EE 337: Microprocessors Laboratory (Spring 2023)

Indian Institute of Technology Bombay

Lab 6: 20 points Date: February 13, 2023

1. [5 points] In this experiment, you will learn to display content on the LCD connected to the Pt-51 kit. Download `lcd.h`, `lcd.c` files and `lcd-control-made-easy.pdf` from the following links.

   - `https://ee337.github.io/2023/downloads/lcd.h`
   - `https://ee337.github.io/2023/downloads/lcd.c`
   - `https://ee337.github.io/2023/downloads/lcd-control-made-easy.pdf`

   The `lcd-control-made-easy.pdf` has general information about LCD operation which is helpful in understanding the code in `lcd.h`. Also, `lcd.h` has comments for each line, try to understand the comments by going through the code line by line.

   - Compile `lcd.c` with header file `lcd.h` and load the hex file on to the kit. Make sure the output on the LCD screen is as shown below:

     ```
       Pt-51
     IIT Bombay
     ```

   - Study the functions used in the `lcd.h` code and their usage. Modify `lcd.c` to display "EE337-2023" on the first line and your **first name** on the second line (truncate to 16 characters if you have a longer name). **Pad the display lines with spaces such that these are centered on the LCD when displayed.** You should load and run this program on the Pt-51 kit.

2. [15 points] Write a program in Embedded C to read five numbers of 4-bit each, sort these numbers in ascending order and display the numbers in the sorted order. Then follow the below steps. For generating the given delays improvise and use `msdelay` function given in `lcd.h` file which generates 1 ms delay.

   - At the start, display **"START PROGRAM"** on the first row of LCD screen for 5 seconds and keep all LEDs OFF.
   - While taking the first input, display **"FIRST INPUT"** on the first row of LCD screen for 5 seconds.
   - Read the first input number from port 1 (P1.3-P1.0) DIP switches and display it on the LEDs connected to port 1 (P1.4 to P1.7) for 5 seconds. Also, store the first input number in an array.
   - While the first input is being displayed on LED, give the second input number using DIP switches and display **"NEXT INPUT"** on the LCD.
   - After the the first input is displayed for 5 seconds, pause for 1 second keeping all LEDs off.
   - Display the second input number on the LEDs for 5 seconds and continue to display **"NEXT INPUT"** on the LCD. Also, store the second input number in the same array. Give the next input through DIP switches during this time.
   - Repeat the above process for the next three numbers as well and place each of them in the array.
   - While the last input is being displayed on the LEDs for 5 seconds, display **"SORTING..."** on the LCD.

- After the last input is displayed, pause for 1 second keeping all LEDs off and display **"SORTING"** in the first row, **"COMPLETED"** in the second row.
- Sort the five numbers in ascending order and display it on LEDs, each for 5 seconds with a pause of 1 second with all LEDs off.
- After displaying the sorted array, give a pause of 1 second with all LEDs off. Then turn on all the LEDs for 5 seconds where you have to take the input to be searched and display **"NUMBER TO BE"** in the first row and **"SEARCHED"** in the second row of the LCD.
- Give a pause of 1 second by clearing both LEDs and LCD.
- Perform any search algorithm on the sorted array of five numbers.
- If the number is present in the array, display the index at which the number is present (assume array index starts from 1) on the LEDs and display **"THE INDEX IS"** on LCD.
- If the number is absent, ON all the LEDs and display **"NUMBER"** in the first row and **"NOT FOUND"** in the second row of the LCD.

TA Checkpoints

- Check the understanding of LCD operation and the `lcd.c` code.
- For question 1, check that the text is properly centered.
- For question 2, check the working of the experiment using the following test cases.
  1. Input array = $\{8, 2, 15, 12, 4\}$
     Item to be searched = 2
  2. Input array = $\{7, 6, 13, 9, 11\}$
     Item to be searched = 14

EE 337: Microprocessors Laboratory (Spring 2023)
Indian Institute of Technology Bombay
Lab 7: 20 points Date: March 1, 2023

In this lab, you will be interfacing a keypad with Pt-51 using Embedded C.

