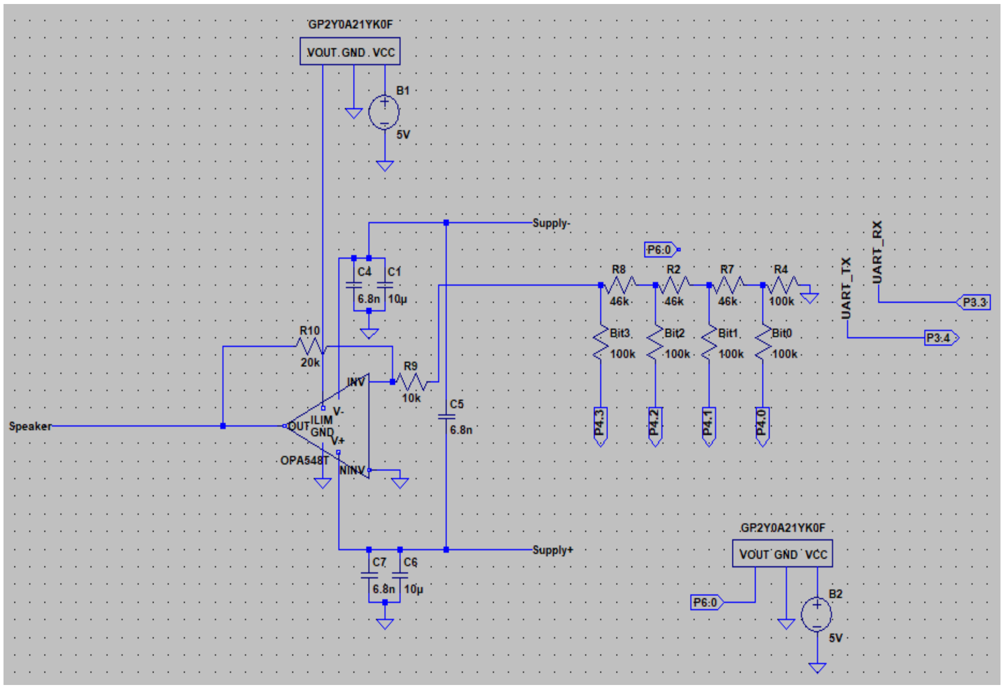


2 System Description

The problem to be solved is to create analog waveforms from digital signals, and control the frequency of these waveforms via an IR distance sensor. Along with creating waveforms, the volume of a speaker should be controlled via a second IR sensor. Traditionally, a Theremin is controlled via an oscillator circuit and uses an antenna and the human body to change a capacitance, which changes the frequency of oscillation. The frequency is then sent to a speaker, which plays sound. As stated earlier, instead of capacitance, the frequency is changed depending on the distance of a hand from the sensor, and instead of an oscillator circuit, the waveforms are made by using code to create the waveforms, and sending the output of the MSP430F5529 to a 4-bit R-2R ladder DAC (digital to analog converter). Originally, an IC DAC was to be used, but was not working properly, so the simpler and cheaper R-2R ladder DAC was used in its place. While using an R-2R ladder DAC, the output needs to be buffered if it's going to be connected to a load, especially a heavy load like a speaker, or else the output voltage will collapse. To accomplish this, a breakout board using the OPA548T power op-amp was set up in an inverting amplifier configuration to drive the speaker. To control volume, one distance sensor was connected to the current limit input of the OPA548T, so it would act as a voltage source with current control. To actually use the distance sensors, the internal ADC (analog to digital converter) of the MSP430F5529 was used to convert the analog voltage output of the sensor, into a digital signal usable by the board. The output characteristics of the distance sensor is non-linear, so the behavior was linearized in the code. Due to the range of the sensor being 4cm to 80cm, and the voltage at 4cm being inaccurate, the IR sensor was limited to a minimum distance of 10cm, and a maximum distance of 40cm in the code. The 40cm distance was also coded with some wiggle room, so the user doesn't have to hold their hand at exactly 40cm to hear the lowest frequencies. Finally, some control of the type of waveform was desired, along with the use of a communication protocol, so UART was used to send a character from a laptop, or potentially another board to the MSP430F5529 which would set the type of waveform. In this system, the selections were ”q” for a square wave, ”s” for a sine wave, ”r” for a ramp or sawtooth wave, and ”t” for a triangle wave, with the square wave being the default.

