

1. [20 points] In order to convert an analog signal to a digital signal, the 3 steps to be followed are sampling, quantization and encoding. In this lab, you will use the board to read inputs (assume these are the sampled inputs), then quantize and encode the inputs, and display the encoded digital outputs using the on-board LEDs.

- Use `delay_1ms` subroutine given in the previous lab to make a subroutines to generate 4 seconds delay and 8 seconds delay.
- Write a function to take 4 user inputs using pins with the help of 8 seconds delay function, and store them in 4 memory locations.

Procedure to take inputs

- First set the pins in the required port (P1.0 to P1.3 since they are connected to on-board DIP switches) as logic 1 (i.e store `0Fh` in the port) to configure the pins as input pins.
- Since the inputs are 8-bit values but only 4 switches are available, only 4-bits of an input can be read at a time. For example, if the input to be provided is `1Ah`, then the input pins should be provided with `0001` first and then with `1010` (after some delay).
- Use delay function to provide gaps between the applications of different inputs. The on-board LEDs can be used to indicate when the delay is complete. The scheme is illustrated below:
 - * LED0 (indicating binary value of 1) should be on for 8 seconds, during which one half of the first input must be provided.
 - * Next, LED1 (indicating binary value of 2) should be on for 8 seconds, during which the second half of the first input must be provided.
 - * Then LED1 and LED0 (indicating binary value of 3) should be on for 8 seconds, during which the first half of second input must be provided, and so on till all 4 inputs are read.
- Before uploading on the board, give inputs using Peripherals → I/O-Ports → Port 1; and verify the outputs by adding P1.4 to P1.7 to Logic Analyzer separately as bits in Keil.
- Write a function that performs quantization and encoding on the inputs based on the given scheme.
- Write a function to display the 4 encoded outputs on the 4 on-board LEDs one after another with 4 seconds delay. Put this in an infinite loop so that you can observe the outputs repeatedly.

Quantization & Encoding scheme:

```
if (sample >= 0 and sample < 64):
    digital_output = 0001
else if (sample >= 64 and sample < 128):
    digital_output = 0010
else if (sample >= 128 and sample < 192):
    digital_output = 0100
else:
    digital_output = 1000
```

Starting Code-

```
// -- DO NOT CHANGE ANYTHING UNTIL THE **** LINE--//
ORG 0H
LJMP MAIN
ORG 100H
MAIN:
CALL TAKE_INP
CALL QUANT_ENC
CALL LED_DISP
HERE: SJMP HERE
ORG 130H
// *****
DELAY:
// ADD YOUR CODE HERE
TAKE_INP:
// ADD YOUR CODE HERE
QUANT_ENC:
// ADD YOUR CODE HERE
LED_DISP:
// ADD YOUR CODE HERE
RET
END
```

TA Checkpoints

1. Check if the student knows how to give inputs from Peripherals → I/O-Ports → Port 1 and how to verify the outputs in Logic Analyzer in Keil.
2. Check if the delays generated by the student are correct using Logic Analyzer (4 seconds and 8 seconds).
3. Check if the student knows how to operate Flip/ DFU Programmer, and shows all the tests mentioned.
4. Check if student knows how to generate the hex file, upload it onto the board, provide inputs using pins and display outputs on the LEDs.
5. Check the working for the following 2 test cases:
 - Input samples : 11H, F5H, 92H, 48H.
Outputs on LEDs : 0001, 1000, 0100, 0010.
 - Input samples : 00H, 40H, 80H, FFH.
Outputs on LEDs : 0001, 0010, 0100, 1000.