1. [20 points] The task is to write an embedded C program to read password using keypad and grant (or deny) access based on the correctness of the password.

   The flow-chart below describes the algorithm to read any key from the keypad.

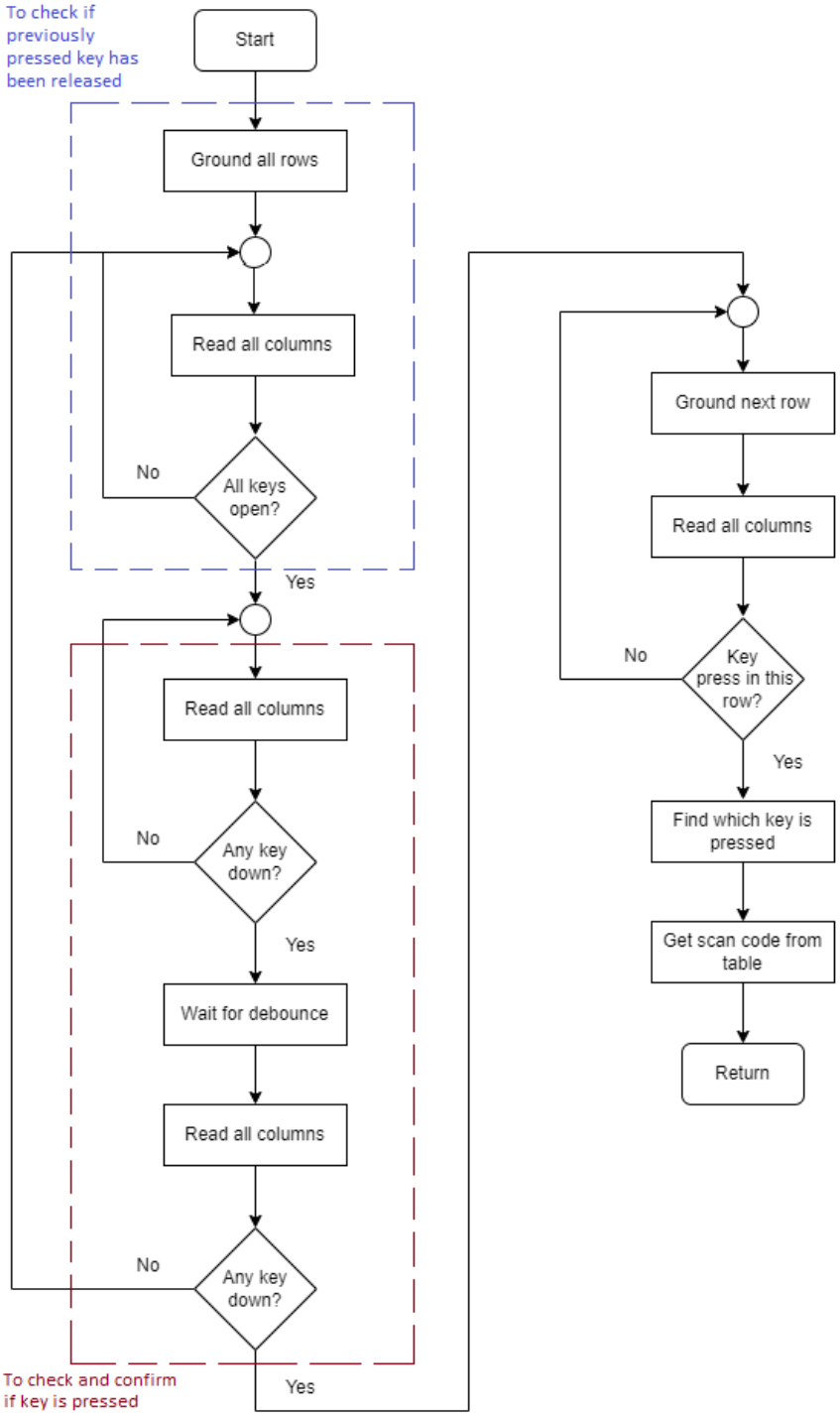


Figure 1: Flowchart describing procedure for reading keys

Note: Keep debounce delay of 20 ms.

