NPTEL
Summer Workshop on Microcontrollers
WEL, IIT Bombay

Problem set: 3 Date: June 7, 2023

Pt-51 Board Introduction

[1] You will be demonstrating that your PT-51 kit works by running the test code for the kit. The test code also checks the peripherals on the kit. Using this, you can also verify that all required software is correctly installed. Before coming to the lab please install the software mentioned below and go through the slide decks to get familiarized with the board.

- Get familiar with PT-51 kit by going through the slides here: **User Manual**
- Install FLIP (on Windows) or DFU Programmer (on Linux/Mac). This software is used to load programs into the 8051 microcontroller on the PT-51 kit. The steps are here: **FLIP/DFU Installations**

In the in-lab session, verify the following in front of your TAs.

- The procedure to load a hex file onto the PT-51 kit using Flip is shown in the video given here: **Running a Hex file on Pt-51**

  Follow the procedure and load the `led.hex` file to see that the LEDs on your kit toggle.
- Download the file `pt51_test.hex` and load it on your PT-51 kit, as described here: **Testing the peripherals of Pt-51**
- Follow the steps in the slide deck titled "PT-51 Test Program" to do the self-test of the kit. Check if all tests run successfully. If there are failed tests, inform your respective TA.
- Please use the installation given here **JRE_Installer** if you are not able to find JRE or the installation with JRE fails.
- Please put your board in boot mode before installing drivers and flashing code. Detach → Press boot → Press Reset → Release Reset → Release Boot → Attach

[2] In this handout, you will use the board to read inputs, quantize and encode the inputs, and display the encoded digital outputs using the on-board LEDs.

Use `delay_1ms` subroutine shown below that generates 1 ms delay to make a subroutine to generate the required delay.

```
delay_1ms:
    push 00h
    mov r0, #4
    h2: acall delay_250us
    djnz r0, h2
    pop 00h
    ret

delay_250us:
    push 00h
    mov r0, #244
    h1: djnz r0, h1
    pop 00h
    ret
```

When using the logic analyzer in Keil, make sure you change the default clock frequency to 24MHz (as it is on the PT-51 board). To verify the delay amount, refer the **8051 Instruction Set** to find number of cycles each instruction runs for and how correct amount delay is achieved. This is called a software delay.

- Using above subroutines write functions to generate 5 seconds delay and 10 seconds delay.

- Write a function to take 4 user inputs using pins with the help of 10 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 10 seconds, during which one half of the first input must be provided.
    * Subsequently, LED1 (indicating binary value of 2) should be on for 10 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 10 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 5 seconds delay. Put this in an infinite loop so that you can observe the outputs repeatedly.

Quantization scheme:

```
if (sample >= 0 and sample < 10):
    quantized_sample = 5
else if (sample >= 10 and sample < 20):
    quantized_sample = 15
else if (sample >= 20 and sample < 30):
    quantized_sample = 25
else:
    quantized_sample = 35
```

Encoding scheme:

```
if (quantized_sample == 5):
    output 0001
else if (quantized_sample == 15):
    output 0010
else if (quantized_sample == 25):
    output 0100
else:
    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
```

Checkpoints

1. Check the working for the following 2 test cases:
    - Input samples : `11H`, `2AH`, `01H`, `18H`.
      Outputs on LEDs : `0010`, `1000`, `0001`, `0100`.
    - Input samples : `3FH`, `1CH`, `0EH`, `06H`.
      Outputs on LEDs : `1000`, `0100`, `0010`, `0001`.