2.1 Detailed Block Diagram

Figure 3: Detailed Block Diagram



2.2 Highlighted Devices

The Devices used for this project were:

- MSP430F5529 for wave generation, UART communication, and as an ADC
- OPA548T breakout board to drive speaker
- GP2Y0A21YK0F IR distance sensor to change frequencies and volume
- R-2R ladder DAC to use digital output of the MSP to generate analog waveforms

2.3 MSP430F5529

The MSP430F5529 was used to create digital signals which would be converted to analog waveforms, converting the analog output of an IR distance sensor into usable digital values, and for UART communication between the board and laptop to change the type of waveform. To create different waveforms, code was written for each individual one, where the square wave just turned the pins on and off, the ramp counted up to 15, then reset to 0, the triangle wave counted up to 15, then counted back down to 0, and the sine wave used a look-up table with 8 values, with a max value of 15. The output of pins 4.0 to 4.3 were put into an R-2R ladder DAC, which converted the digital outputs into analog waveforms. The internal ADC converted the analog voltage output by the IR sensor into usable digital values, and the value of the distance was used to change the value of the CCR0 register, which changes the frequency of the Timer A interrupt, which changes the frequency of the waveforms.

2.4 OPA548T Breakout Board

The OPA548T was used to drive a speaker on the output without having the voltage from the R-2R ladder collapse due to driving a large load. The op-amp was used in an inverting amplifier setup to ensure that enough voltage should be across the speaker. The current limiting input was connected to the output of one of the distance sensors to limit the amount of current delivered to the speaker, thus changing the volume. The resistor values used were a 2k feedback resistor, with a 1k resistor connected to the inverting input of the amplifier, which means the gain of the configuration was -2.

2.5 GP2Y0A21YK0F IR Distance Sensor

The IR distance sensors were used to both change the frequency of the output waveform, and change the volume. The supply voltage of the sensor was 5 volts for both, and the maximum voltage output of the sensor was around 3 volts. The frequency controlling sensor was connected to pin 6.0 of the MSP430F5529, which is the ADC input port. The volume controlling sensor was connected to the current limit input of the OPA548T which directly controlled the maximum current that could be output by the op-amp. The distance sensors themselves were already connected to a provided PCB which took care of any filtering of the output signals.

2.6 R-2R Ladder Digital to Analog Converter

The R-2R ladder was used to convert the digital output of the MSP430F5529 pins 4.0 to 4.3 to analog voltages which could be used to drive a speaker, and were able to be viewed on an oscilloscope. The output of the DAC cannot be connected to a large load such as a speaker due to the load causing the output voltage to collapse to 0, which is why it was connected to the OPA548T, which could provide plenty of current to drive a speaker, and isolates the DAC from the speaker. The value for 2R was 100k ohms, and because there were no 50k ohm resistors, 46k ohm resistors were used, which was close enough. The original plan was to use a dedicated DAC IC, but the device proved too difficult to use, so a simple DAC was used in its place. The simple DAC is also much cheaper than a dedicated IC.

3 SYSTEM DESIGN THEORY

Within this project, the micro-controller was used as the brains for a distance controlled Theremin. A traditional Theremin uses an oscillator circuit connected to an op-amp to create sounds, where the frequencies change due to capacitance, where the Theremin in this project changes frequencies based on the distance of a hand from a sensor. Most Theremins also have volume control via hand distance, so a second distance sensor was connected to the current limiting pin of an OPA548T to change the amount of current provided to a speaker. To use the sensors, an ADC must be implemented in the system to convert the output of the sensors into values which can be used by the micro-controller. To create analog waveforms, a DAC must be implemented to convert the digital output of the micro-controller into analog signals. The op-amp was also used to isolate the DAc from the large load of a speaker, because the voltage output of the DAC will collapse if not isolated from the load. To change the output waveform, UART was used to communicate to the board, and send a character which corresponded to a waveform. The micro-controller can in theory be controlled by another micro-controller to change the type of waveform. The waveforms themselves have very visible steps, due to the DAC only being 4-bits due to space constraints on the breadboard. This issue can be solved by adding a low-pass filter to smooth out the steps, but this was not used in the design due to it changing the pure square wave output into a triangle wave.

