EE 337: UART interface with 8051
Lab 7

October 4, 2018

1 Objectives

1. To configure the Universal Asynchronous Receiver/Transmitter (UART) interface to perform asynchronous serial communication.
2. To perform a real time communication between the PC and 8051 using UART (TTL).

2 Introduction

In the previous lab session, we used Serial Peripheral Interface (SPI) communication to interface with an ADC. 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 given below. 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 + Even Parity (11 bit frame). 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. Parity bit should not be hard coded for A. It should be evaluated by checking the parity flag. (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.

```
// Template for homework on UART
void ISR_Serial(void) interrupt 4
{
//ISR for serial interrupt
}
void init_serial()
{
//Initialize serial communication and interrupts
}
void main()
{
init_serial();
while(1);
}
```

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. Use keypad.c and lcd.c programs to display the key pressed on the keypad. You have done similar exercise in assemply.
3. Write a program to communicate between Pt-51 and PC. For communicating with PC, you will need a USB to UART(TTL) adapter on Prolific PL2303. The documents for the driver installation is given in the link `http://www.prolific.com.tw/UserFiles/files/PL2303_Prolific_DriverInstaller_v1_9_0.zip`

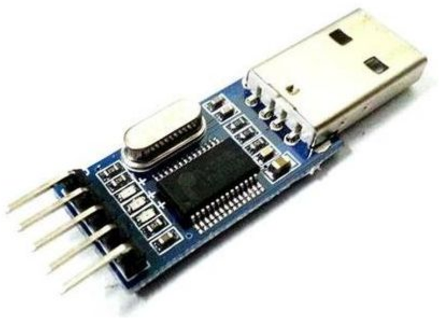
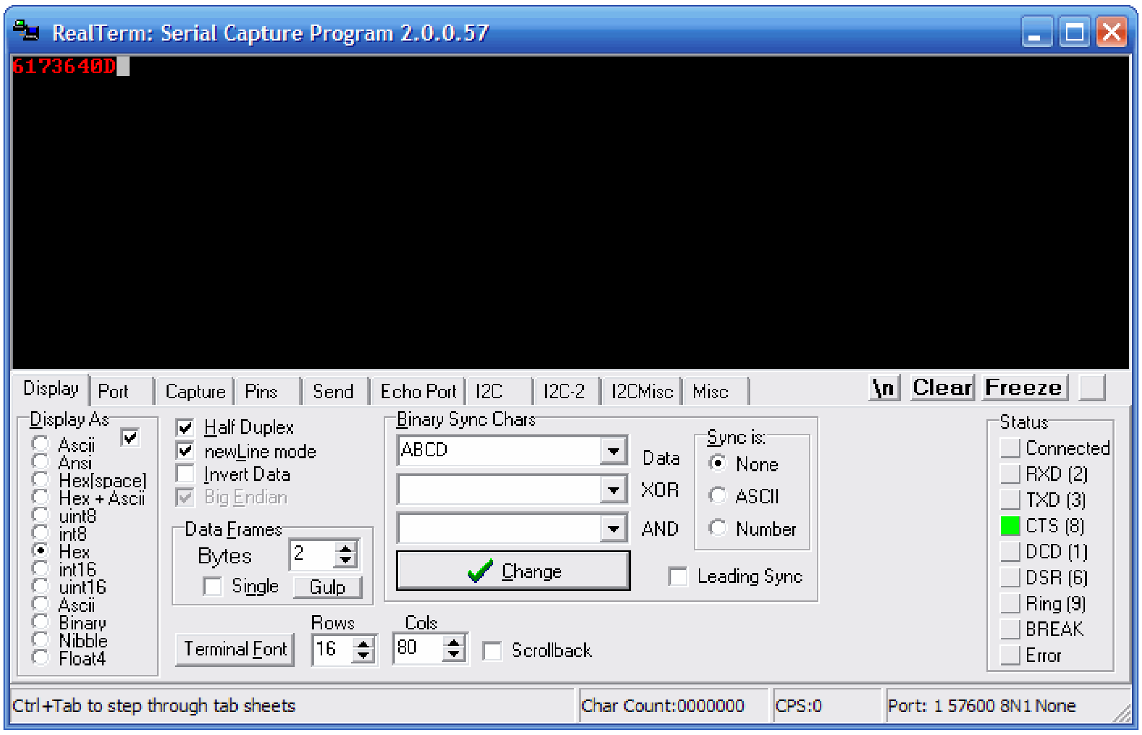


Figure 1: USB to UART (TTL) converter on Prolific PL2303

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). 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 (TTL) adapter and baud rate specified by your program and Click on Open. The link to download Realterm is given in the following link
https://sourceforge.net/projects/realterm/


RealTerm: Serial Capture Program 2.0.0.57

```
6173640D
```

Display As: Hex [✓]; Half Duplex [✓]; newLine mode [✓]; Invert Data [ ]; Big Endian [✓]
Data Frames: Bytes 2; Single [ ]
Binary Sync Chars: ABCD
Sync is: [✓ None] ASCII Number; Leading Sync [ ]
Rows 16; Cols 80; Scrollback [ ]
Status: CTS (8) 🟩

Ctrl+Tab to step through tab sheets | Char Count:0000000 | CPS:0 | Port: 1 57600 8N1 None

Figure 2: RealTerm: Serial/TCP Terminal

The flow of the program is given below:

(a) Write a function init serial() to configure the serial port and enable the corresponding interrupts.

(b) Press Y for sending the data(the key pressed) from the terminal

(c) Wait till 'Y' is pressed(from the keyboard of your pc)

(d) When 'Y' is pressed, read the key pressed (as done in keypad experiment).

(e) Display the keys pressed on PC

(f) To stop sending the data (the key pressed)press N

(g) If ‘N’ is not pressed wait for 2 sec (no need of timer. You can use ’for’ loop) and accept the next data.

(h) Continue reading key pressed, till ‘N’ is pressed

(i) If ‘N’ is pressed stop.

```
// Functions to be used for UARTs
void ISR_serial(void) interrupt 4
{
//ISR for serial interrupt
}
void init_serial()
{
//Initialize serial communication and interrupts
}
unsigned char receive_data(void)
{
//function to receive data over RxD pin.
}
void transmit_data(unsigned char str)
{
//function to transmit data over TxD pin.
}
void transmit_string(char* str, n)
{
//function to transmit string of size n over TxD pin.
}
```

Note: The code given in ’keypad.c’ file returns character ’0’ by default. You may modify the code accordingly so that no data is displayed if no key is pressed.