# EMBEDDED SYSTEMS

## LAB PROJECT

COE17B010

## Group Details

- Group number: 8
- Group members:
  1. COE17B010
  2. COE17B036
  3. COE17B047
  4. CED17I001
  5. CED17I046

## Project Outline

Create a DAC using R-2R Ladder for Piano using Buttons and Resistance for 8 bit.

## Group Member Contribution

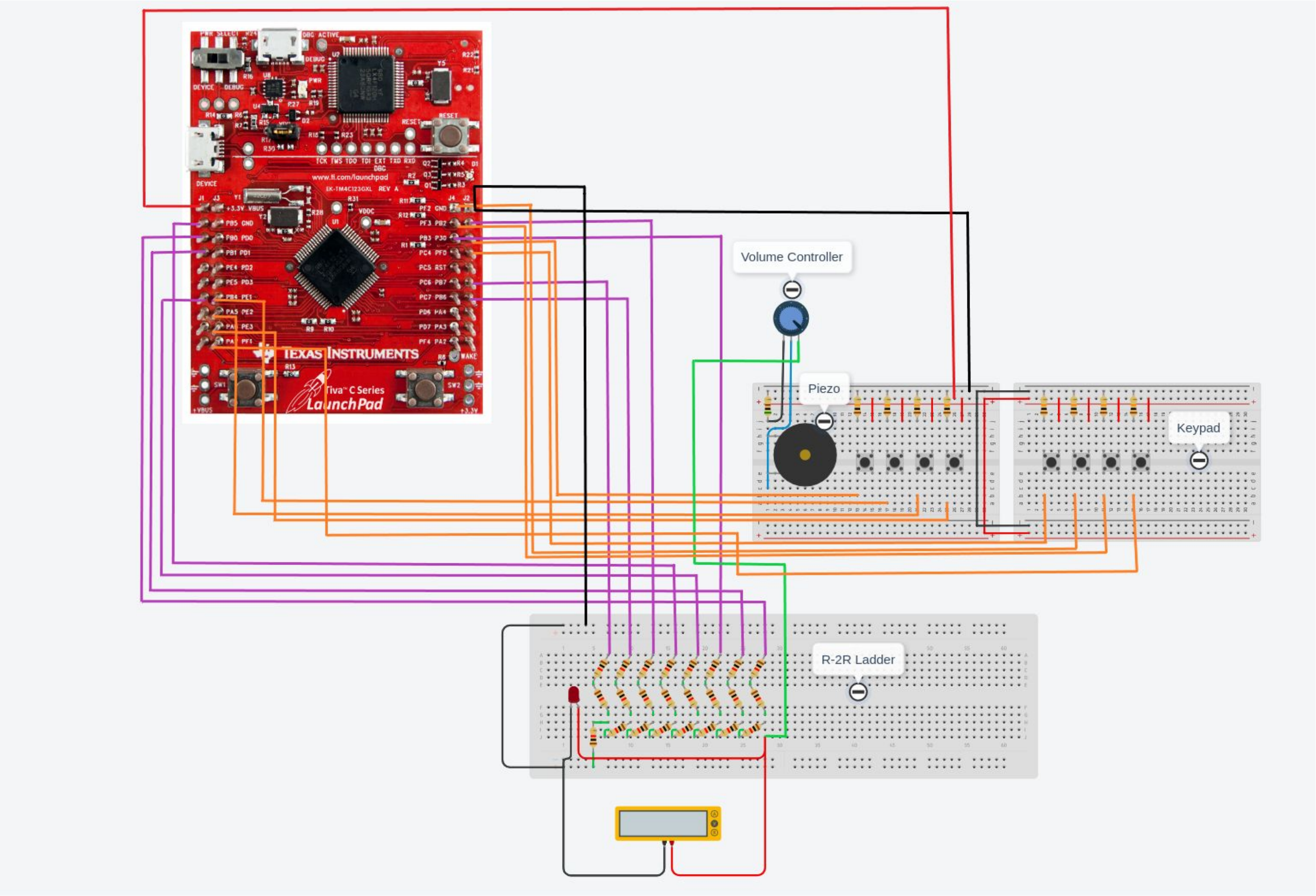
- Circuit Design
  a. Integration of Modules - CED17I001
  b. Button Configuration Setting - **COE17B010**, COE17B047
  c. R2R Ladder - COE17B036, CED17I046
- Tivac Code Logic
  a. *dac.c and dac.h* - **COE17B010**, CED17I001
  b. *sound.c and sound.h* - COE17B047, CED17I046
  c. *piano.c and piano.h* - COE17B036
  d. *main.c* - **COE17B010**, COE17B036, COE17B047, CED17I001, CED17I046