The internal circuit diagram of the keypad is given in the below figure.

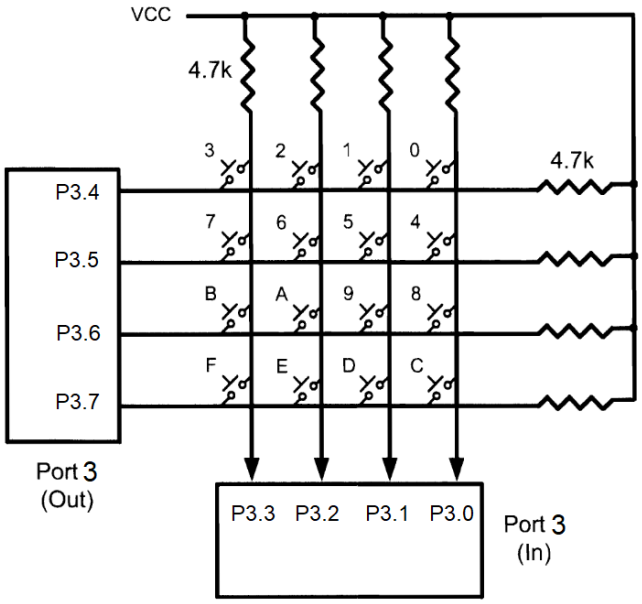
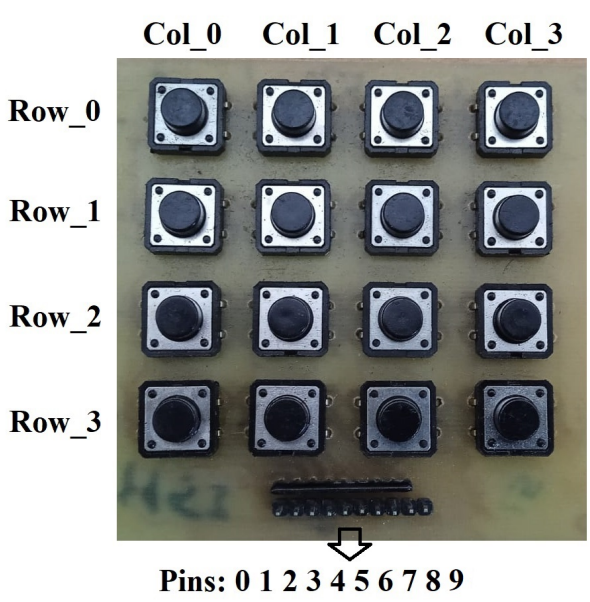


Figure 2: Keypad Circuit

Note that the rows are outputs and columns are inputs.

Refer the below figures to understand the keypad.



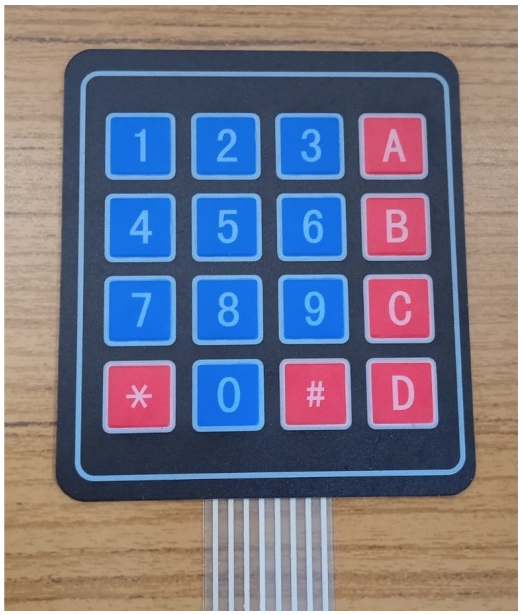


Figure 3: Keypad Mapping

The image on the right describes which key corresponds to which character. For example, the key at Row 0, Column 0 corresponds to the character '1'.

