

PES Final Project Test Plan and Final Report

Musical Notes Player based on-board MMA8451Q Accelerometer over I2C

Project Functionality Matches the Proposal:

- Musical Tones Player plays tones based on horizontal plane (roll angle) using on-board MMA8451Q Accelerometer over I2C communication.
- User can move FRDM-KL25Z in horizontal angle (roll) from 0 to 180 degrees. (Reference angle: 0 degrees) and it gets detected.
- User can access the command processor through UART to display the current roll angle and to run all the tests. Terminate command works to come out of the command processor and to dive into the musical player application.
- LED indication based on angle measured.
- Different musical notes that are one second apart are played indefinitely in different angle ranges when user moves the KL25Z horizontally.
- To stop the musical player, user can lay down the board flat. And then reset to restart.
- Click for [Video Demo of the Functionality](#)

S.No.	Accelerometer Roll Angle Range	LED	Musical Notes
1	0° - 5°	Green	-
2	6° - 45°	White	A4, D5, D6
3	46° - 90°	Yellow	D4, E5, F5
4	91° - 135°	Magenta	E6, F6, G6
5	136° - 180°	Cyan	D4, D5, D6

Embedded Software Development - Technologies:

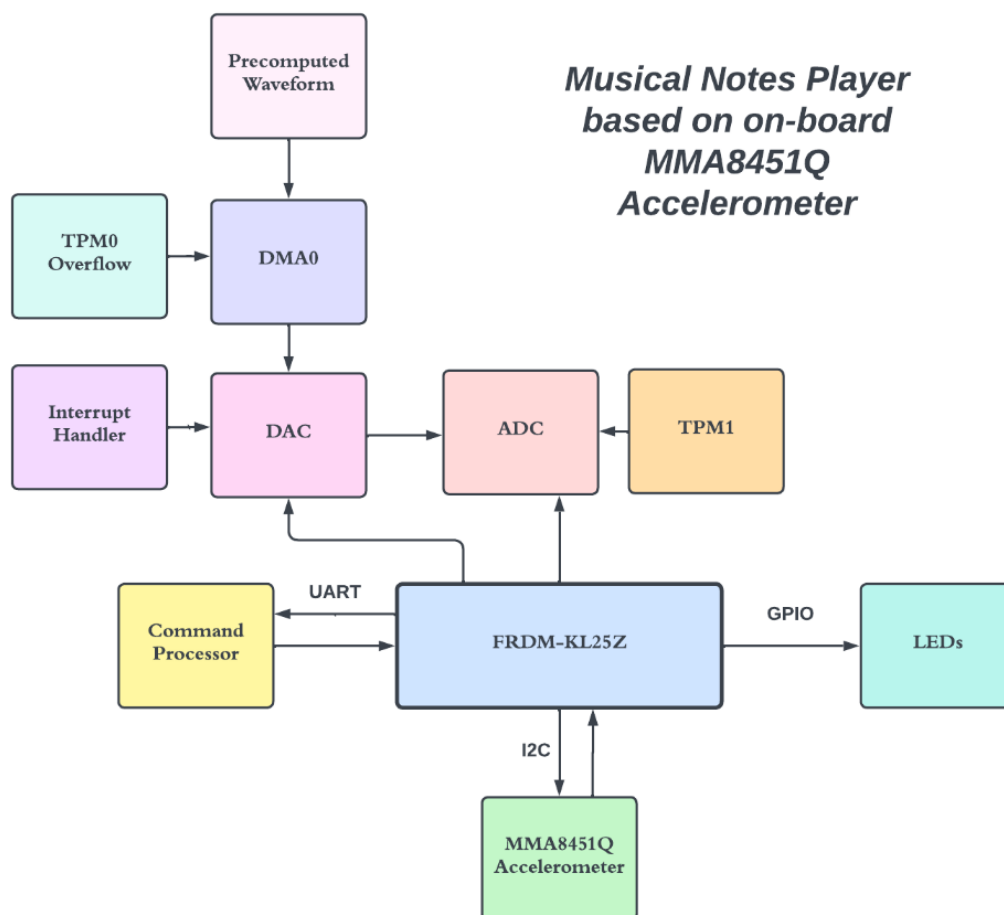
- **I2C** - Interfacing on-board MMA8451Q accelerometer over I2C communication.
- **Command Processing** – User Interface that gets input commands from the user and executes the corresponding command functionality.
- **UART** – Communication established between command processor and FRDM-KL25Z to send and receive data.
- **Interrupts**
- **GPIO lines** – LEDs configured to visually display the roll angle of the accelerometer on-board.
- **Timers /System Clock** – SysTick timer and clocks are used for delay functions.

- **Interrupts** – DAC, DMA interrupts.
- **Circular Buffer** – For UART transmission and reception
- **DMA**– Direct Memory Access copies from source to destination without involving CPU and it's much faster. It copies from tone_to_samples function to DAC.
- **ADC**– ADC is used to check the correctness of the generated sine wave musical tones.
- **DAC**– Generates sine wave musical tones.