3.1 Design Requirements

The requirements for this project were very flexible, and students were allowed to choose any project they would like, as long as it uses one of the MSP development boards, used sensors, and used a communication protocol. In this case, the requirements were to create analog waveforms, and control the frequency of these waveforms, as well as controlling the volume output from a speaker. To control the frequency and volume, IR distance sensors were used, and for the frequency sensor, it had to be input to the MSP430F5529. To read the distance detected by the sensor, the voltage output of the IR sensor was connected to the on board 12-bit ADC of the

MSP430, ADC12. the ADC converts the analog voltage into a digital signal, usable by the MSP430, and that voltage was converted to a distance through arithmetic on the board. The ADC continuously converted the values by toggling the enable sampling and conversion register via ADC interrupts. To create different waveforms, code was written for each individual one, where the square wave just turned the pins on and off, the ramp counted up to 15, then reset to 0, the triangle wave counted up to 15, then counted back down to 0, and the sine wave used a look-up table with 8 values, with a max value of 15. To actually count within this code, Timer A interrupts were used, where the timer was set to up mode, used SMCLK as the clock source, and CCR0 would trigger interrupts. The code worked by whenever an interrupt fired, depending on the selected waveform, the loops which created the waveforms would increment. Due to the code causing slowdown, SMCLK was set to 25 MHz, which caused the maximum frequency to be around 6Khz, which was only achievable by the square wave, due to its code being the fastest. The maximum frequency set was 1KHz, as it was achievable by all of the waveforms. To actually output these waveforms, pins 4.0-4.3 were connected to a 4-bit R-2R ladder DAC to convert the digial outputs into analog audio. Due to the property that if a R-2R ladder dac is connected to a large load, the voltage will collapse, the output of the DAC was connected to the input of an inverting op-amp amplifier to isolate the DAC, which allowed for a large load to be driven on the output. To control volume, the output of the other IR distance sensor was connected to the current limit input of the op-amp to control the amount of current provided to the speaker, thus changing the volume. The op-amp was configured to have a gain of -2 to try to counteract any voltage collapse caused by the large load.

4 Getting Started/How to use the device

Subsection 4.1 will go over how to connect hardware components, and subsection 4.2 will go over how to set up the required software.

4.1 Hardware Setup

First, view the detailed block diagram in order to see how to properly set up the circuitry on a breadboard. A power supply that can provide at least 15V at 2A is required for proper op-amp operation. The resistor values, component models, and any extra voltages are all listed in the detailed block diagram. Be sure when setting up to avoid accidentally plugging any pins into the supply rails, at risk of one damaging or destroying the board. Plug the voltage output of the distance sensor into pin 6.0 of the MSP430, and connect the other sensor to the current limit input of the op-amp. Plug both the MSP430 power cable and the UART cable into USB ports, connect the white wire of the UART cable to pin 3.3, connect the green cable to pin 3.4, and connect the black cable to any ground pin on the board. DO NOT connect the red cable, for it is a 5V cable and can damage or destroy the board.

4.2 Software Setup

First, be sure both code composer studio and Realterm are installed on the computer. In code composer studio, once the board is connected via the USB, debug the project, proceed on the low power mode message, and once complete, click the resume button on the top bar to start debugging. In Realterm, under the ports tab, be sure the baud rate is set to 9600, and make sure the COM port of the UART cable is selected. To find the COM port, open up device manager (if on windows) and expand the ports tab. The COM port of the UART cable should be labelled. Once set up, navigate to the display tab and check off ASCII and navigate to the send tab to send either a ”s”, ”q”, ”r”, or ”t” to change the waveform type.

5 Getting Started Software/Firmware

This section will go into more detail on how to setup the software, mainly Realterm setup.

5.1 Communicating with the Device

To communicate with the device, the program Realterm, and a UART cable are required. To start, start up Realterm as shown in the Software Set up subsection. Once navigated to the text box, one can send a predesignated character to change waveforms. Send waveform designations as a lowercase character that’s either an ”s”, ”q”, ”r”, or ”t”.If an invalid character is input, the code will default to a square wave. Please only send one character at a time, or else the code will default to a square wave. Once typed in, click the send button, which will send the code to the board, and set the output waveform type.

6 Test Setup

To setup the system for test, follow all guidelines in the hardware and software getting started sections. Once Realterm is open and ready to send data, turn on the power supply and view the following waveforms:

- Square wave with no load
- Ramp wave with no load
- Triangle wave with no load
- Sine wave with no load
- Square wave loaded by speaker
- Ramp wave loaded by speaker
- Triangle wave loaded by speaker
- Sine wave loaded by speaker

6.1 Test Data

The following subsection includes pictures of the oscilloscope connected to the output of the op-amp, which shows how the waveforms look.

