Trimodal Musical Instrument
Project Report

Student: Saad Saeed
Instructor: Dr. Ravi Shankar
Course: CDA 3331 Spring 2017

Introduction

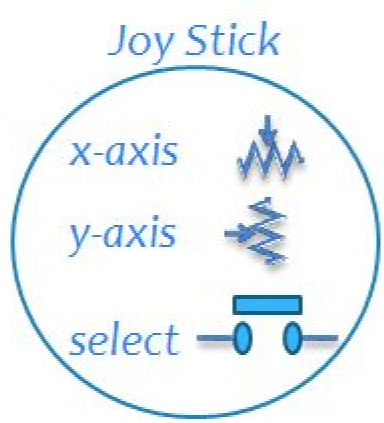
MSP430 is an impressive microcontroller. We have used MSP430G2553 version of this controller which came mounted in a LaunchPad development kit. We were asked to propose a project using a 40-pin Education Booster Pack MKII. II initially proposed a project that would use the buzzer the Booster Pack and generate varying frequency tone based on the X-position of the joy-stick on the Booster Pack.
Later, I decided to expand upon this idea and introduced several other features using other resources available on the Booster Pack.

Resources of Booster Pack Used

The resources used in this project were the following:

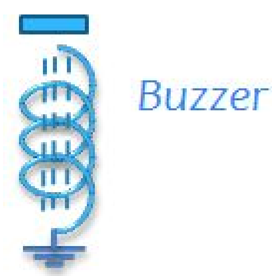
- X-axis and y-axis of the joystick
- The “select” button of the joystick to play Pre-programmed mini-song
- X-axis and y-axis of the accelerometer(z-axis not used)
- Buzzer, as the device to play tones.
- Upper Button to enable/disable Joystick Instrument.
- Lower Button to enable/disable Accelerometer Instrument.

Joystick

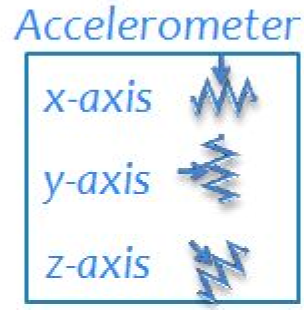


The joystick of the booster pack is composed of two variable resistors, that act as potentiometers. For instance, moving the joystick from left to right varies the voltage on J.5 pin between 0 and VCC. In other words joystick converts position into a voltage. Pressing joystick enables the “select” button that can be used as a binary input. The schematic Model of Joystick is shown in the diagram.

Buzzer



The buzzer’s input (incorrectly labeled as Buzzer out) in the scant insert in the kit. Buzzer output is digital, so it can be either low or high. To generate different tones, we have to bring the voltage up and down on the buzzer port periodically.

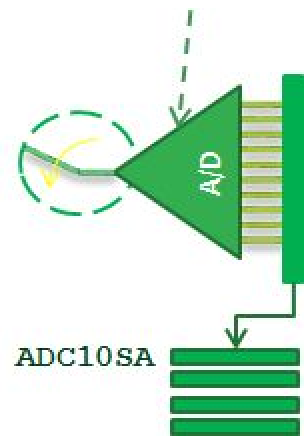


Accelerometer

The Accelerometer is a 3-axis analog accelerometer that measures g-forces. Moving the board along
the axes will change the analog signal generated by the accelerometer. Accelerometer can be visualized like a joystick with three potentiometers. We have used only the x-axis and y-axis.

Resources of LaunchPad Used

Here is a brief description of peripherals that I will use in my project.