Challenges Faced:

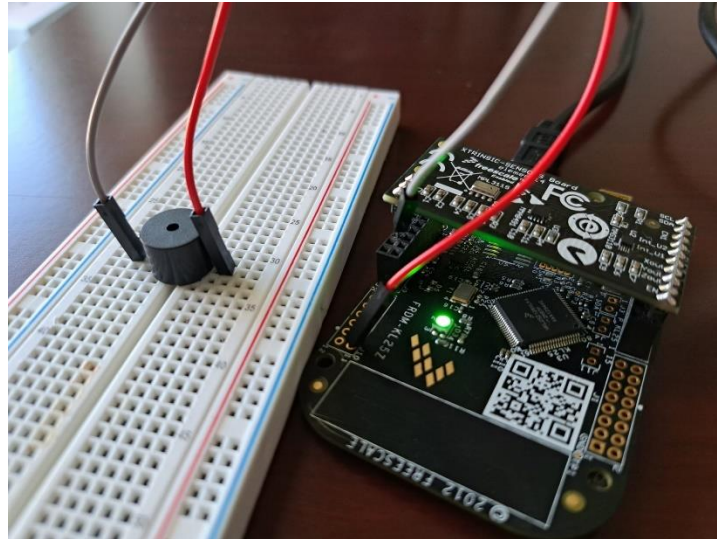
- Getting the DAC output without tearing at the wrap point when the tone changes.
- The musical tones stopped playing in between at random occasions. After a long debug session, I figured that if the printf statements are too long, it messes with the timer and application throws unreliable working. So, I shortened all the printf statements and reduced the number of statements.

Block Diagram:

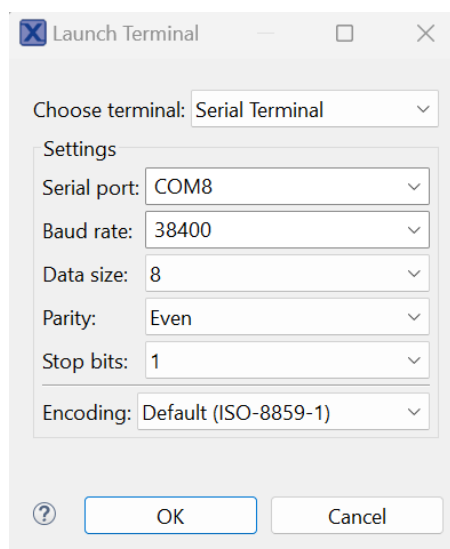


Additional Hardware Requirement Fulfilled:

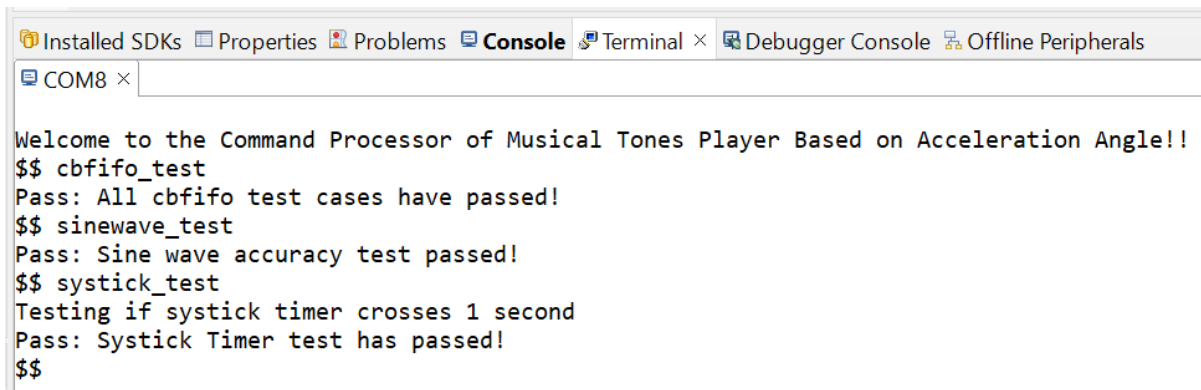
My project uses an additional hardware, AT-1224-TWT Buzzer as a means to hear the musical notes out of the DAC's output. This is connected to J10-11 pin of the FRDM-KL25Z Board.

Testing Strategy:

UART communication:



Automated tests were used to check the individual modules: CBFIFO, DAC output, SysTick Timer.

