

Refer to Prof. Dinesh Sharma's slides and notes on serial I/O to do this lab. The link is given here: https://ee337.github.io/dks/serial_io.html.

1. [10 points] In this lab 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 1200 (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.

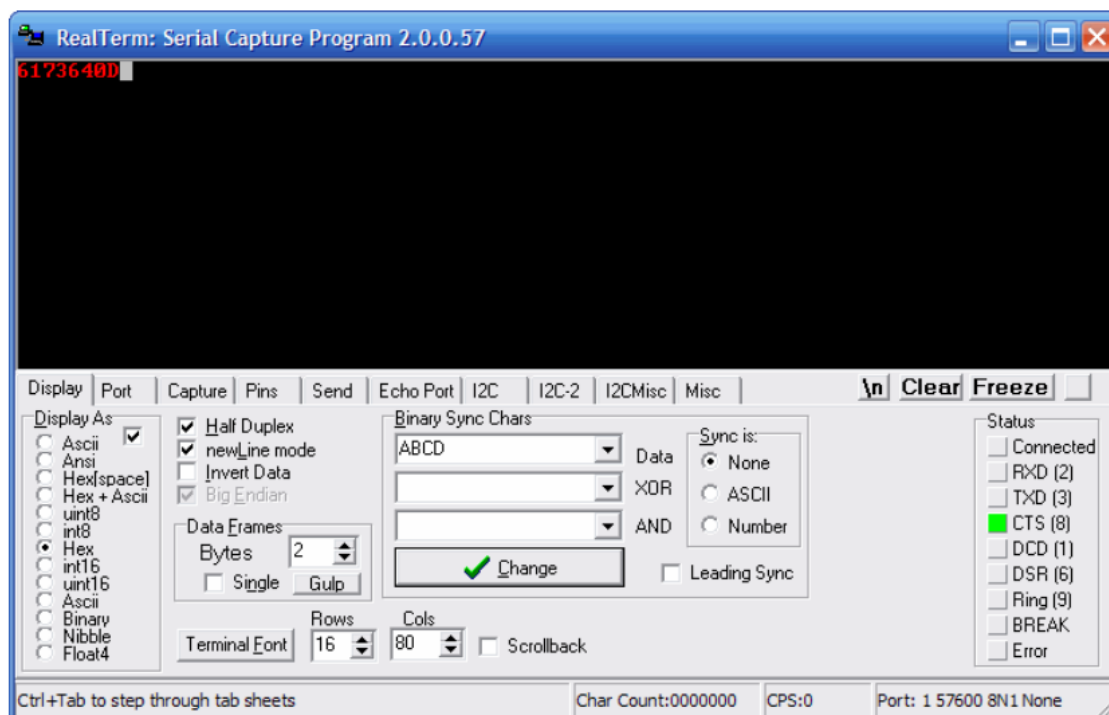


Figure 2: RealTerm: Serial/TCP Terminal

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 1200 bps.
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 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 "LCD Ok" 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 and display "LED tested" on the `Realterm` window. 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.

-
2. [10 points] In this question, you will be writing a program to simulate the behaviour of an ATM which can dispense notes having denominations 500 and 100. The actions of the ATM user will be simulated using key presses on a keyboard connected to Pt-51 using UART with baud rate 9600.

The denominations available are of Rupees 500 and 100 respectively. Assume there is an infinite pool of notes available. There are 2 accounts available with details as follows

- Account No.: 1; Name: Steven; Password: EE337
- Account No.: 2; Name: Gordon; Password: UPLAB

Assume both accounts have Rupees 10,000 available initially. You need to take care of the following cases. There are **three** states of operation.

1. Initial state

- Upon reset, the kit should enter the initial state.
- Whenever the kit is in the initial state, the UART display must show the following message:
> “Press A for Account display and W for withdrawing cash”

HINT: Use “\n” for encoding a newline character at the end of your display messages. For Windows, use “\r\n”.

2. State A: Account display

- This state is for displaying account balances. It is entered when the user types the character “A” or “a” through UART. When this happens, UART display must show the following message
> “Hello, Please enter Account Number”
- When a **valid** account number is entered, the following information should be displayed on UART.
> “Account Holder: <insert name>”
> “Account Balance: <insert figure>”
- For instance, if the user typed 1, then the information displayed would be
> “Account Holder: Steven”
> “Account Balance: 10000”
- For an **invalid** account number, the following must be displayed
> “No such account, please enter valid details”
- Irrespective of whether the account number is valid or invalid, the kit should **go back to the initial state.**

3. State W: Withdraw cash

- This state is for withdrawing cash. This state is entered by pressing “W” or “w” in the initial state. After this state is entered, UART window should display
> “Withdraw state, enter account number”
- The response to an account number entry must be same as described in state A. That is, for a valid account number the account holder and balance are displayed.

For an invalid account number, the error message is displayed.

- For a valid account, the following must be displayed after displaying balance:
> “Enter Amount, in hundreds”
- Here, you must input exactly 2 digits. The requested amount would be that number multiplied by 100. For instance, if I enter the number 42, the amount I want is Rupees 4,200. If I enter 07, the amount needed is Rupees 700 and so on.
- If the entered character is not a number, then display
> “Invalid Amount”
In this case, the kit should **go back to the initial state.**
- For a valid number, check whether the account has sufficient balance. If not, then display
> “Insufficient Funds”
Even in this case, the kit should **go back to the initial state.**
- If a valid number is entered for a valid account with sufficient balance, then
 - The **MINIMUM** number of notes required to pay the requested amount should be calculated.
 - The minimum number of notes should be displayed on UART (as described below)
 - The amount must be deducted from the account.
 - Details of the number of 500 and 100 Rupee notes dispensed and the updated account balance should be displayed on the UART display as follows.
> “Remaining Balance: ZZZZ”
> “500 Notes: XX, 100 Notes: YY”

TA Checkpoints

1. Verify that the student can configure UART for the required baud rate.
2. Verify that the student can demonstrate the UART setup to measure communicate between PC (RealTerm) and Pt-51.
3. Verify that the student has implemented the ATM application with proper working of all 3 states of operation.