Name of the Project: Synthesizer
Group Member's Name: Aidan Sharpe, Elise Heim
Date: 5/01/23
Course Name: Introduction to Embedded Systems

Contribution Chart

	Aidan	Elise
Initial Research	X	X
Block Diagram and UML	X	X
PCB Design	X	X
Soldering	X	X
Coding	X	X
Analyzing Results	X	X
Project Related Research for Report Writing	X	X
Writing Report	X	X
Making Presentation Slides	X	X

Abstract

Our project consists of a synthesizer that plays notes from C4 to B5 using buttons. The MSP430 was to provide square, triangle, and sawtooth waves. The audio would be comprised of a combination of these waves, which could be altered. The volume would be adjustable.

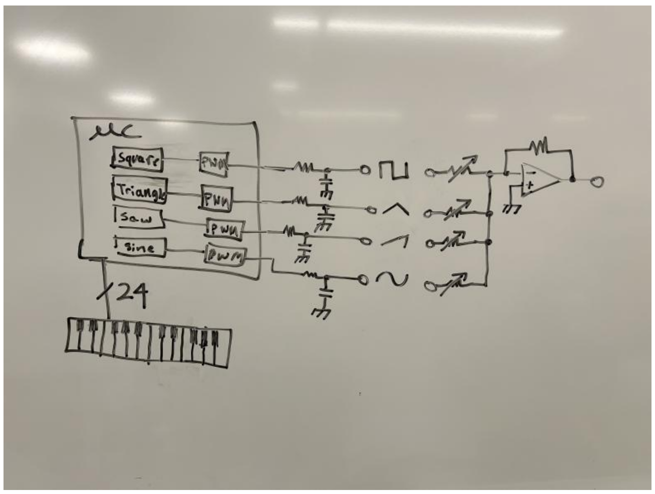
Background and Introduction

Four waveforms are generated by the MSP430. The sound from the synthesizer is a mixing and amplification of these waveforms. Each waveform provides a different sound. Square waves contain odd harmonics, which mean the odd intervals (3rd, 5th, 7th, etc). This means that square waves have a kind of nasally sound to them. Sawtooth waves contain a lot of even harmonics. They have a tense quality. Triangle waves contain only odd harmonics, but they are weaker than square waves'. These waves have a reed-like sound. When any of these distinct waveforms are combined, they create unique sounds that exhibit the qualities of the waves used to create them. The result is some really groovy music.

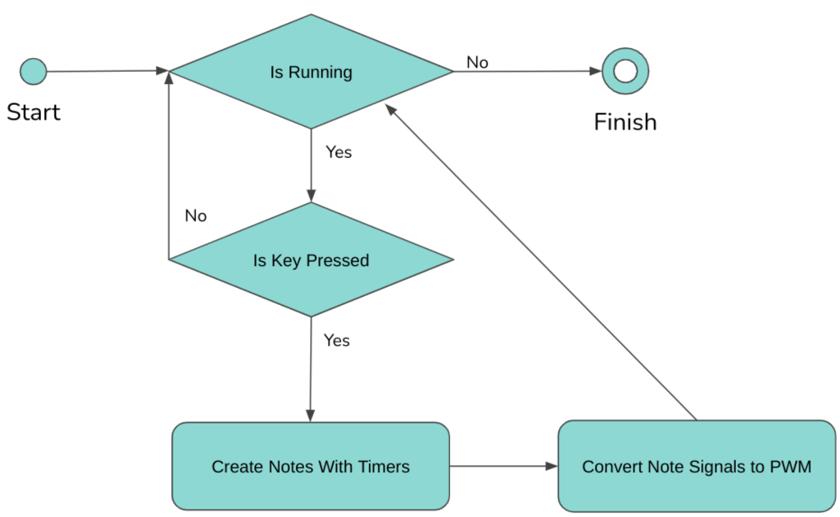
Methodology and System Design

Twenty-four buttons arranged like piano keys are connected to 24 of the MSP430's GPIO pins. These buttons will play two octaves worth of notes. When a button is pressed, it sets the input value of its corresponding pin to logic 1. In turn, a predefined frequency is selected. Using the onboard timers, 3 concurrent waveforms—square, triangle, and sawtooth waves—are generated and encoded as PWM outputs to be outputted as digital signals. These digital outputs are then fed through first order low pass filters to convert PWM signals into analog signals. The signals are then fed through a rail-to-rail voltage follower to isolate the filtered signals, thereby preparing them for mixing. The signals are mixed with a power amplifier constructed in a summing amplifier topology. Finally, the output of the power amplifier is connected to a speaker driver.