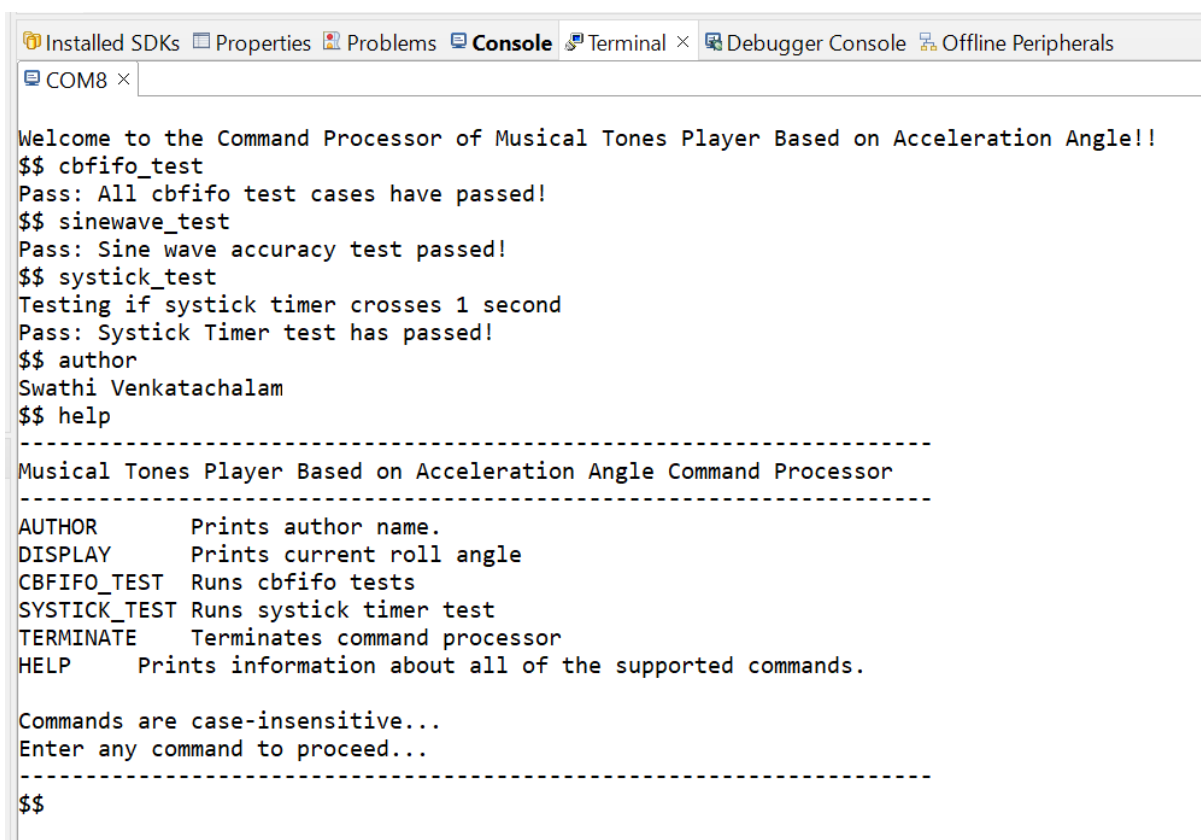


```

Welcome to the Command Processor of Musical Tones Player Based on Acceleration Angle!!
$$ cbfifo_test
Pass: All cbfifo test cases have passed!
$$ sinewave_test
Pass: Sine wave accuracy test passed!
$$ systick_test
Testing if systick timer crosses 1 second
Pass: Systick Timer test has passed!
$$

```

The automated tests are executed in the command processor. The other commands such as author and help are displayed.



```

Welcome to the Command Processor of Musical Tones Player Based on Acceleration Angle!!
$$ cbfifo_test
Pass: All cbfifo test cases have passed!
$$ sinewave_test
Pass: Sine wave accuracy test passed!
$$ systick_test
Testing if systick timer crosses 1 second
Pass: Systick Timer test has passed!
$$ author
Swathi Venkatachalam
$$ help
-----
Musical Tones Player Based on Acceleration Angle Command Processor
-----
AUTHOR      Prints author name.
DISPLAY     Prints current roll angle
CBFIFO_TEST Runs cbfifo tests
SYSTICK_TEST Runs systick timer test
TERMINATE   Terminates command processor
HELP        Prints information about all of the supported commands.

Commands are case-insensitive...
Enter any command to proceed...
-----
$$

