Making a piano/synthesizer with 5 modes and frequency adjustments using ADC and capacitive touch sensing
PROJECT 4 Final Write UP

Mkhanyisi Gamedze
CS342 FALL 2018

Pitch

Abstract

This project extrapolates on a failed initial idea I had for my project 3 proposal, where I aimed to use capacitive touch sensing to make a more reactive piano, than the slow keyboard keys I had to enter via the serial port. After lab 11 however, I had a much better understanding on the technical depth relevant register requirements to achieve that goal. Using the ADC sensing capabilities from project 3, coupled with copper tape in our kit, that was enough to get me started creating a synthesizer piano instrument. Using the keyboard keys, an input is used to switch the frequency modes and also set the ADC flag to adjust the frequency when pitch is sustained. The full project demonstrates these capabilities and the inspiration behind it was the ability to play around with sounds and understand counters more, as those gave me a hard time in lab 4 and 5. My model is highly sensitive to capacitive touch and this allowed ease for users when playing some songs of choice.

Resources

Parts list -- electronics

- Adafruit metro mini board
- 1 m copper wire
- 1 potentiometer ($10\ k\Omega$) limit
- Piezo piece
- Long breadboard Small Breadboard (for the joystick)
- Analog thumb joystick
- Solder table (tool)
- MPR121 capacitive touch controller

Helpful resources

- https://www.youtube.com/watch?v=npS2E4fLYe4 - gave me an understanding of how circuitry is possible within the context of such a class
- https://exploreembedded.com/wiki/Analog_JoyStick_with_Arduino - circuitry of joystick
- https://www.brainy-bits.com/arduino-joystick-tutorial/

- Lab 7 theremin code gave me the required outline for capturing an ADC input
- Lab 11 - Colby CS342
- https://github.com/chipsi007/LittleArduinoProjects/tree/master/playground/CapacitiveTouchOrgan
-