Refer to the following table for pin mapping.

Pins	Mapping	Pt-51
Pin0	Row 3	P3.7
Pin1	Row 2	P3.6
Pin2	Row 1	P3.5
Pin3	Row 0	P3.4
Pin4	Column 0	P3.3
Pin5	Column 1	P3.2
Pin6	Column 2	P3.1
Pin7	Column 3	P3.0
Pin8	Pull-up resistance	$5V$ on board
Pin9	Not connected	-

Table 1: Pin mapping

Note : For $5V$ connect to the on board power supply pin near USB attach/detach switch.

Task procedure-

- The correct password is - "15A8*D6#". Store this as a string in your program.
- Print "Enter Password:" in the first row of LCD.
- Use the flowchart in Figure 1, the images in Figure 2 and Table 1 to write code to read an input from the keypad.
- Create a loop that iterates 8 times and reads a character from the key-pad in each iteration. As a new character is read, it should be displayed on the second row of the LCD. Also store each of these characters in a string.
- After all 8 characters are read, compare the entered password to the correct password.
  - If the password is correct, display - "Correct Password" in the first row and "Access Granted" in the second row.
  - If the password is wrong, display - "Wrong Password" in the first row and "Access Denied" in the second row.
- Before trying on board with the keypad verify the correctness of the code by using debug session on Keil with breakpoints using I/O peripherals $\rightarrow$ Port3 to give inputs. **When key is pressed the output given to the row value gets reflected on the corresponding column value**.
- For verifying that all the keys in the keypad are working, you can use the hex file provided here : Keypad_Test

TA Checkpoints

- Check that 8 characters can be read from the keypad and each character is displayed on the LCD after it is entered.
- Enter the correct password and check if the proper messages are being displayed.
- Enter some random wrong password and check if the proper messages are being displayed.

EE 337: Microprocessors Laboratory (Spring 2023)
Indian Institute of Technology Bombay
Lab 8: 20 points Date: March 13, 2023

In this lab, you will be learning how to use Timers and Interrupts in Embedded C.

To learn about timers, refer the following document: 8051 Timers
To learn about interrupts, refer the following document: 8051 Interrupts

1. [5 points] Fill in the blanks in the Embedded C code at `https://ee337.github.io/2023/downloads/lab8_1.c` to create unsymmetrical square wave at port P3.6 using timer 1 in polling mode. The square wave should have ON time 4 ms and OFF time 12 ms.
   Observe the output using Keil Logic Analyzer first. Then upload the code on the Pt-51 board and use a Digital Storage Oscilloscope in lab to observe the output.

2. [5 points] Fill in the blanks in the Embedded C code at `https://ee337.github.io/2023/downloads/lab8_2.c` to generate two square waves simultaneously at P3.6 and P3.7 pins of time periods 2 ms and 3 ms using timer 0 and timer 1 respectively in interrupt mode.
   Observe the outputs using Keil Logic Analyzer first. Then upload the code on the Pt-51 board and use a Digital Storage Oscilloscope in lab to observe the outputs.

3. [10 points] Write an Embedded C code to create a **STOP-WATCH** using external event counter and LCD. Steps to perform this task are elaborated.

   - Use timer 0 as an event counter. The timer should be configured to work in mode 1 and the pin P3.4 should be used as input to count the number of events.
   - Provide a $60Hz$ square waveform (with voltage levels $0V$ and $5V$) using an Arbitrary Function Generator to the pin P3.4.
   - Use the DIP switch P1.0 to start and stop the event measurement. When P1.0 is turned ON, the measurement should start and when it is turned OFF, the measurement should stop.
   - Use the LCD to display the measured time in the following format - `MM:SS`. Note that you need to count the number of seconds and minutes.