```

Invalid/ Error Case:

```

Welcome to the Command Processor of Musical Tones Player Based on Acceleration Angle!!
$$ printf
Unknown command: printf
$$

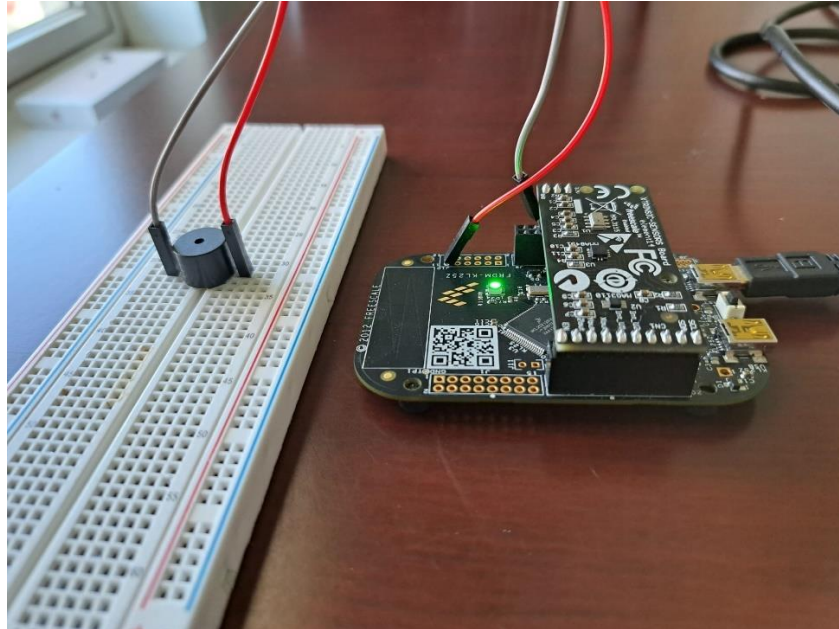
```

Happy Cases/ Corner Cases:

Accelerometer and LED manual test:

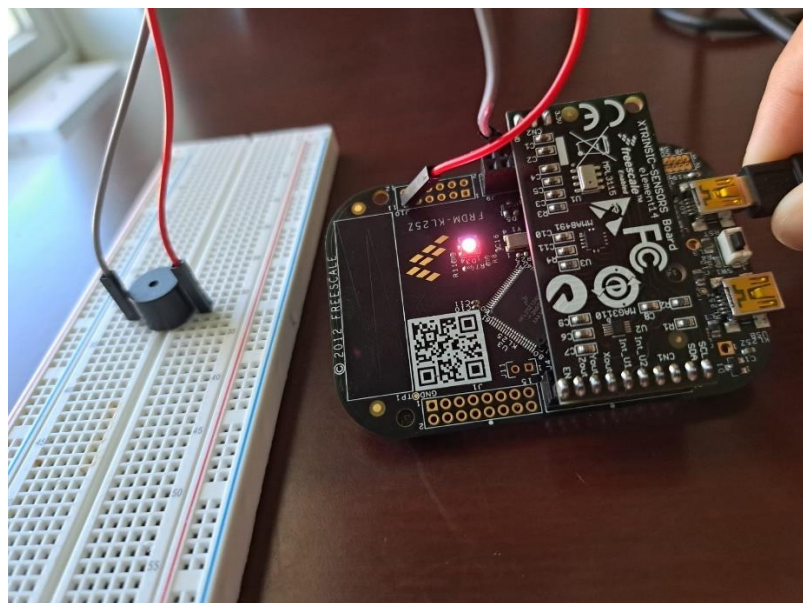
- 1) When the board is flat, the Green LED glows from 0° - 5° .

```
Welcome to the Command Processor of Musical Tones Player Based on Acceleration Angle!!  
$$ display  
The roll angle in degrees is: 1  
$$
```