- **COE17B010**
    1. Worked on button to pin connections (VCC, GND, OUTPUT from Button)
    2. Worked on deciding which pins to use for DIGITAL INPUT from 8 buttons and assigned those pins as DIGITAL INPUT
    3. Worked on TIVAC Code for Initialising DAC Pins and required Clocks
    4. Worked on TIVAC Code for Writing 8 bit binary value to 8 ports of Port B
- COE17B036
    1. Constructed R-2R Ladder Design for 8-bits in Tivac Board using RESISTORS and WIRE CONNECTORS
    2. Connection of Multimeter and LED in the R-2R Ladder.
    3. Initialize the GPIO Ports (E and F) as digital pins and set them to input mode. Unlock PORT E and F and disable the analog function. We also clear the PCTL register for Ports E and F.
    4. Worked on Tivac Code for checking which buttons are clicked by reading corresponding PORTS(Port E and F) and setting a frequency index for each button. Each button is corresponded with a different keynote.
- COE17B047
    1. Worked on sending the pushbutton signals to Tiva C pin connections (VCC, GND, taking output signal from push buttons when pressed and feeding the input to Tiva C - 4 port E and 4 port F GPIO pins).
    2. Worked on deciding which pins to use for DIGITAL INPUT from 8 pushbuttons along with their configurations and the Tiva C board pin connections for resistors in R-2R ladder (port B pins).
    3. Worked on Tiva C code for sending binary input values (corresponding frequencies of piano notes) from Tiva C to corresponding R-2R Ladder resistors in the circuit.
    4. Calculated the frequencies of 8 different piano notes and worked on changing the output volume from the piezo speaker by connecting a potentiometer and verifying the correctness of output voltage using a multimeter.
- CED17I001
    1. Integration of different modules(R-2R Ladder, PushButton and Arduino).
    2. Worked on WIRE CONNECTORS between Buttons to TIVAC, TIVAC to R-2R Ladder, R-2R Ladder to Piezo (Connections between all major modules)
    3. Worked on Initialising DAC Ports by enabling ports and initialising data directions
    4. Worked on Address Definitions for Port B and other required DAC Ports in header file
- CED17I046
    1. Constructed R-2R Ladder Design for 8-bits in Circuit using RESISTORS and WIRE CONNECTORS with LED Indicator and Multimeter for Validation of frequency
    2. Checking and Setting Frequency for 8 Basic notes in music for the generation of Sine Wave to get through DAC R-2R Ladder

c. Worked in Tiva C Code for reading GPIO Port B in range of  PB0-PB7 Pin from R-2R Ladder to corresponding Tiva C Button Connected GPIO Port ranging from PE0-PE3 and PF0-PF3 Port

d. Worked in Tiva C Code for giving GPIO  PE0-PE3 and PF0-PF3 port  from Tiva C to R-2R ladder DAC to Piezo to play the specific frequency

# Diagram



## Components Used

- Tiva TM4C123GH6PM Microcontroller
- Buttons
- Multimeter
- Potentiometer
- Piezo speaker
- 10kΩ resistors - 8
- 1kΩ resistors - 24
- 500Ω resistor - 1
- 1 Red LED

## Pin Configurations

| Pin Name | Pin Type | Connection |
|---|---|---|
| PB0 | DIGITAL OUTPUT | R-2R Ladder Bit 1 |
| PB1 | DIGITAL OUTPUT | R-2R Ladder Bit 2 |
| PB2 | DIGITAL OUTPUT | R-2R Ladder Bit 3 |
| PB3 | DIGITAL OUTPUT | R-2R Ladder Bit 4 |
| PB4 | DIGITAL OUTPUT | R-2R Ladder Bit 5 |
| PB5 | DIGITAL OUTPUT | R-2R Ladder Bit 6 |
| PB6 | DIGITAL OUTPUT | R-2R Ladder Bit 7 |
| PB7 | DIGITAL OUTPUT | R-2R Ladder Bit 8 |
| PE0 | DIGITAL INPUT | BUTTON 1 |
| PE1 | DIGITAL INPUT | BUTTON 2 |
| PE2 | DIGITAL INPUT | BUTTON 3 |
| PE3 | DIGITAL INPUT | BUTTON 4 |

| | | |
|---|---|---|
| PF0 | DIGITAL INPUT | BUTTON 5 |
| PF1 | DIGITAL INPUT | BUTTON 6 |
| PF2 | DIGITAL INPUT | BUTTON 7 |
| PF3 | DIGITAL INPUT | BUTTON 8 |

# Code Logic

1. dac.c and dac.h
    - a. InitDac() - Initialize the 8-Bit DAC by doing the following
        - i. unlock port B clock
        - ii. allow time for clock to start
        - iii. unlock GPIO Port B
        - iv. allow change to PB7-PB0
        - v. disable analog function on PB7-PB0
        - vi. clear PCTL register on PB7-PB0
        - vii. PB7-PB0 outputs
        - viii. disable alternate function on PB7-PB0
        - ix. enable digital pins on PB7-PB0
        - x. enable 8-mA drive select on PB7-PB0
    - b. OutDac(data) - Writes the data to Port B Pins 0 to 7
2. sound.c and sound.h
    - a. PlaySound(period) - reset the RELOAD value for different frequencies and hence play a specific frequency
        - i. Use Sine Wave to generate amplitudes by sampling frequency
    - b. StopSound() - Stop playing by clearing PB7-PB0
    - c. SysTick_Handler() - SysTick interrupt service routine
3. piano.c and piano.h
    - a. InitPiano() - Initialize the Piano Button Inputs by doing the following
        - i. unlock port E clock
        - ii. allow time for clock to start
        - iii. unlock GPIO Port E
        - iv. allow change to PE3-PE0
        - v. disable analog function on PE3-PE0
        - vi. clear PCTL register on PE3-PE0
        - vii. PE3-PE0 inputs
        - viii. disable alternate function on PE3-PE0
        - ix. disable pull-up resistors on PE3-PE0
        - x. enable digital pins on PE3-PE0

      xi. unlock GPIO Port F
      xii. allow change to PF3-PF0
      xiii. disable analog function on PF3-PF0
      xiv. clear PCTL register on PF3-PF0
      xv. PF3-PF0 inputs
      xvi. disable alternate function on PF3-PF0
      xvii. disable pull-up resistors on PF3-PF0
      xviii. enable digital pins on PF3-PF0
   b. PianoIn() - Check which button input is pressed
4. main.c
   a. Initialise the TExaS module
   b. Initialise the sound playing pins (initialize SysTick and DAC)
   c. Initialise the piano button inputs
   d. Enable interrupts
   e. Check which buttons are pressed and appropriately send binary data to DAC to play sound
5. TExaS.h - Initialises Input and Output Ports for DAC
6. tm4c123gh6pm.h - Library for accessing TIVAC pins