TA Checkpoints

- Check whether proper ON time and OFF time is obtained for the square wave in Logic Analyzer. Also, check whether the output in DSO is proper.
- Check whether students have set proper TH,TL, TMOD and enabled interrupts appropriately for generating square waves simultaneously.
- Check whether stop-watch is working as expected in LCD and TH,TL,TMOD and interrupt values is set appropriately.

EE 337: Microprocessors Laboratory (Spring 2023)
Indian Institute of Technology Bombay
Lab 9: 20 points Date: March 15, 2023

In this lab, you will be learning about Serial Peripheral Interface (SPI), which is a serial communication protocol. Refer to Prof. Dinesh Sharma's slides and notes on SPI given here- `https://ee337.github.io/dks.html#serial-io`

Use the given `main.c`, `spi.c`, `mcp3008.h` and `lcd.h` in the `https://ee337.github.io/2023/downloads/lab9_spi.zip` zip file to complete this lab. **Make sure that all the DIP switches are OFF.**

1. [10 points] In the first part, you need to use the serial port interface (SPI) to interface an analog-to-digital converter (ADC) MCP3008 with the 8051 micro-controller. The MCP3008 is part of the lab kit. Use this setup to measure the input given from a potentiometer.

   i) The micro-controller is to be configured for serial communication with the ADC MCP3008. Complete the function `spi_init` in `spi.c` to configure the SPI so that the micro-controller is the master and the ADC is the slave.

   ii) Test the ADC setup using a potential divider circuit (series of resistors provided in the kit can be connected between supply and ground terminals) as shown in Figure 1. By varying the point at which the output is taken, voltages ranging from 0 to 3.3 V can be obtained as output.

   iii) This output is connected to `CH4` of the ADC. By compiling the project and running the code on the Pt-51 kit, the measured output voltage will be displayed on the LCD in the format shown.

   ```
   Volt : 03229 mV
   ```

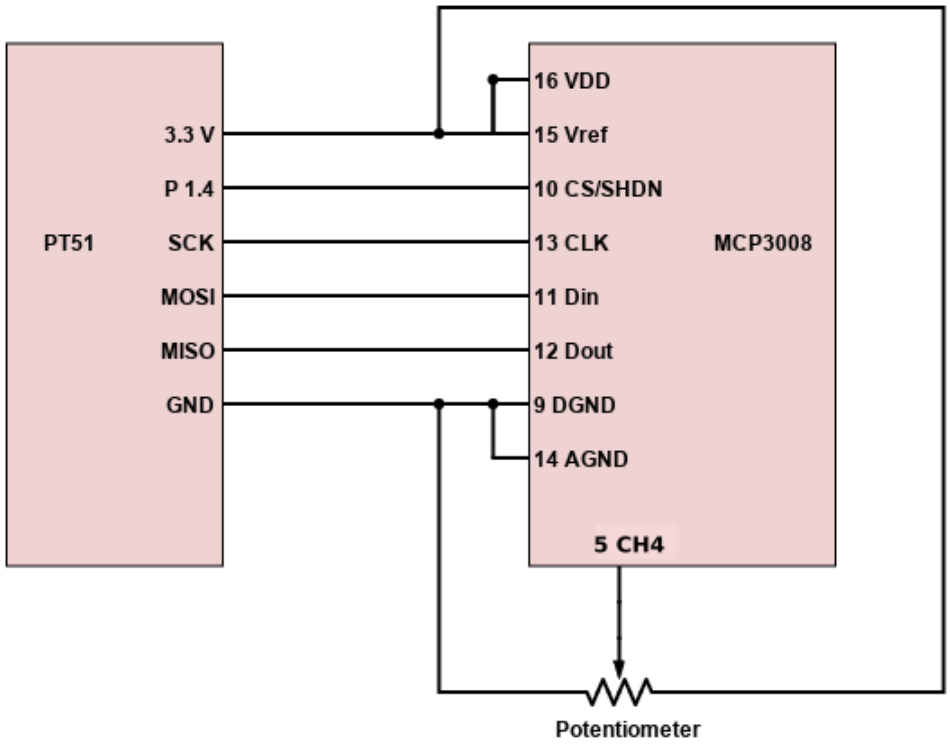


Figure 1: Pt51 interfacing with Potentiometer and ADC

2. [10 points] In the second part, you will use the same setup to measure the input given from arbitrary function generator (AFG).