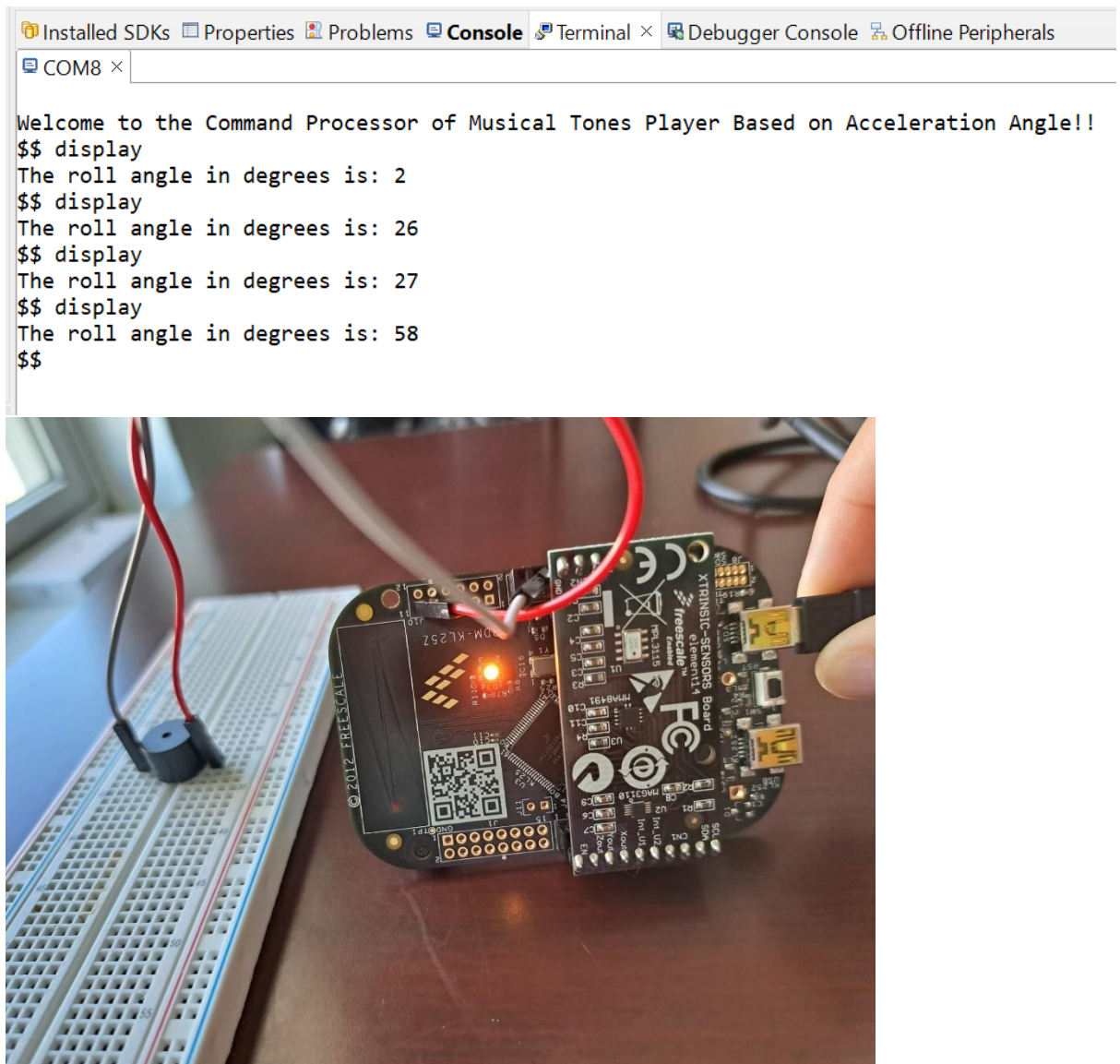


2) When the board is tilted from 6° - 45° , the white LED glows.

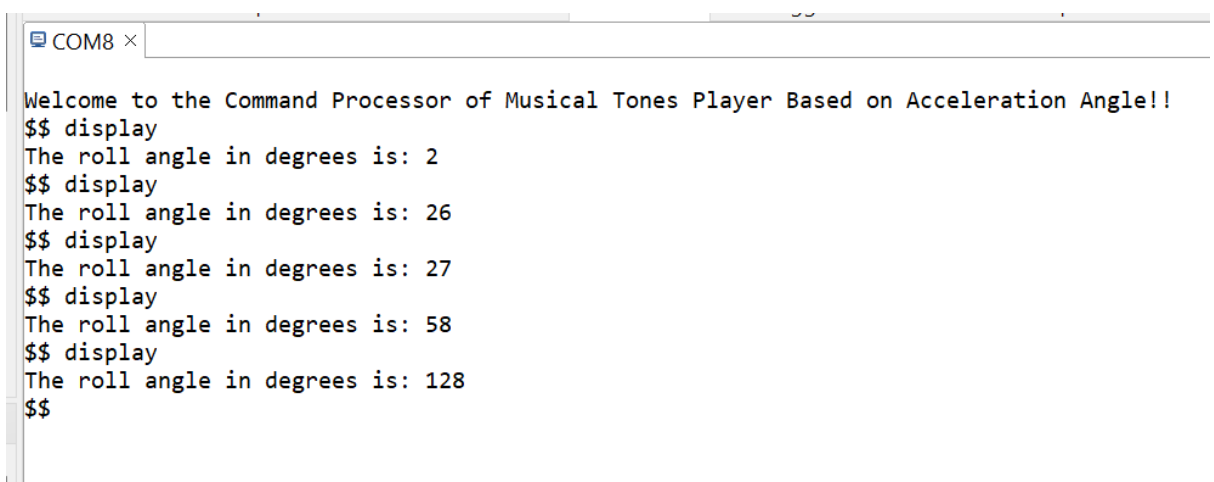
```
COM8 x  
Welcome to the Command Processor of Musical Tones Player Based on Acceleration Angle!!  
$$ display  
The roll angle in degrees is: 2  
$$ display  
The roll angle in degrees is: 26  
$$ display  
The roll angle in degrees is: 27  
$$
```