Functional Block Diagram



UML Activity Diagram

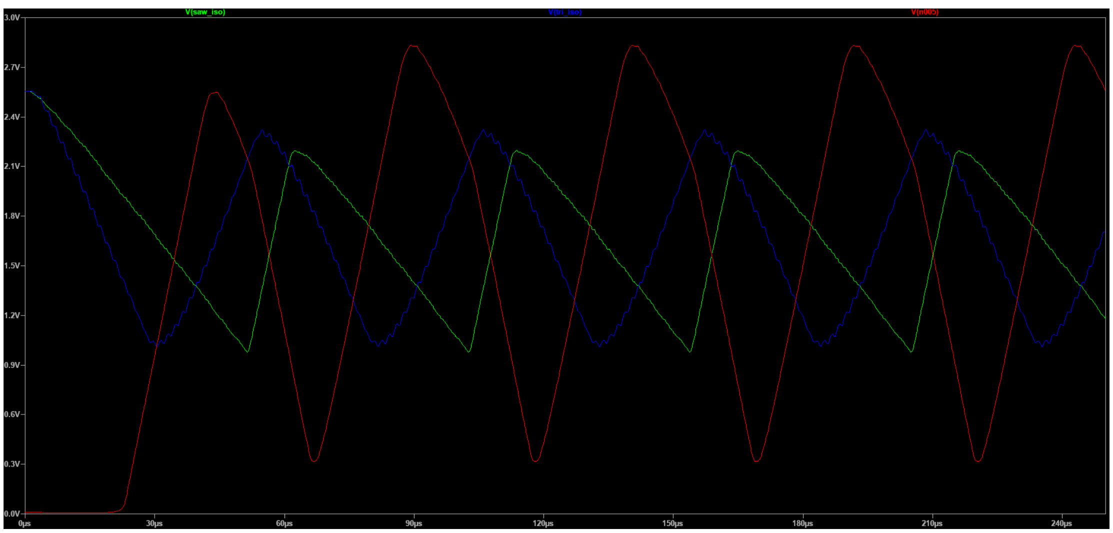


Parts List and Design Cost

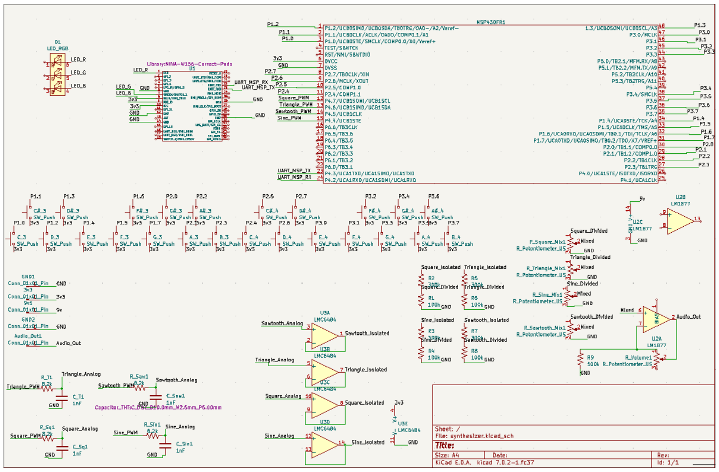
Item	Quantity	Unit Price	Notes
Through-hole resistors	13	~$0.10	8.4kΩ
Through-hole ceramic capacitors	4	~$0.10	1 nF
LMC660CN	1	$1.13	Rail to Rail Op Amp
LM1877N	1	$1.84	Power Audio Amp
Buttons	25	~$0.10	Keys on Keyboard
Potentiometer (100k)	5	~$0.10	To tune sound channels and volume
MSP430FR2355	1	$4.19	Provided from IES
Adafruit speaker	1	$3.60	Found on floor after ProfHacks
PCB	1	$1.00	$5 for 5
HW 131 Power Supply	1	$2.04	Supplies 3.3v and 9v DC