   i) Configure the function generator to provide a sinusoidal waveform of some frequency less than 5 Hz and amplitude 0 V to 3 V.

   ii) Make the connections as shown in Figure 2. Note that the output of the AFG is connected to `CH4` of the ADC.

   iii) By compiling the project and running the code on the Pt-51 kit, the measured output voltage will be displayed on the LCD in the same format.
   Observe that the displayed voltages represent a sinusoidal signal. You can see values ranging from 0 mV to 3003 mV continuously varying.

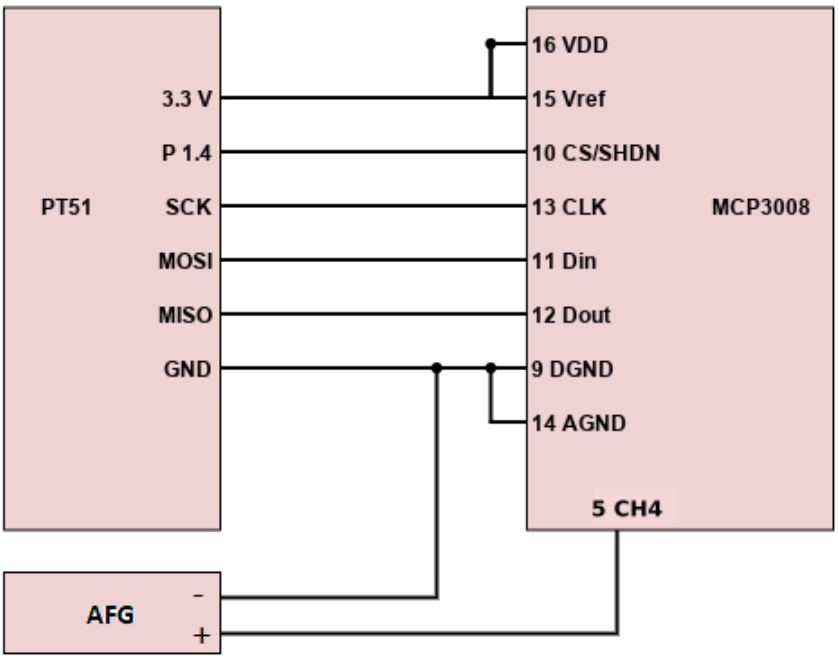


Figure 2: Pt51 interfacing with AFG and ADC

TA Checkpoints

1. Verify that the student can demonstrate the SPI setup to measure the voltage across the potentiometer and display it on the LCD.

2. Verify that the student can demonstrate the SPI setup to measure the sine wave generated by the AFG and display it on the LCD.

3. Make sure that the content on LCD is displayed in the specified format.

EE 337: Microprocessors Laboratory (Spring 2023)
Indian Institute of Technology Bombay
Lab 10: 20 points Date: March 22, 2023

In this lab, you will be learning about Universal Asynchronous Receiver-Transmitter (UART), which is a serial communication hardware. Refer to Prof. Dinesh Sharma's slides and notes on UART given here- `https://ee337.github.io/dks.html#serial-io`

1. [20 points] In this question, you will understand communicating between Pt-51 and a computer using UART. You need to send values from Pt-51 board to your computer, display those values on your computer and store it in a file.

   i) To connect the kit to a computer, you will be using the USB-to-UART adapter `Prolific PL2303` (Figure 1) that is part of 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 inputs from the board and capturing to text file in PC 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 Figure 2.

   For Windows, download `Realterm` at: `https://realterm.i2cchip.com`
   For Linux, you can download `Putty`.