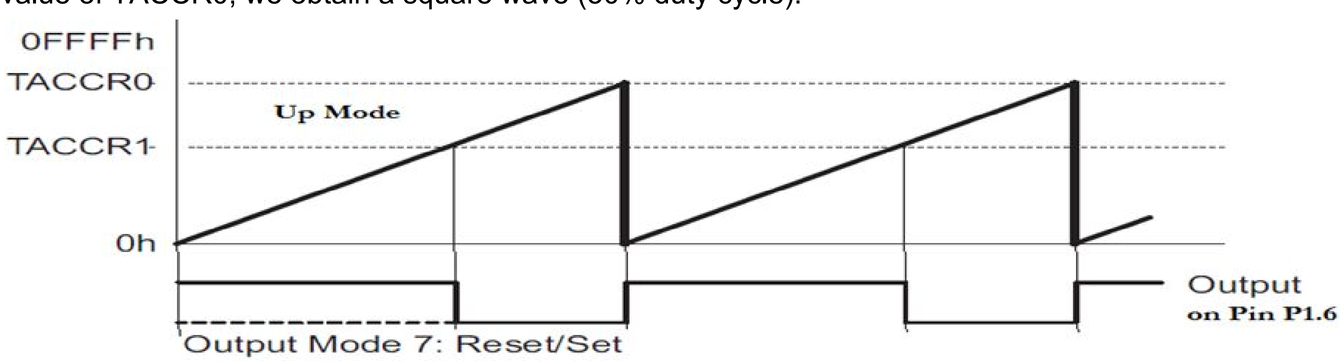


A/D Converter

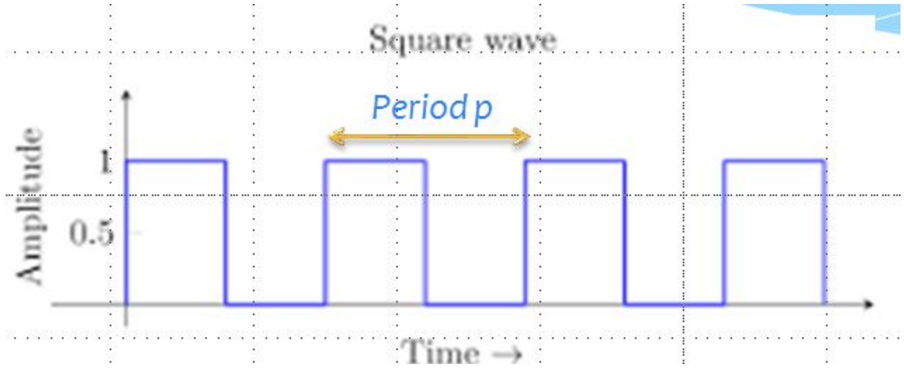
A/D converter is part of the the MSP430G2553 microcontroller. I used it in Sequential mode to convert four analog inputs (A0, A1, A2, A3) from x-axis and y-axis of both joystick and accelerometer. Direct transfer control was used to transfer sampled values to an array. A/D converts an analog value to a 10-bit digital value that is stored temporarily in the ADC10MEM register. ADC10MEM first stores the value of the current output of A/D. In sequential mode, these values have to be rapidly copied to another memory location before being overridden by the next conversion.Fortunately, DTC provides facility to do just that. Once ADC10SA is programmed to an address in memory, all results are copied sequentially in a array.

Timer A

We have programmed the timer to produce a square wave of a chosen period at the output pin P1.6. Two registers TACCR0 and TACCR1 are used to achieve that. The timer is used in up-mode, that means it counts up till count reaches the value stored in TACCR0. This determines our period *p*. However, while the counter is counting, as soon as the count equals TACCR1 (which is less than TACCR0) the output on the pin drops to a low voltage as shown below. If we set TACCR1 to half the value of TACCR0, we obtain a square wave (50% duty cycle).

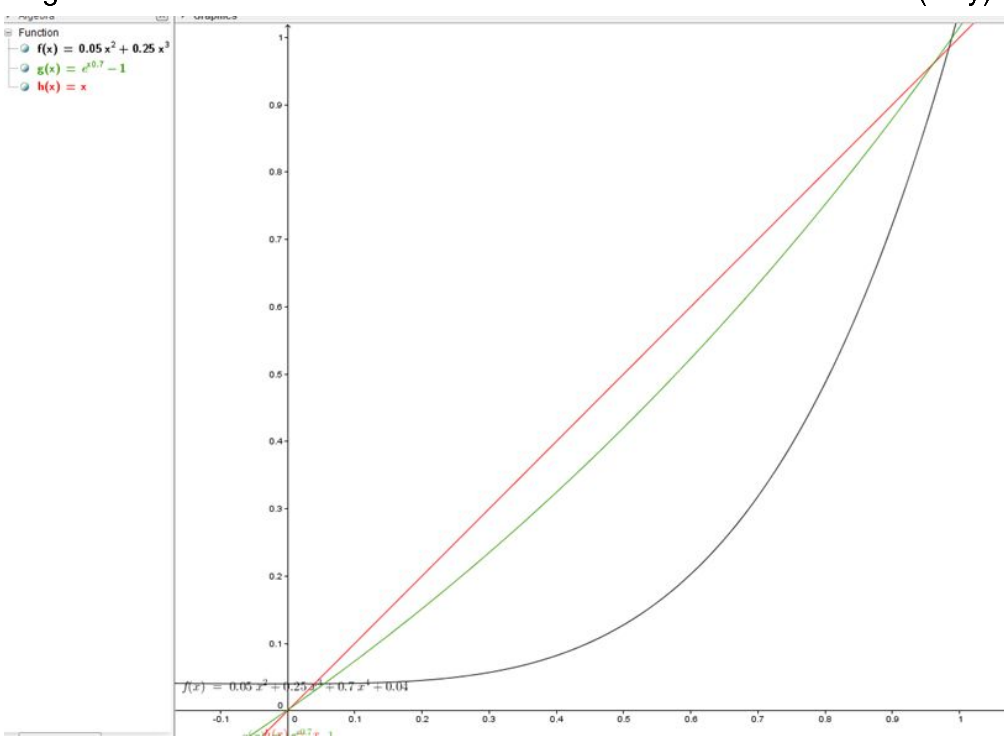


In our case, we have connected pin P1.6 directly to the buzzer to generate the tone corresponding to the frequency *1/p*.



Frequency Generation

If p is the period of the square wave applied to the buzzer, the frequency will be reciprocal of the period, i.e. Frequency = 1/p cycles/sec or Hz. I controlled the period using the variable resistance provided by the inputs (x-axis and y-axis of joystick or x-axis and y-axis of of accelerometer). For the joystick, to get one value, I simply took the average (x+y)/2 and adjusted the values so that they would fit within the range. For the accelerometer I subtracted a constant from the sum i.e. (x+y)-1200.