Rough sketch

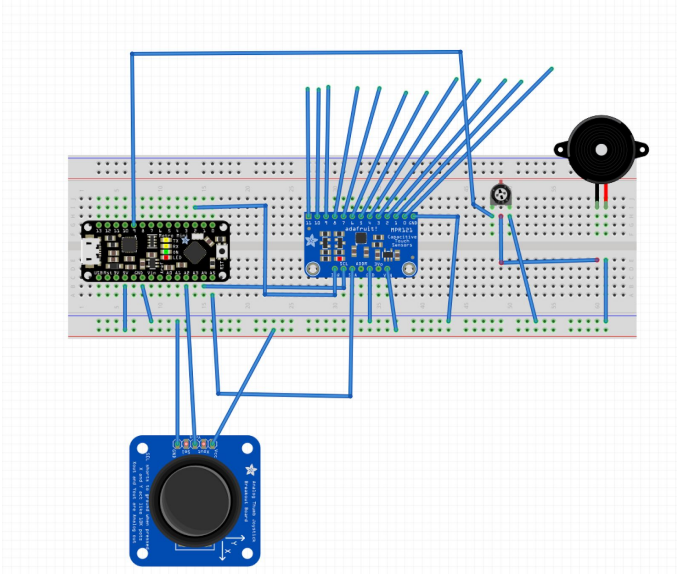

Fig 1: Circuit Design

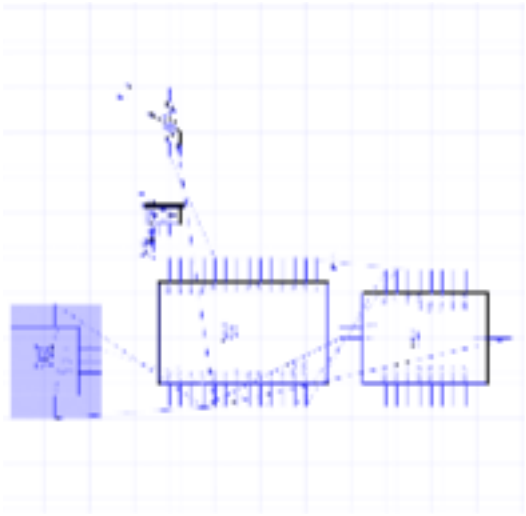

Fig 2: Schematic Diagram

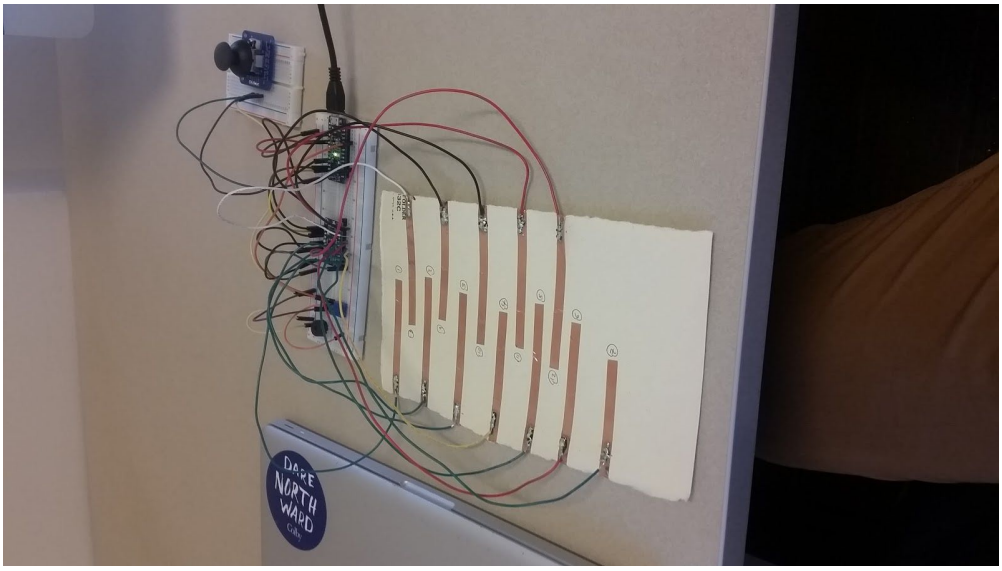

Fig 3: Actual Setup

NO INTERRUPTS BUTTONS USED, USART KEYS INSTEAD VIA USB PORT

Design Criteria & Procedure

I connected to PB1 as my only output for the sound device, my peizo. The peizo also had a potentiometer connected in series to adjust the volume. The main goal was to control sound and the MPR121 Capacitive touch sensor and Analog joystick served this purpose. The MPR121 was connected as highlighted above, with the SCL and SDA communication lines connected to Analog inputs 4 and 5. The IRQ was connected to Pad 2. The joystick was connected to analog input 2 and it served the purpose of adjusting the sound frequency with an offset. The USART library was used to read in input, as the letters ASDF represented 4 different octave modes and anything outside of that was thrown to the default mode.

The piano design's main goal was to replicate the sound frequencies in each octave, and my code represented octaves 2 to 5 , the random frequencies for the default out of mode. The capacitive touch sensors could only represent 12 frequencies at a time. The USART serial input generated interrupt to control the modes and also set the "sustain pitch" (received character "r") and the ADC flag "ADC" (receiver character "w"). The ADC flag only controls the frequency if the pitch is sustained. Case switch statements were used to control the mode switch and the MPR 121 h library was used to read in which capacitive touch sensor was triggered in order to play the corresponding tone. The analog joystick did not need characterization for my design, as the Y component voltage varied by a factor between 0 and 1024. This was accounted for in my code, when adjusting the frequency with

the offset. For efficiency in code, one of my functions was setFrequency which adjusted the oscillation frequency of the piezo to timer clock.

Test Results

My system was highly sensitive to capacitive touch, allowing itself to replicate a real musical device with the tones it played. Using the USART character input to switch modes slowed down the system and if more capacitive touch sensors were to be connected, these could have been used for the switches. The sustain pitch mode worked perfectly, and it was highly as reactive when switching notes played. The sounds were smooth. For sustained pitches, the frequency offset adjusted by the ADC input resulted in a squeaky sound as expected.

Discussion

My system performed as expected for all my functions, successfully generating sounds and switching modes when relevant serial inputs were added. The controls were able to sustain pitches well and adjust the played frequency using the analog joystick. Further improvements on the project could include making a more reactive system using only capacitive touch sensors instead of USART key inputs, for a full piano/synthesizer device. Using capacitive touch sensing for controls as well is further improvement, which which would require us to make use fully of the analog inputs to read in the values.

References

https://en.wikipedia.org/wiki/Piano_key_frequencies