CECS 447 – Project 1: Digital Piano

Total: 30 points (+2 extra credit)

Overview

In this project you will build **two embedded systems** (two implementations of the same digital piano features) using the **TM4C123 LaunchPad** and external push buttons/speaker hardware. Both systems support the same user-facing behavior, but generate audio differently:

- **System P1 (Digital Output):** outputs a **square wave** to drive the speaker.
- **System P2 (Analog Output):** outputs a **sine wave** using a **6-bit R/2R DAC** and a **64-sample sinusoid wavetable**.

Each system has two modes:

- **Piano Mode:** press-and-hold one of 4 piano keys to play a note continuously; release stops the note. Only one key is assumed pressed at a time.
- **Auto-Play Mode:** plays one of three songs; the selected song repeats until changed.

Learning objectives

By the end of this project, you should be able to:

- Configure and use GPIO pins for input (push buttons) and output (audio).
- Use a hardware timer interrupt to generate precise periodic signals.
- Use GPIO interrupts for responsive user input handling.
- Design and build a simple 6-bit R-2R DAC and drive it from the MCU.
- Play notes and simple songs by controlling tone frequency and note duration.
- Learn how to generate sound with both digital signals and analog signals.

Hardware/Components

You will use: TM4C123 LaunchPad, **4 push buttons** (piano keys; optional 10kΩ resistors), speaker + audio amplifier, and resistors to build a **6-bit R/2R DAC** (System P2).

Reference example projects: Music, DAC_3bit.

System requirements

You will build two systems that implement the same user-visible behavior but differ in how audio is generated:

System	Audio output
System 1	Digital square wave directly driven from the MCU to the speaker.
System 2	Analog sine wave produced by a 6-bit R2R DAC driven by the MCU and going to the amplifier/speaker.

User controls

Assume that at any time no more than one piano key is pressed.

Control	Action	Behavior
4 piano keys (push buttons)	Press and hold	Play the assigned note continuously while held; stop immediately on release.
Switch 1 (left switch)	Press	Toggle between Piano mode and Auto-play mode.
Switch 2 (right switch)	Press	In Piano mode: change octave. In Auto-play mode: change song.

Operating Modes

Mode 1 – Piano mode (default on reset)

In Piano mode, each push button acts as a piano key. When a key is pressed and held, output the corresponding tone. The tone must continue without gaps while the button remains pressed and must stop when the button is released.

Octave selection: Switch 2 cycles through at least three octaves in round-robin order: Lower C, Middle C, Upper C. The system starts in Lower C after reset.

Note Frequencies

Note	Lower C (Hz)	Middle C (Hz)	Upper C (Hz)
C	262	523	1046
D	294	587	1174
E	330	659	1318
F	349	698	1396

The octave selected in Piano mode will be used as the starting octave when entering Auto-play mode.

Mode 2 – Auto-play mode

In Auto-play mode, the system automatically plays one of the following songs:

- Mary Had a Little Lamb
- Twinkle Twinkle Little Star
- Happy Birthday

On entry to Auto-play mode, start playing the currently selected song at the octave previously chosen in Piano mode. While a song is playing, Switch 2 cycles to the next song (round-robin). When the song selection changes, the newly selected song must restart from its beginning immediately. Song switching may occur at any time during playback.

Implementation constraints

- System clock must be greater than 16 MHz and less than 80 MHz.
- You must use interrupts for: (1) a hardware timer and (2) the push buttons/switches.
- You may choose any hardware timer(s) for timing control.

Waveform generation details

System 1 (square wave): Use a hardware timer interrupt to toggle a GPIO output at the correct rate for the selected note frequency.

System 2 (sine wave via DAC): Build a 6-bit R-2R ladder DAC and output a sampled sinusoid. Use a 64-sample sine lookup table for one cycle. The timer interrupt should advance the sample index at a fixed sample rate so that the output frequency matches the desired note.

Extra credit (2 points)

- Add a short video-game song as a fourth option in Auto-play mode.

Appendix: Song note sequences

The following note and duration sequences may be used in your project. Your implementation may represent notes and durations in any convenient data structure. Duration table: 4 for a quarter note.

Song	Note Table	Duration TAble
Mary Had a Little Lamb	|E D C D| E E E-| |D D D-| E G G-| |E D C D | E E E -| |D D E D C - |	|4, 4, 4, 4, 4, 4, 8| |4, 4, 8, 4, 4, 8| | 4, 4, 4, 4, 4, 4, 8| | 4, 4, 4, 4, 8|
Twinkle Twinkle Little Star	|C C G G|A A G -|F F E E|D D C -| |G G F F|E E D -|G G F F|E E D -| |C C G G|A A G -|F F E E|D D C -|	|4,4,4,4|4,4,8|4,4,4,4|4,4,8| |4,4,4,4|4,4,8|4,4,4,4|4,4,8| |4,4,4,4|4,4,8|4,4,4,4|4,4,8|
Happy Birthday	|C C|D C F|E C C|D C G|F C C| |C' A F|E D B B|A F G|F|	|3,1|4,4,4|8,3,1|4,4,4|8,3,1| |4,4,4|4,4,3,1|4,4,4|12|