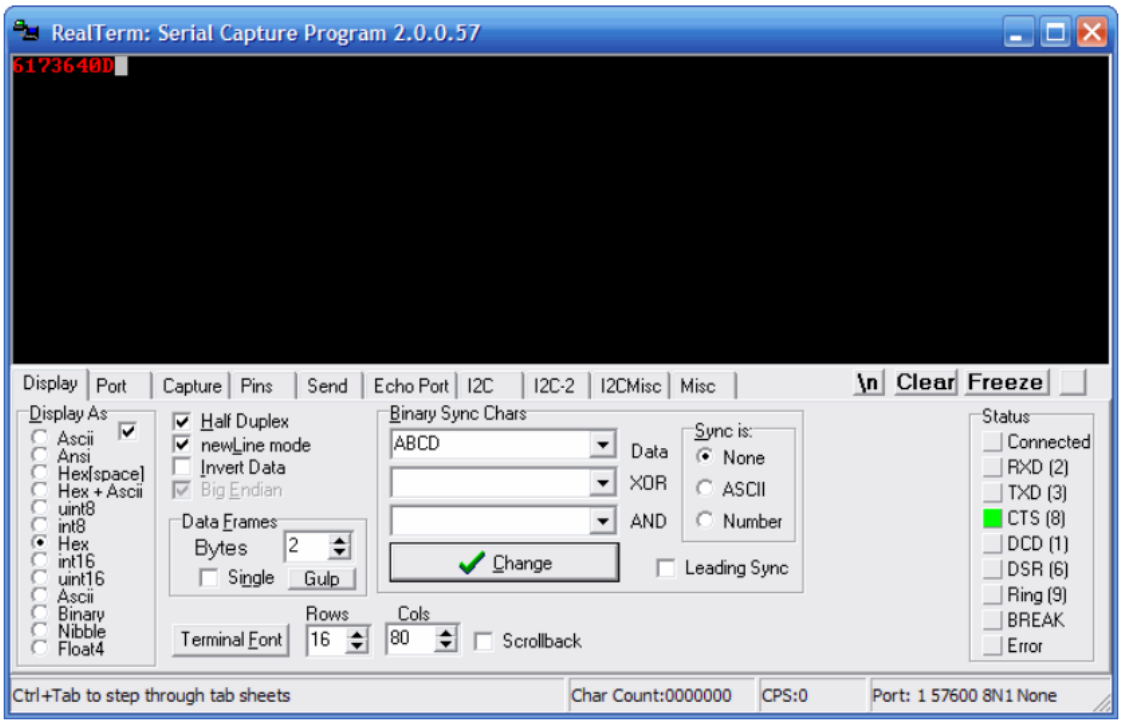


Figure 2: RealTerm: Serial/TCP Terminal

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 (the baud rate you have used in your program). Then click on `Open`. A screenshot of `Realterm` window with port and baudrate setting is shown in Figure 3.

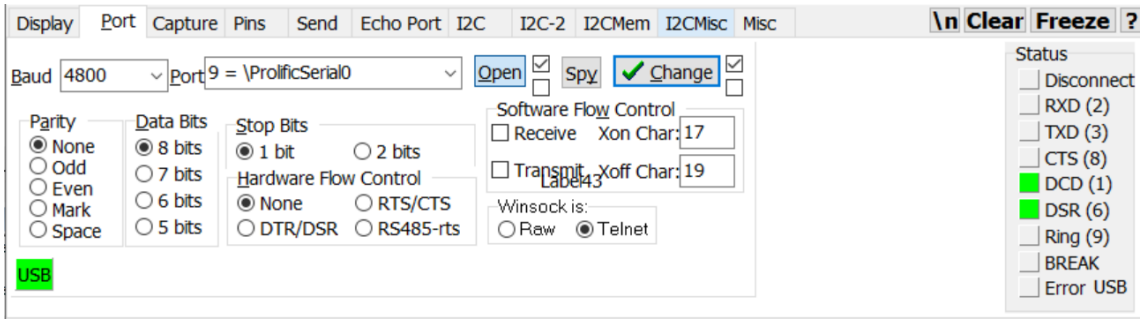


Figure 3: RealTerm: Baudrate

v) **Use the same setup designed in previous lab to read sinusoidal signal from AFG onto Pt-51 board using SPI protocol**. With the setup connected to the computer using the USB-to-UART adapter, you will use it to get the digitized input printed on LCD to display it 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 present in Lab10_UART folder.