Figure 4: Square with no Load

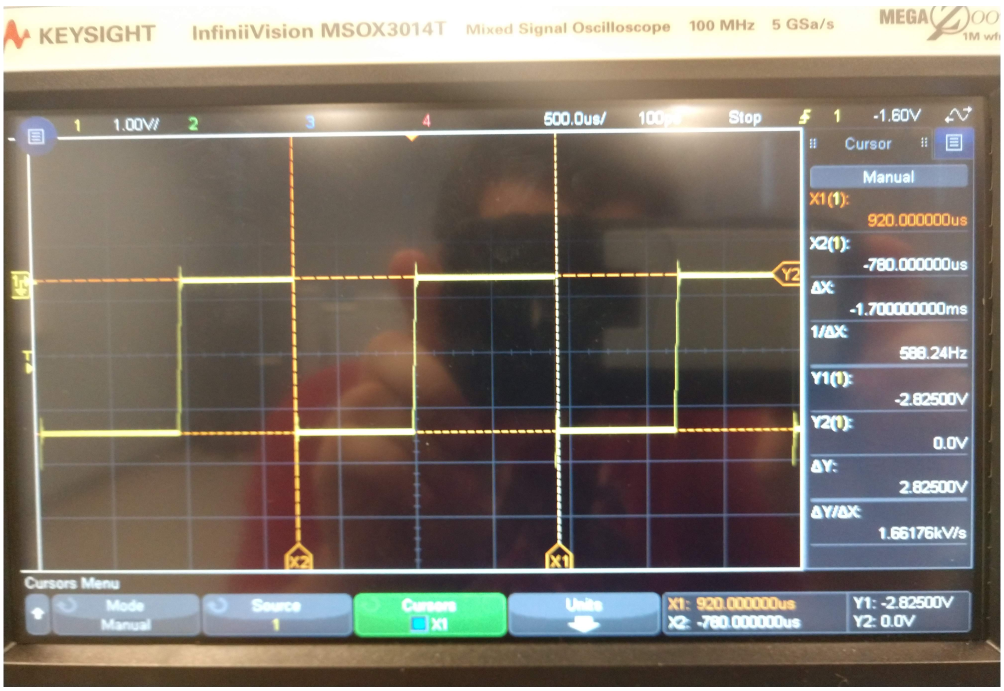


Figure 5: Ramp with no Load

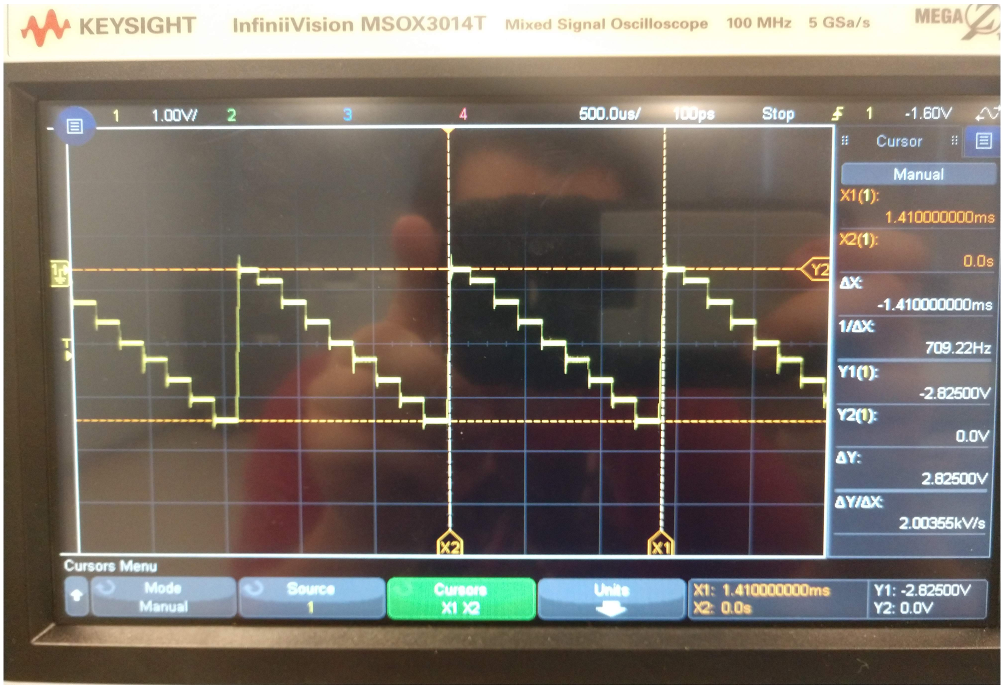


Figure 6: Triangle with no Load

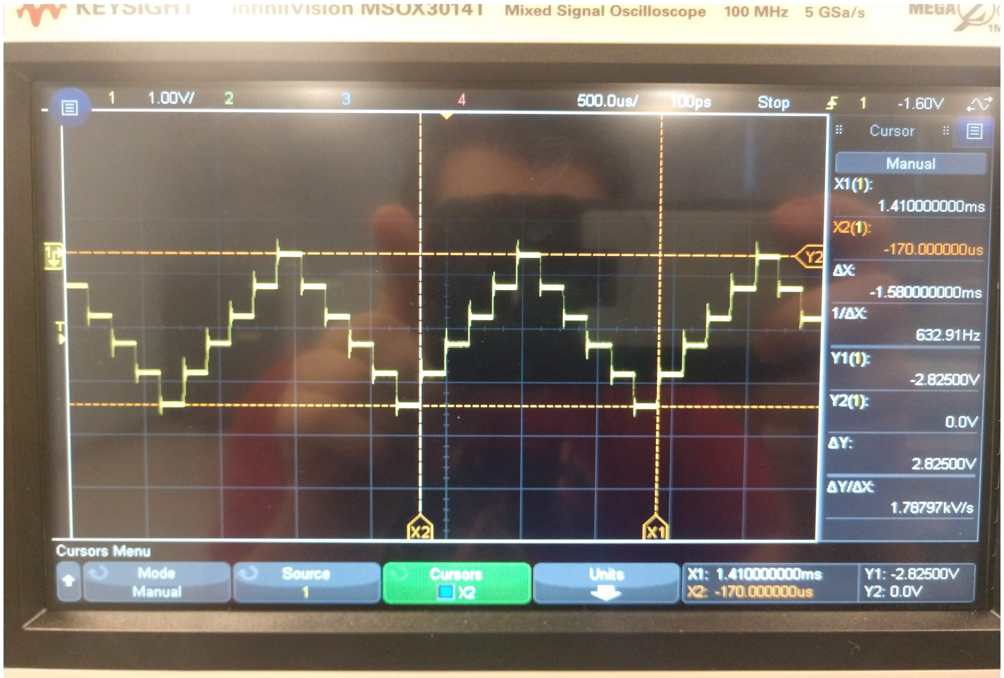


Figure 7: Sine with no Load

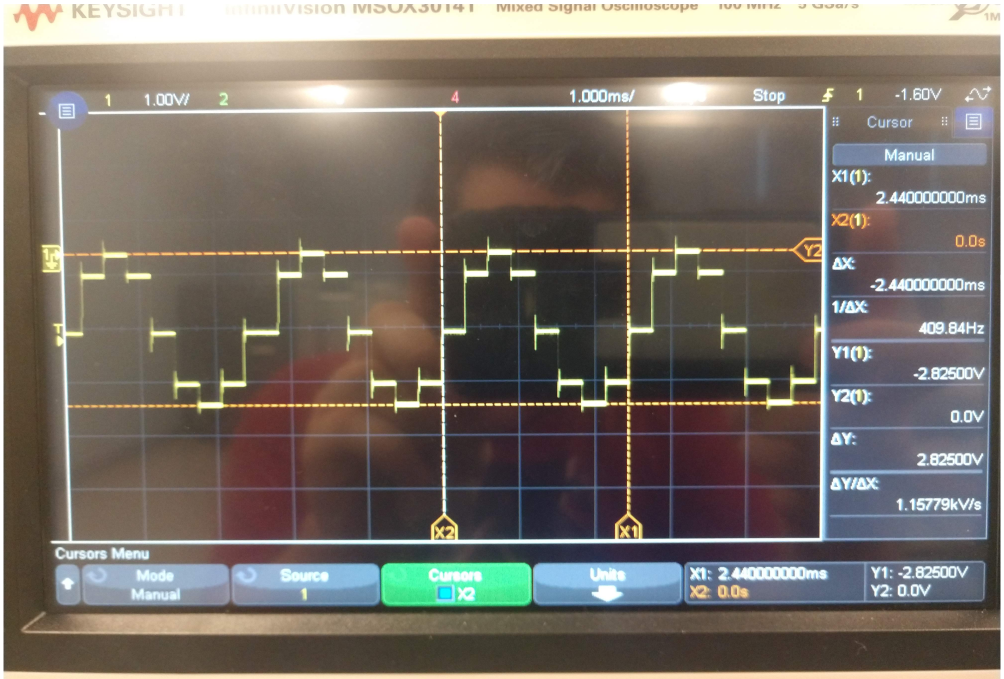


Figure 8: Square with Load

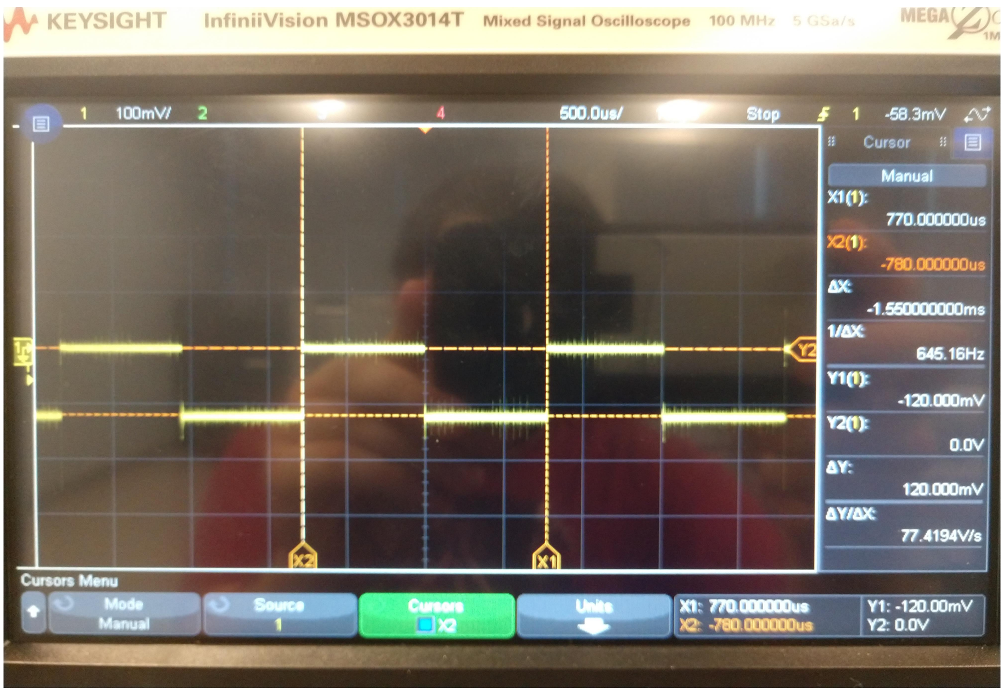


Figure 9: Ramp with Load

