Fall 2022 ECEN-5813:
Principles of Embedded Software
Final Project Report

Audio Playback System

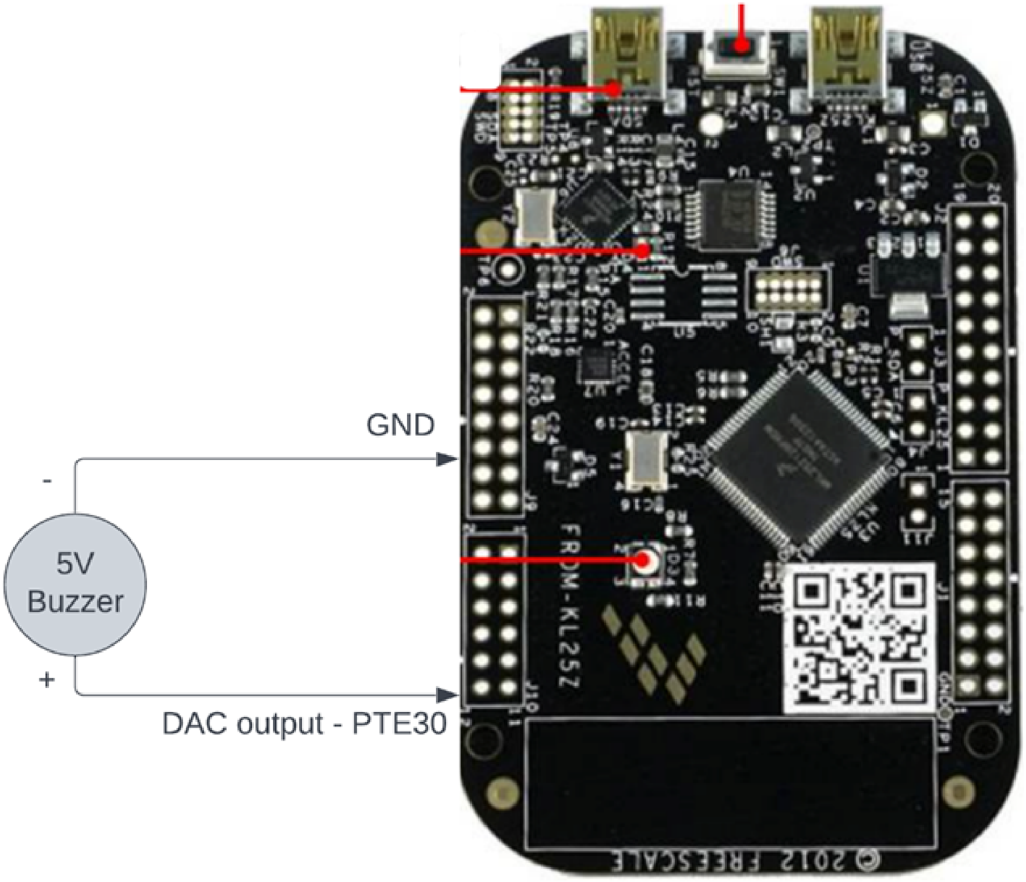
By

Vignesh Vadivel
Advisor: Lalit Pandit

Project Description:

This project builds a musical note generator **“CLASSIC HAPPY BIRTHDAY MUSIC”** where the frequency of the note being played would be changed every 500 milliseconds in the KL25Z development board in order to generate tone. I am able to generate a **Happy Birthday tone** with these set of frequencies - 262 Hz, 277 Hz, 294 Hz, 311 Hz, 330 Hz, 349 Hz, 370 Hz, 392 Hz, 415 Hz, 440 Hz, 466 Hz, 494 Hz. The volume of the note being played is controlled using the onboard MMA8451Q 3-axis accelerometer. The peripherals in use are as follows: I2C0, DAC0, TPM0 and DMA0. I used **DMA** and **Systick interrupts** to play this particular tone. I have used two buffers (low and high volume) to store samples of each frequency.

Block Diagram:



A Stereo Speaker is used to listen to the musical notes being produced. The steps to use this product is given below:

1. Connect a speaker to the board in this manner: positive lead to the DAC output pin (PTE30), and negative pin to any of the GND pins.
2. Power on the board, then the user could hear the HAPPY BIRTHDAY tone.
3. If the freedom board is along the surface, the BLUE LED would be turned ON and the Tone would be played at lower volume (25% of max) Peak to peak voltage is 860mV.
4. If the freedom board is perpendicular to the surface, the GREEN LED would be turned ON and the Tone would be played at higher volume. Peak to Peak voltage is 3.4V.

Program files and associated functionality:

1. i2c.c & i2c.h - contains all code to initialize the I2C0 peripheral. Further, it also contains code to allow I2C read and write by accepting device address, register address and input data as the parameters.
2. mma8451q.c & mma8451q.h - contains all code to initialize MMA8451Q 3-axis accelerometer in its active mode and use 14-bit mode at an output data rate (ODR). Further, it contains code that reads the acceleration data from the sensor and converts it into a usable roll-angle in degrees with a range of [0-180].
3. DAC.c & DAC.h - contains all code to initialize DAC0 peripheral to generate the required musical notes.
4. DMA.c & DMA.h - It contains code to initialize the DMA peripheral, Sample values for different sine frequencies, and start DMA transfer from DMA buffer to DAC output. It's an interrupt driven DMA, and based on accelerometer value volume would be controlled.
5. Systick.c & systick.h - It contains all necessary code for Systick handler, which triggers every 500 milliseconds and updates the new frequency index to be played.
6. Fp_trig.h & fp_trig.o - contains functions which would give sine value for a given degree.

Testing:

The test plan is described below. These tests were performed before creating the final demo, and submitting the final project:

1. DAC waveform output: The sine outputs from the DAC were tested with an oscilloscope to verify the various different frequencies generated (12 notes), and the peak-to-peak voltage of the output, which corresponds to the volume of the waveform. Sample screenshots for the different frequencies are attached to this submission, as a separate directory in the Github repo.
2. Accelerometer output: the raw acceleration values from the accelerometer were processed to determine the roll angle, which was verified against a known surface with an oil-level and a protractor. The two input angles used in this project correspond to placing the board flat on the surface (25% of max volume), and placing it perpendicular to the surface (max volume).

References:

1. Alexander G Dean github supporting code and materials for textbook, https://github.com/alexander-g-dean
2. NXP MMA8451Q reference manual and calibration document: https://www.nxp.com/docs/en/data-sheet/MMA8451Q.pdf