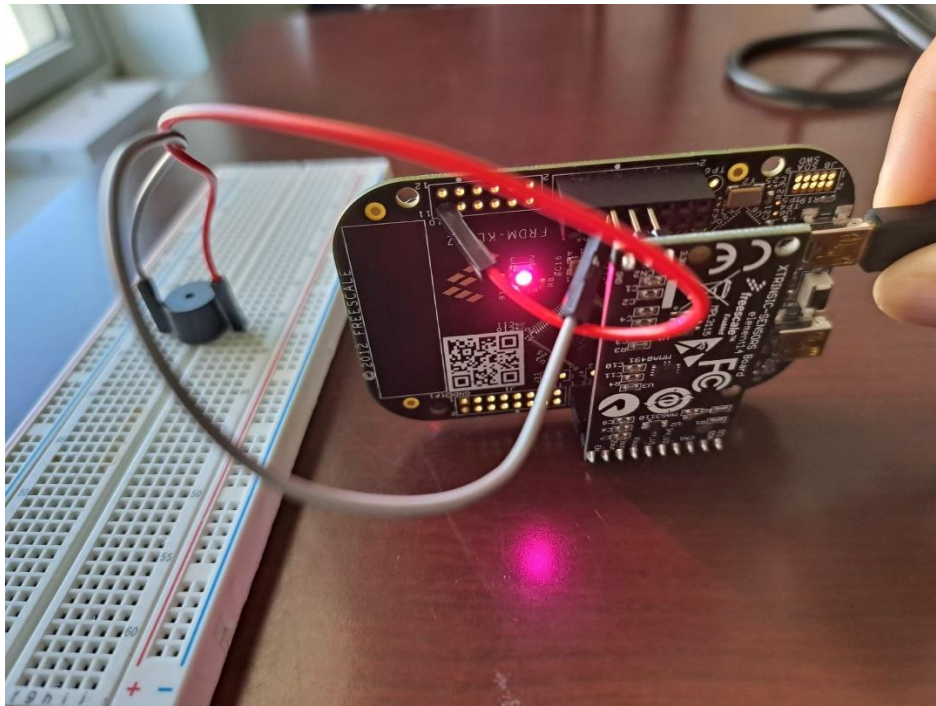


3) When the board is tilted from 46° - 90° , the yellow LED glows.



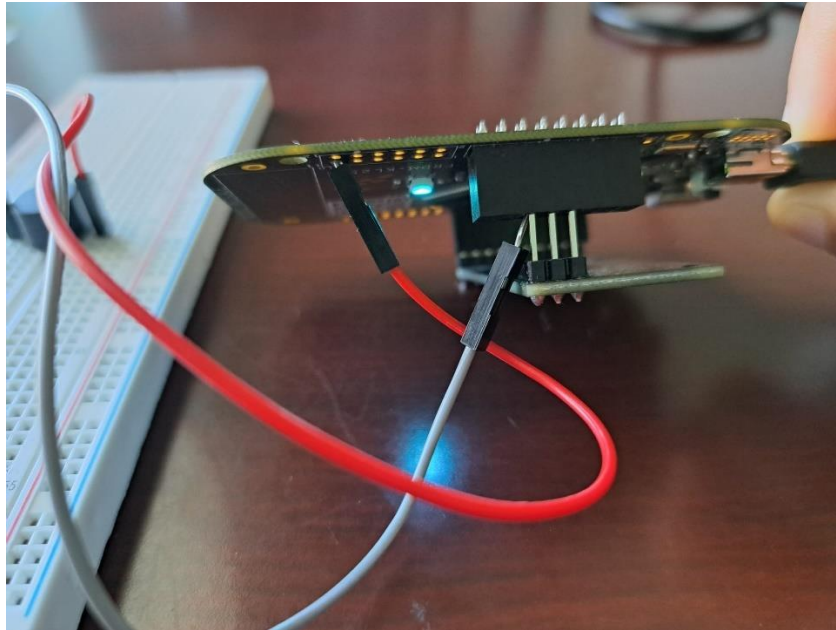
4) When the board is tilted from 91° - 135° , the magenta LED glows.



