EE 337: UART interface with 8051
Lab 9

October 16, 2019

1 Objectives

1. To configure the Universal Asynchronous Receiver/Transmitter (UART) interface to perform asynchronous serial communication.

2. To develop a debugger to test for hardware and perform real time communication between the PC and 8051 using UART.

2 Introduction

In the previous lab session, we used Serial Peripheral Interface (SPI) communication to interface with an ADC/DAC. SPI is a high speed synchronous serial communication interface using 4 lines for communication, whereas UART is low speed asynchronous serial communication interface using 3 lines for communication. For further details on serial communication refer to Sections 1 and 2 in *Serial.pdf*.

3 Homework

Write a C program to configure the micro-controller to use the UART to transmit a character continuously. A possible template for this is also attached. To achieve this do the following steps:

1. Configure timer1 (T1) in mode 2 to generate a baud rate of 1200. Mode 2 is 8 bit auto reload mode of timer which does not put any load on the processor. Refer to Section 4 in *Serial.pdf*.

2. Configure the serial port for 8 bit data. Serial port interrupts are to be enabled. Refer to Sections 3.1, 3.2, 3.3 and 5 in *Serial.pdf*

3. Write an interrupt service routine for serial communication, which clears TI and transmits the character A whenever the serial port interrupt occurs and TI is found set. (Then you will be able to use this routine for any character, not just A). After writing

the character, toggle an on-board LED (within the ISR) so that we are able to identify that a frame of data is transmitted.

4. You can observe the transmission of character/string in Keil Debugger window, go to View $->$ Serial Windows $->$UART. Also, you can observe SCON, SBUF, Baudrate, etc at Peripherals $->$ Serial.

Refer to Homework.c for template code. Also, consider serial.c for initialisation of UART.

4 Lab Work

1. Observe the frame (the last question in the homework) being transmitted using an oscilloscope, and identify all bits in a frame of the waveform.

2. Write a program to communicate between Pt-51 and PC. For communicating with PC, you will need an USB to UART (TTL) adapter (Prolific PL2303). Prolific.exe is already attached in zip file. Also, the document for the driver installation is given in the link,
urlhttp://www.miklor.com/COM/UV$_D rivers.php$
or

   For linux users, you can download putty.

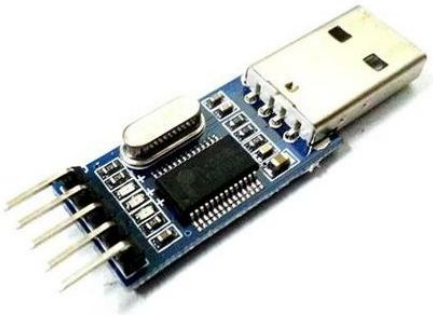

Figure 1: USB to UART adapter.

Port pin P3.0 is the serial data input (RxD or receive data line) and P3.1 is the serial data output (TxD or transmit data line) in the microcontroller. Transmit data line to kit should be connected to the receive data line of USB to UART (TTL) adapter and vice versa. Also the GND of both should be connected.

To display the values on PC. Use the software 'Realterm' or any other software used as a serial port terminal. Click on *Serial Port/ Port* tab. Select the port connected to USB to UART adapter and baud rate specified by your program and Click on Open. The link to download Realterm is:
https://realterm.i2cchip.com

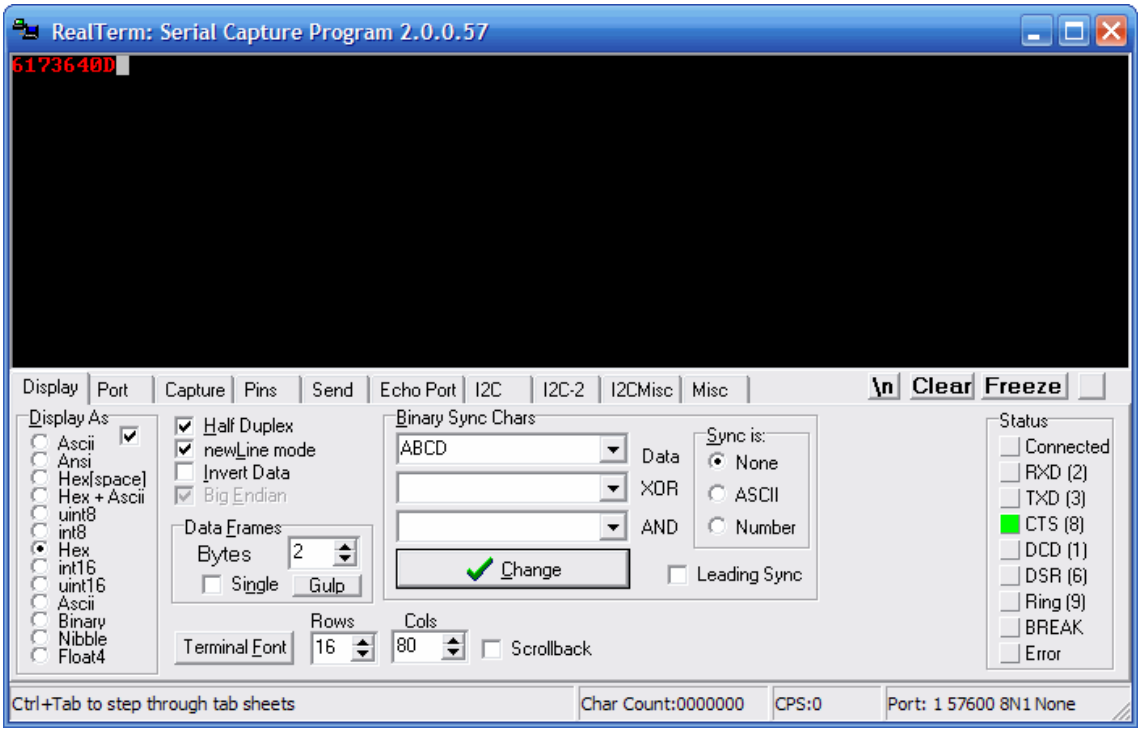

Figure 2: RealTerm: Serial/TCP Terminal

We will develop a debugging session by creating a menu to test for hardware. A communication will be established between PC and Pt-51 on UART where a user can send command from PC to test for any faulty hardware. The flow of the program is given below:

1. Write a function init serial() to configure the serial port and enable the corresponding interrupts.
2. Make a menu as shown below in Pt-51 and transfer the content through UART to USB to your PC. Write the below instructions in pt-51 and display the below menu on your PC screen in RealTerm.
   (a) Press 1 to test the LCD.
   (b) Press 2 to test the LED as GPIO.

(c) Press 3 to test the timer.

(d) Press 4 to test the switch as GPIO.

Now, Pt-51 should be responsive to the instructions from PC's keyboard. Any key press on PC's keyboard should be transferred to Pt-51 through Serial port terminal over UART protocol.

3. When 1 is pressed, it should be detected by pt-51 and your code should display a message "LCD is tested" on LCD screen.
4. When 2 is pressed, it should be detected by pt-51 and your code should glow all four LEDs.
5. When 3 is pressed, it should be detected by pt-51 and your code should toggle a LED at a interval of 0.5 sec.
6. When 4 is pressed, it should be detected by pt-51 and your code should read the status of switches and glow the respective LED corresponding to it. If your switch combination is 1010 then alternative LEDs, i.e., first and third LED should be ON.
7. Repeat the above task.

Refer to main.c for template code for labwork. Also, consider serial.c and lch.c for initialisation of UART and LCD respectively.