Experimental Set Up and Results

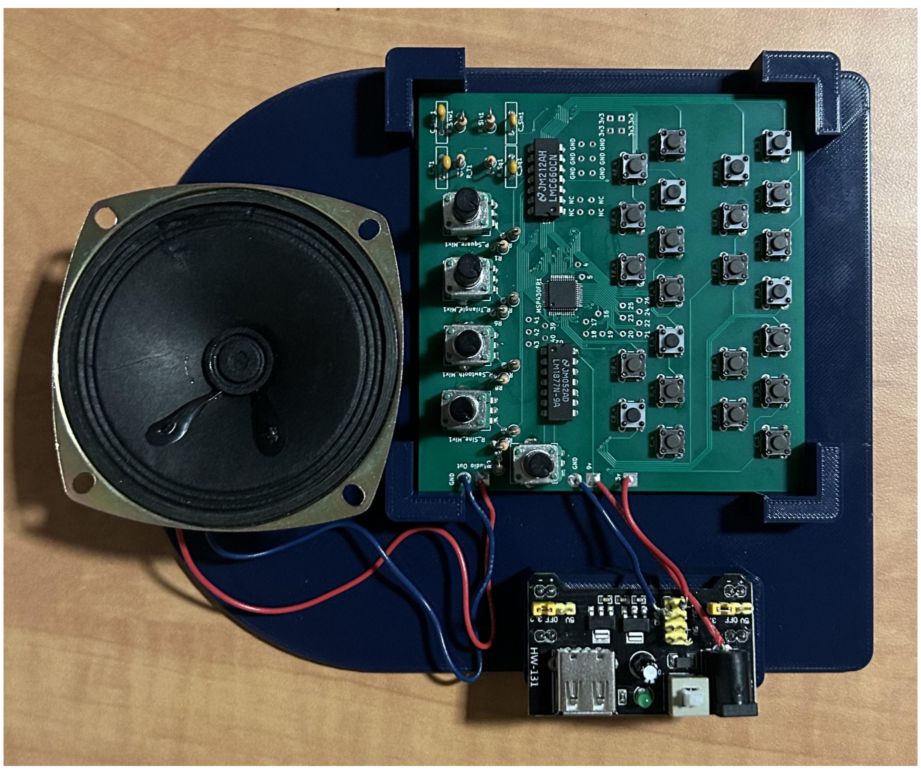
Triangle and sawtooth PWM outputs were simulated in LTspice. The signals were passed through identical low pass filters, and their outputs are shown in the image below.



Schematic



Soldered PCB connection with MSP430



Results from ThingSpeak

As part of our original goals, we planned to include a bluetooth audio out feature, rather than a web connection, for our connectivity component. However, since so much programming time was spent just trying to generate waveforms, there was not enough time at the end to introduce the bluetooth feature.

Conclusion/Future Work

Unfortunately getting waveforms to generate proved to be much more difficult than anticipated. The method that we originally intended to use was flawed. Originally, the plan was to use internal timers to count up at a known rate to determine the output value at any given time. This output value would then be processed rapidly and regularly using an additional “sampling” timer. However, despite the sampling timer running at 1 MHz, the maximum speed that we could get sampling interrupts to be handled was about 1.2 kHz. This frequency was much too low as our low pass filters were tuned for a cutoff frequency of 20 kHz, which would produce a noisy signal. Additionally, 1.2 kHz was dangerously close to some of our desired output frequencies, which would lead to output noise due to sampling being too infrequent.

These unforeseen complications to our waveform generation methodology meant that certain features had to be sacrificed for purposes of presenting. These features include sine wave generation and bluetooth audio output. In the future, it is planned to get the bluetooth output working in order to connect to a wireless speaker. We also plan to program the MSP430 to generate a sine wave, to increase the amount of unique sounds that could be generated.

References

What is a synthesizer, how it works, types and uses (higherhz.com)
Basic Waveforms - YouTube

Appendix

All project files can be found here.