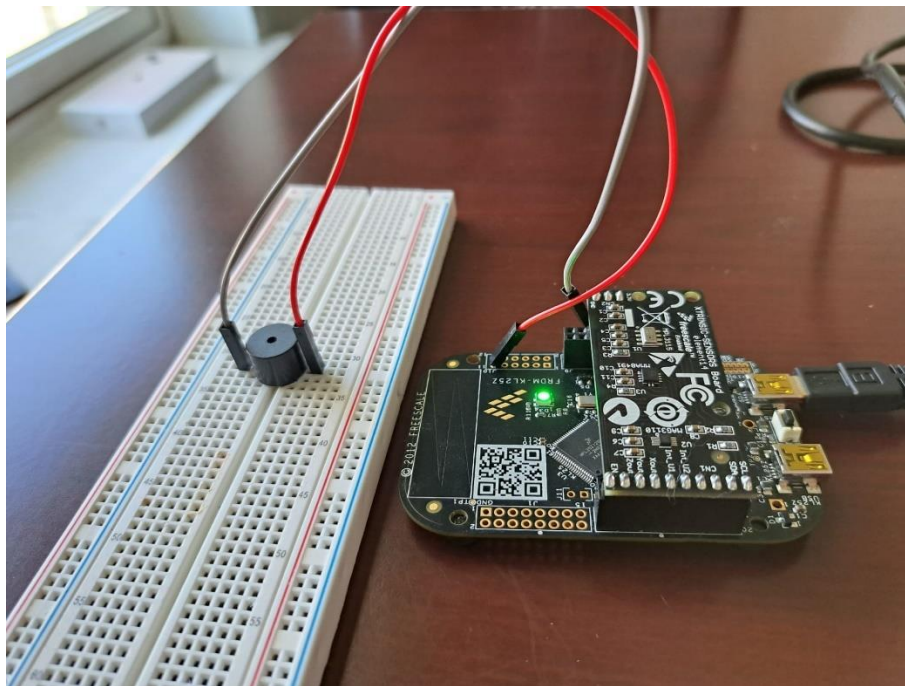


5) When the board is tilted from 136° - 180° , the cyan LED glows.

```
COM8 x
Welcome to the Command Processor of Musical Tones Player Based on Acceleration Angle!!
$$ display
The roll angle in degrees is: 2
$$ display
The roll angle in degrees is: 26
$$ display
The roll angle in degrees is: 27
$$ display
The roll angle in degrees is: 58
$$ display
The roll angle in degrees is: 128
$$ display
The roll angle in degrees is: 146
$$
```



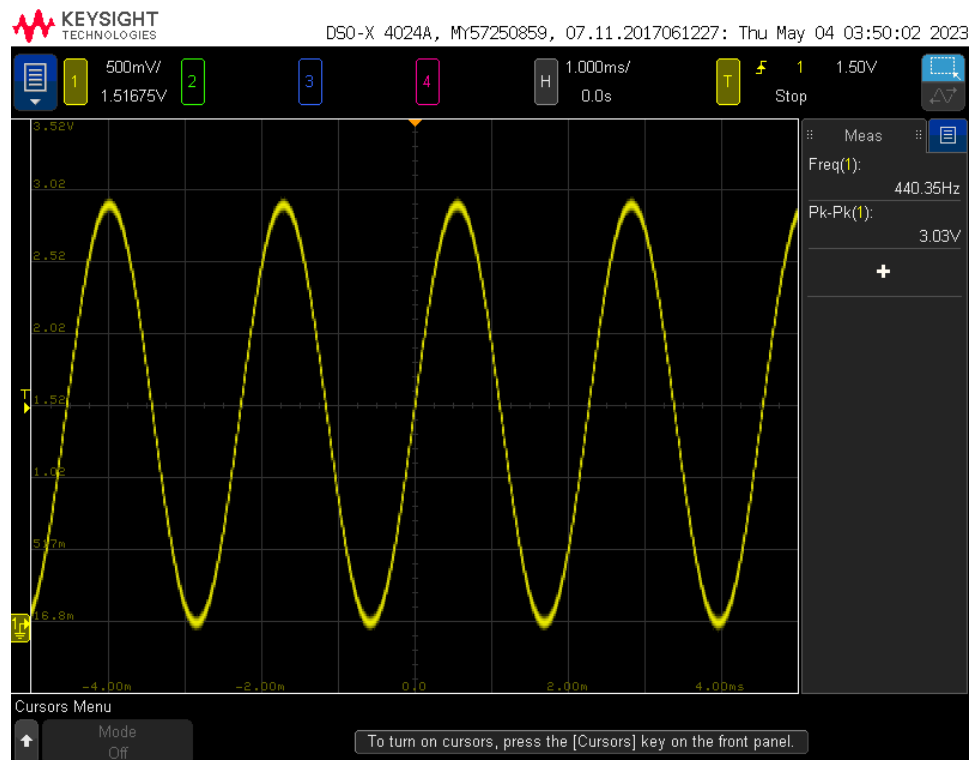
6) When the board is placed flat, the musical player stops and green LED glows.



