EE 337: Microprocessors Laboratory (Spring 2022)
Indian Institute of Technology Bombay

Lab 8: 20 points Date: March 15, 2022

From this lab, you will be starting to use embedded C instead of assembly language to write your programs. Please note the following.

- The Moodle announcement of this lab has a zip file attached containing skeleton code for this lab.
- If you face issues with connecting the USB-to-UART adapter to your laptop, please contact WEL staff member Mahesh Bhaganagare mab@ee.iitb.ac.in to get access to a WEL PC. Apple laptop users may face this issue. The lab PCs are busy during other lab slots. So, **do not wait till the last minute** to contact Mahesh.

1. [5 points] In this question, you shall once again interface the speaker using 8051, but instead of playing a song, you shall generate the Morse Code of 5 different symbols, which shall be given as input using the switches. To choose the symbol using input, use a priority encoder type logic as shown below. Note that the below code is Python-style pseudocode and cannot be used for the 8051. You have to rewrite this logic in C.

```
if P1_0:
    symbol1
elif P1_1:
    symbol2
elif P1_2:
    symbol3
elif P1_3:
    symbol4
else:
    symbol5
```

In the above logic, the following is true.

- `symbol1` represents the letter 'A'
- `symbol2` represents the letter 'B'
- `symbol3` represents the letter 'C'
- `symbol4` represents the letter 'D'
- `symbol5` represents the letter 'E'.

The Morse code for these symbols can be found at `https://en.wikipedia.org/wiki/Morse_code`.

A Dot should be represented by 750ms duration of a single frequency and a Dash by 1500ms of the same frequency. The exact frequency are irrelevant. Between consecutive sounds, there must be a 1000ms pause. Additionally, the character should be displayed on the LCD screen as well. Use the functions provided in 'LCD.h' to achieve the same.

The Morse code for the corresponding symbol should play only once per reset. At reset, the symbol must be played as per the value on the switches and to play a new symbol, switch positions must be changed.

You may use the skeleton code provided for this question. **Shoot a 30 second video of your kit emitting all five symbols. Make sure your face appears at the beginning of the video.** The video needs to be uploaded using the **Video upload link for lab 8** assignment in the course team on MS Teams (it has an option to attach files). The file size cannot exceed 200 MB.

2. [15 points] Refer to Prof. Dinesh Sharma's slides and notes on serial I/O to do this question. `https://ee337.github.io/dks.html#serial-io`

   In this question, you will understand and use a program for communicating between Pt-51 and a computer using UART. This program will take inputs from the computer's keyboard that can be used in programs running on the kit, to perform appropriate operations.

   i) To connect the kit to a computer, you will be using the USB-to-UART adapter `Prolific PL2303` (Fig. 1) that was shipped along with the kit. The driver software for this adapter and the instructions for installing it can be found at the following link: `http://www.miklor.com/COM/UV_Drivers.php`
   After installing the software, connect the PL2303 adapter to one of the USB ports of your PC.

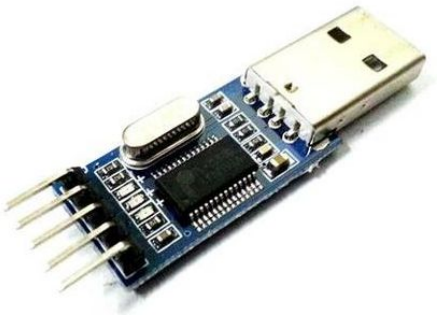

Figure 1: USB to UART adapter.

   ii) Connect the Pt-51 kit to the USB-to-UART adapter using F-F wires as described next. In the Pt-51 kit, port pin `P3.0` is the serial data input and `P3.1` is the serial data output. Connect `P3.0` of the kit to transmit data line (TxD) of the adapter. Connect `P3.1` of the kit to receive data line (RxD) of the adapter. Connect the `GND` pin of the kit to `GND` pin of adapter.

   iii) For recognizing keyboard inputs on the computer and transmitting to the kit, through the serial terminal, you need to use the `Realterm` software (or any equivalent software). This software will also be used to display the messages received from the Pt-51 kit on the PC. A screenshot of `Realterm` window is shown in Fig. 2.
   For Windows, download `Realterm` at: `https://realterm.i2cchip.com`
   For Linux, you can download `putty`.

   iv) Configure `Realterm` (or equivalent tool) to use the appropriate COM port and baud rate. This can be done by clicking on `Serial Port`/ `Port` tab and choosing appropriate COM port as the port to which the USB-to-UART adapter is connected. Set the baud rate to 4800 (or that used in your program). Then click on `Open`.

