**Lab Experiment 1: Playing musical tones through a buzzer**

**Introduction:**

This lab introduces students to the EduBase-V2 Trainer board and the Tiva EK-TM4C123GXL Launchpad. I

**Procedure:**

The EduBase-V2 has a speaker at the PortC4. This is the one that we will be using for playing musical tones.

PortC4 is the name of the Pin that is on the EduBase-V2 board. The Pin # is 7 of J19. J19 is the on-board jumper.

Table

Description automatically generated

The TM4C123GH6PM Microcontroller has six GPIO ports i.e. PORTA, PORTB, PORTC, and PORTD. Each port has a different number of pins. For this lab we will be using the GPIO Port C. The address range for this port can be seen below.

Table

Description automatically generated

To use a GPIO port we need to configure it first. Our first step in GPIO configuration is to enable the clock for a particular peripheral we want to use. Before we move on, we need to make sure we are including the microcontroller’s essential header file. This is the one that comes with the installation of Keil IDE.

Text

Description automatically generated

Now, a particular port can be enabled by setting an appropriate bit field for the required GPIO port in the RCGCGPIO register. Here, the name of the GPIO clock register is a crucial component. Physical addresses associated with GPIO pins are used to define the register names. These names **must not be changed**. All peripheral register names are given using pointers and structures. You can check these out by looking at the contents of the “TM4C123GH6PM.h” header file.

Now instead of using the physical address to initialize everything, we will be using the header file. RCGCGPIO is mapped to the address 0x400FE608. These memory address mappings can be seen in the TM4C123GH6PM datasheet.

To enable port A to port F on the peripherals, we use the bit 0 to bit 5 of the RCGC\_GPIO\_R register, respectively.

Since, for our lab we are using Port C, therefore, to enable to the clock for Port C we set the second bit of the RCGCGPIO as shown below.

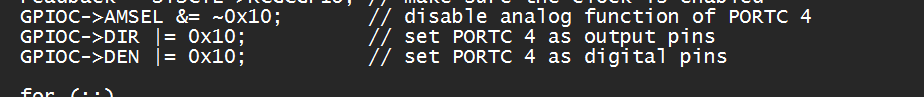
To make sure our clock is enabled, we are accessing the contents of the structure SYSCTCL by using the -> symbol and assigning it to the readback.

Text

Description automatically generated

Now that we have configured the clock, it is time to configure the GPIO PORT C itself.

We set the PORT C4 as the output port by setting the DIRECTION REGISTER (“Dir”) HIGH. GPIODIR register decides, which pins of the PORT will be configured either as a digital input or a digital output.



Now that our GPIOC port is fully configured it is time to use it to play musical notes.

Our first step is to write some kind of a delay function that we can use for setting the pitch of the notes.