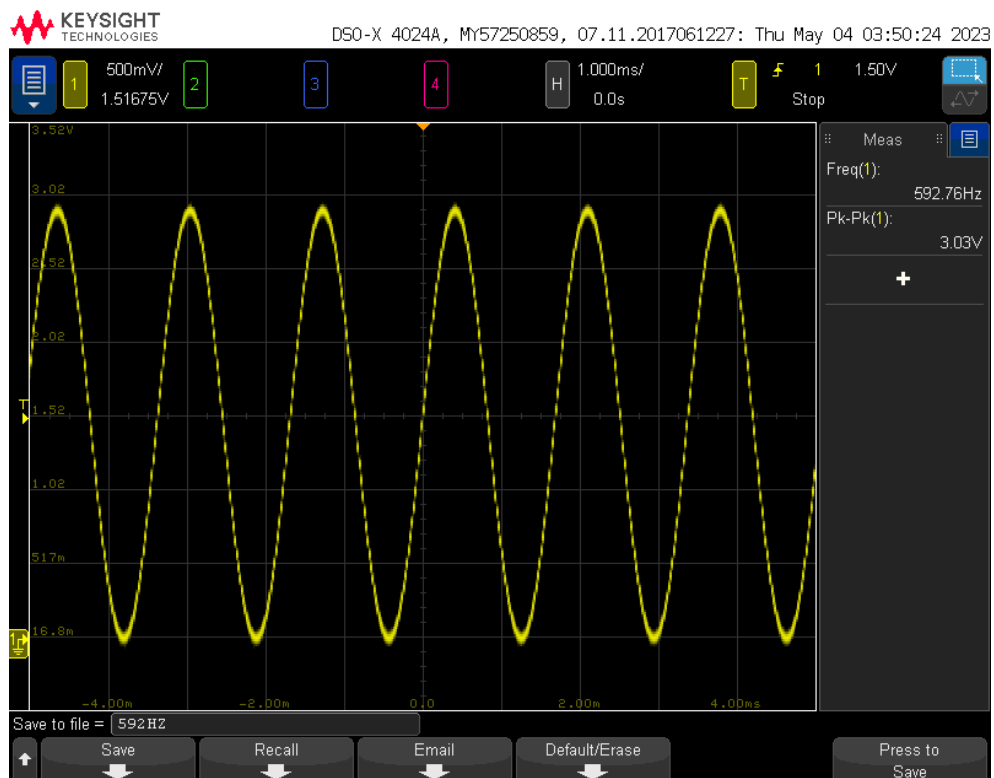
Oscilloscope Output:

I manually checked all the DAC output on the oscilloscope. I've included screenshots for 2 sine wave musical tones.

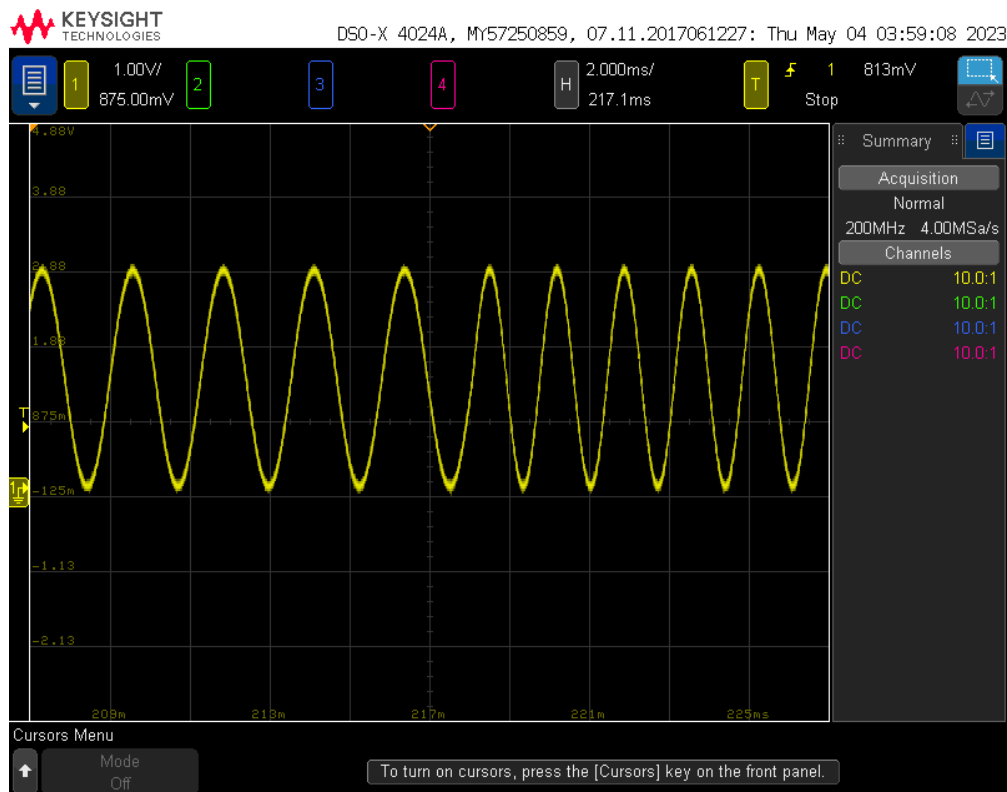
440 Hz Sine Wave Musical Note A4:



587 Hz Sine Wave Musical Note D5:



Smooth Transition from 440Hz to 587Hz:

ADC Output:

The ADC is feeded with the output from DAC. And the generated sine wave on DAC is analysed in ADC and displayed in terminal.

```
Generated 981 samples at 440 Hz; computed period=109 samples
min=33 max=65200 avg=34148 period=216 samples frequency=444 Hz

Generated 972 samples at 587 Hz; computed period=81 samples
min=45 max=65248 avg=33739 period=162 samples frequency=592 Hz
```

I've just displayed 2 sine wave tones here, but I've printed information for all of the tones generated.