Figure 10: Triangle with Load

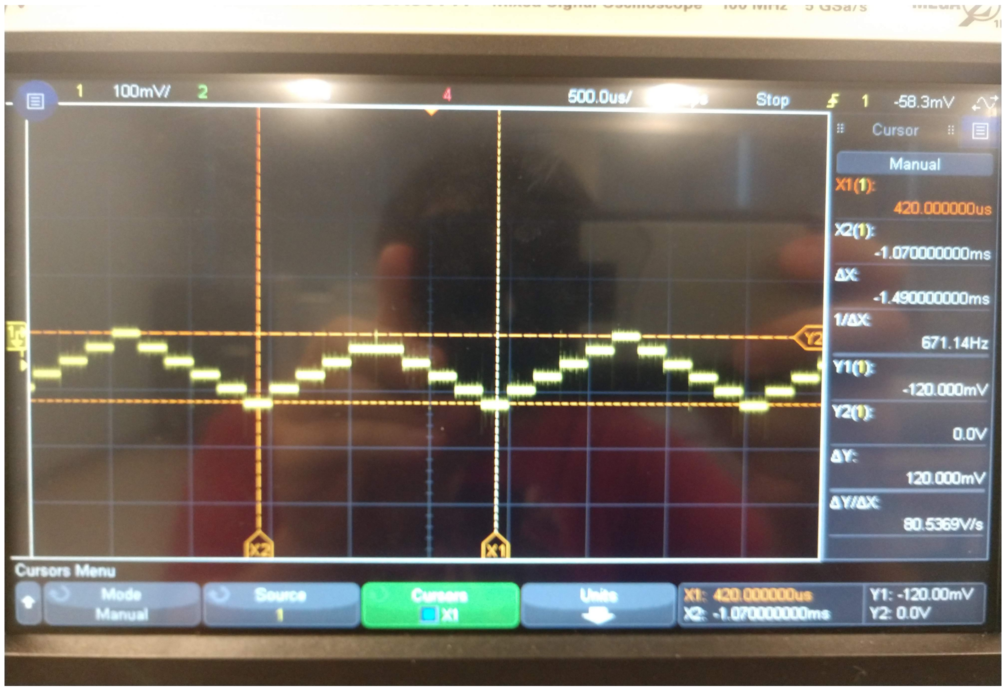
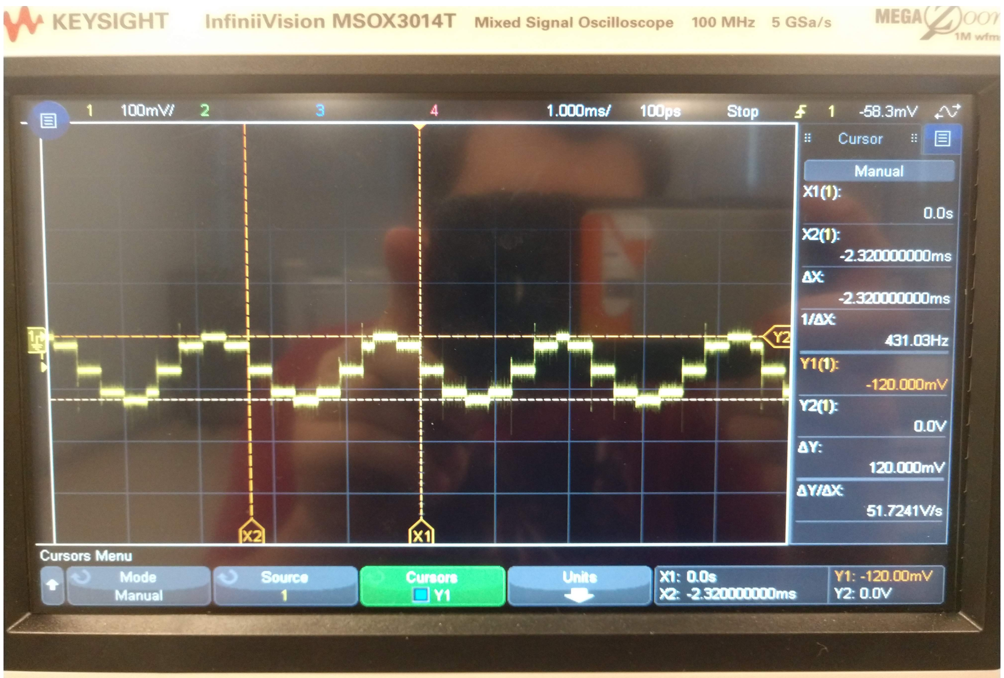


Figure 11: Sine with Load



As shown, even with the op amp in an inverting amplifying setup with a gain of -2, the voltage output still collapsed due to the large load of the speaker. A potential solution could be to use one op-amp as a buffer between the amplifier and the DAC to ensure the DAC is isolated from the load. The speaker also added noise to the output waveforms, but the overall shape of them is the same.

7 Design Files

Design Files can be found in the github repository located at:
https://github.com/RU09342-F18/intro-to-embedded-final-project-theradac.git

7.1 Schematics

The Schematic of the OPA548T breakout board and the R-2R ladder DAC can be found in the detailed block diagram.

7.2 Bill of Materials

Used materials include 1 4 ohm 3 watt speaker, the MSP430F5529, 1 OPA548T power op-amp, 5 100k ohm 5 percent tolerance resistors, 3 46k ohm 5 percent tolerance resistor, 1 2k ohm 5 percent tolerance resistor, 1 1k ohm 5 percent tolerance resistor, 2 6.8 nF capacitors, 2 10uF capacitors, 2 GP2Y0A21YK0F IR distance sensors, a UART cable, and a breadboard.