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.
2. Capture the values sent to the computer via UART using real term serial capture. Refer to Figure 4 to capture the values and save them in a text file.
   "Start Overwrite" will start saving the values printed on RealTerm to the text file (path specified). "Stop Capture" will stop saving the values.

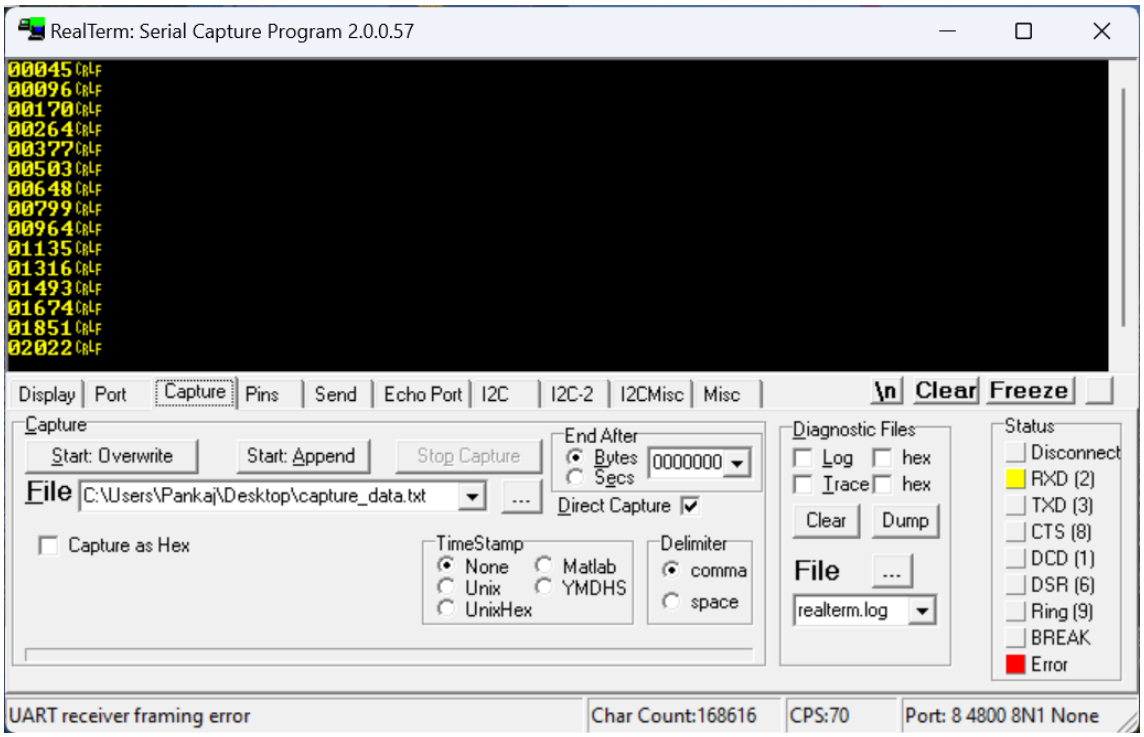


Figure 4: RealTerm: Serial/TCP Terminal

3. Plot the saved values using Python (or some plotting tool). You should be able to observe a sinusoidal wave. Note : The y-axis should be value and x-axis should be sample index.

TA Checkpoints

1. Check whether students have captured the input from board to the text file and all values are obtained properly.
2. Check whether students have plotted the sinusoidal wave output.