

EE 337: UART interface with 8051

Lab 7

September 13, 2017

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.
3. To record the cooling curve for a heating element given to you.

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.

4. Observe the frame being transmitted using an oscilloscope, and identify all bits in a frame of the waveform.

```
// 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. Write a program to communicate between 8051 and PC. For communicating to PC, you will need a USB to UART 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

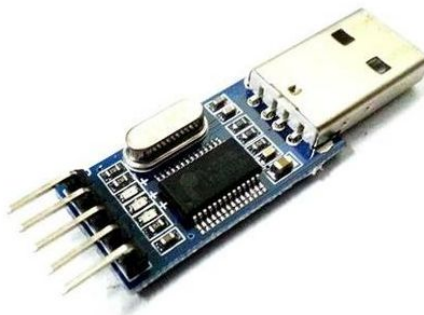


Figure 1: USB to UART converter on Prolific PL2303

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 adapter and vice versa. Also the GND of both should be connected.

To display the values in 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 given in the following link

<https://sourceforge.net/projects/realterm/>

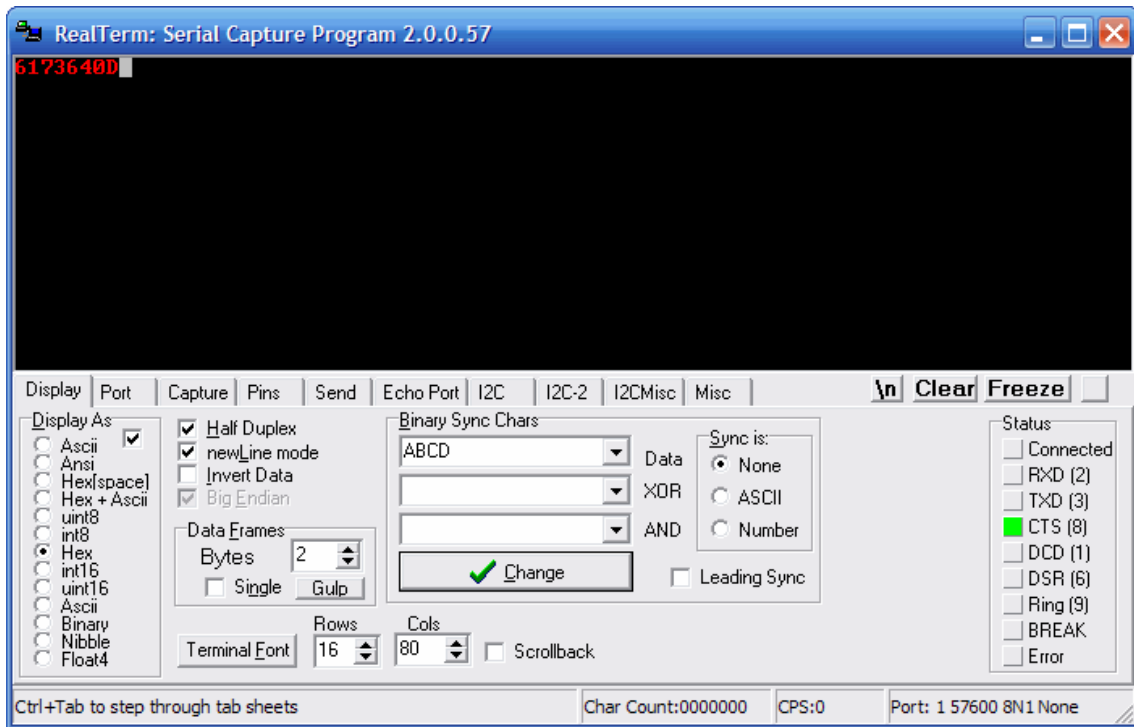


Figure 2: RealTerm: Serial/TCP Terminal

The flow of the program is given below:

- Write a function `init serial()` to configure the serial port and enable the corresponding interrupts.
- Sent a string to PC "Press Y to start sampling the data"
- Wait till 'Y' is pressed
- When 'Y' is pressed, read the temperature value from LM35 using SPI (as done in previous experiment)
- Display the value on PC
- Display "To stop sampling press N"
- If 'N' is not pressed wait for 2 sec (no need of timer. you can use for loop) and sample the next data

- (h) Continue sampling till 'N' is pressed
 - (i) If 'N' is pressed stop the sampling.
2. Heat the heating element to 60 °C. Switch off the power supply and start sampling the temperature (by pressing 'Y' in the Realterm terminal). Stop the sampling after 5 minutes (by pressing 'N' in the Realterm terminal). Plot the cooling curve, ie. graph with time on x-axis and temperature on y-axis.

```
// Functions to be used for UART
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.
}
```