   With the kit connected to the computer using the USB-to-UART adapter, you will next write code to recognize key presses, perform an operation on the kit, and display messages sent by the kit on the computer.

   Use `main.c` as starting (template) code. This uses `serial.c` and `lcd.h` for initialisation of UART and LCD, respectively. These files are part of the zip file shared with you.

   The flow of the program is given below.

   1. Complete the function `uart_init()` in `serial.c` to configure the serial port for UART communication. Use timer `T1` and a baud rate of 4800 bps.

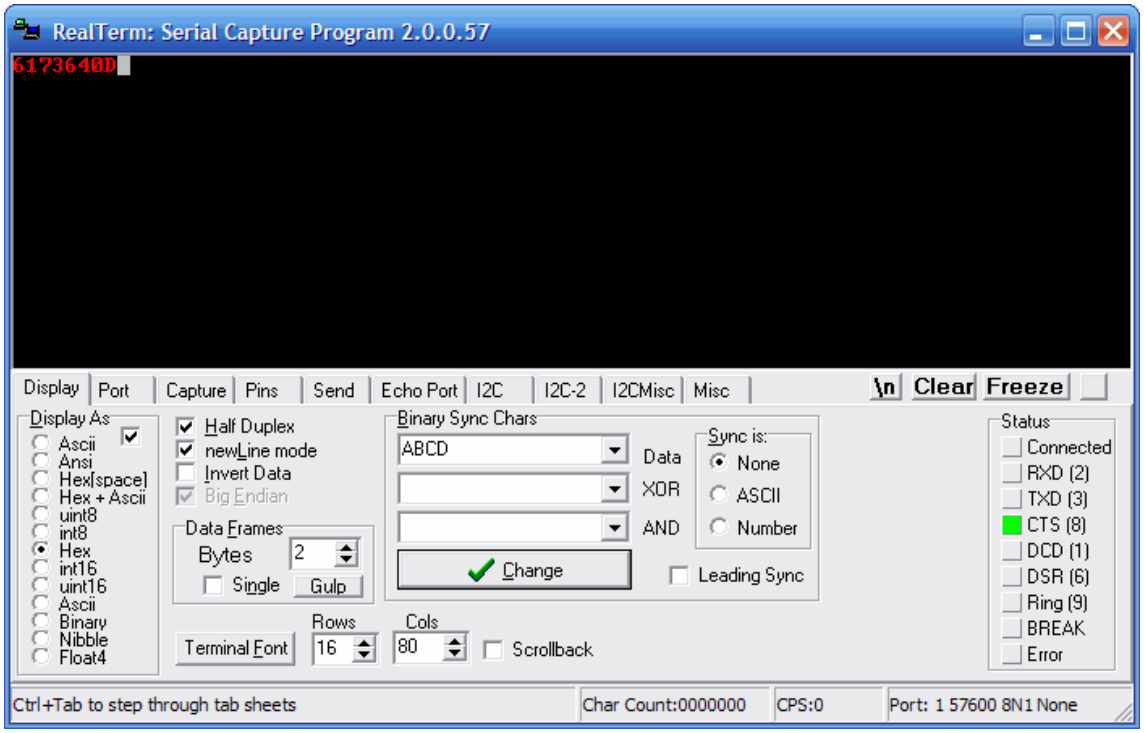

Figure 2: RealTerm: Serial/TCP Terminal

2. The given `main.c` has code for transmitting a set of strings and to capture a key press on the PC. Compile the project, dump the hex file on to Pt-51 and run the program. Launch the `Realterm` window on your PC and you can see that the following is displayed on the window.

```
**********************
******8051 Tests********
**********************
Press 1 to test the LCD
Press 2 to test the LED
```

3. Now, Pt-51 should be respond to the key presses on the PC's keyboard. With `Realterm` window in focus (active window), any key press on PC's keyboard will be captured by it and transferred to Pt-51.
   - When 1 is pressed, it will be detected by Pt-51 and you will see a message "EE 337" on the LCD screen and "LCD tested" display on the `Realterm` window. This happens because of the function `lcd_test` in `main.c`.
   - When 2 is pressed, it should be detected by pt-51 and should glow all four LEDs serially (In any order you like) with a gap of 500ms and display "LED tested" on the `Realterm` window after the pattern is shown. For this to happen, you will have to update the function `led_test` in `main.c`
   - When any other key is pressed, you will see the message "Incorrect key pressed" displayed on the `Realterm` window.

TA Checkpoints

1. For question 1, check that the student has used the correct logic to generate the sounds.

2. For question 2, verify that the student can demonstrate the UART setup.