I did normalize the multiplier for my period within the range of 0 and 1. However,using this normalized number to directly modify the period od did not make it very interesting as most of the frequencies were jumbled up on the lower side of the scale due to the reciprocal 1/p relationship. I ventured into experimentation by trying different polynomials to make this mapping nonlinear. I settled with this function:

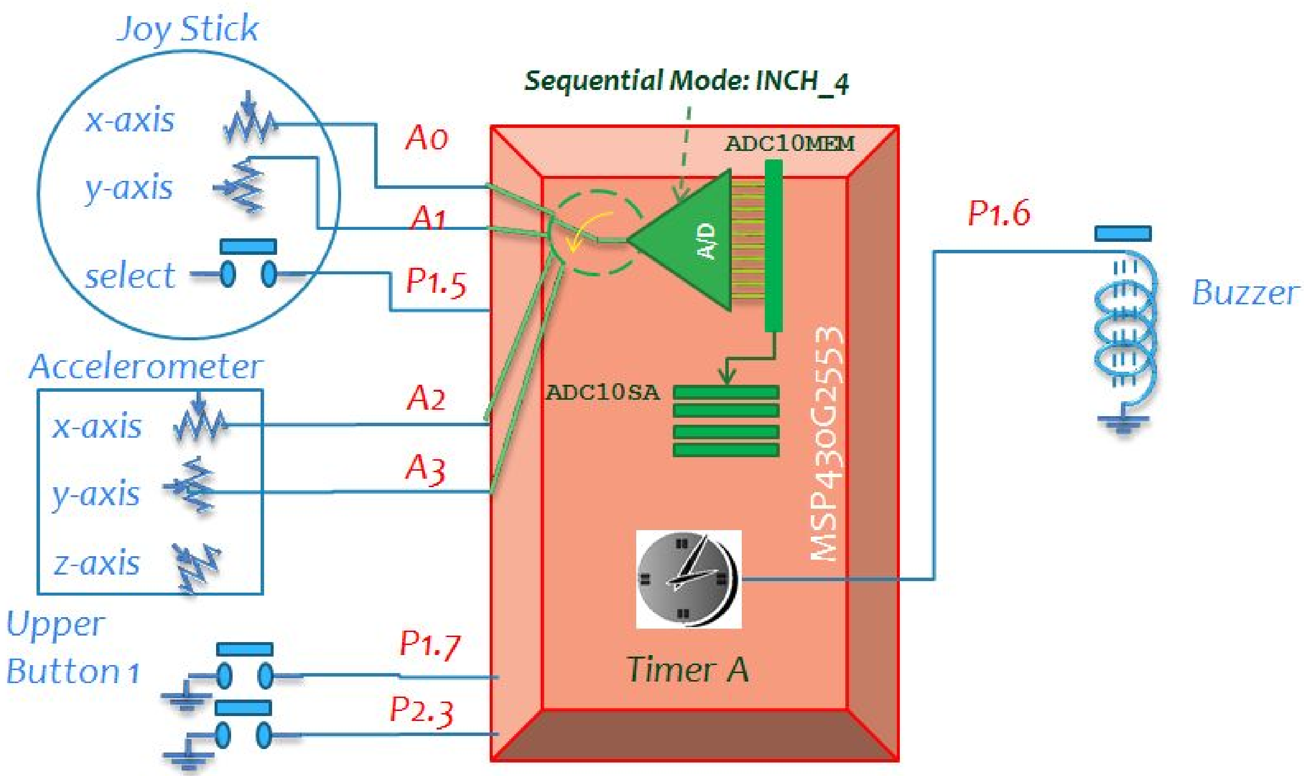
$$f(x) = 0.05x^2 + 0.25x^3 + 0.7x^4 + 0.04$$

(read x=d)

Where x is the normalized input and f(x) is the actual multiplier I used with the period.

Full Operation

The full operation of can be explained with the help of the following diagram.



A/D sequentially reads the four analog inputs and stores them in an array. To remove noise, we read five consecutive values and average the results .If the Upper Button has been pressed, then joystick X and Y values are averaged, normalized (say d), and the f(d) is multiplied with the maximum period to give us the corresponding tone on the buzzer. If the Upper Button is pressed again, the tone stops.

If the Lower Button is pressed, the inputs from X-axis and Y-axis of the accelerometer sumed and teh result is reduced by 1200. The result is used in a similar manner as above.

If the user presses the “select” button of the Joystick, all other sounds are stopped and a pre recorded mini-song is played.

Conclusion

It was fun to learn and combine many of the ideas we learned during the semester into a single project. I learned PWM in Lab 2 which I modified to change time period instead of duty cycle. I learned how to use ADC with a potentiometer in Lab 2, that I used several times here. I learned how to deal with switches in Lab 5 that I used to deal with the buttons. I learned how to use multiple channels of ADC in lab 6 that was very helpful in this project. This was a fun project to work with.