EE337: UART interface with 8051
Lab 7

26 September 2016

Objectives

- To configure the Universal Asynchronous Receiver/Transmitter (UART) interface to perform asynchronous serial communication.
- To communicate data from one 8051 kit to another using the UART interface.

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*.

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:

- 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*.
- 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*.
- 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.
- 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);
}
```

Lab Work

Write a program so that two 8051 kits can communicate with each other using serial communication. 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 of one kit should be connected to receive data line of the other kit and vice versa. Also the GND pins of both the kits should be connected. Each kit should send a 16 character string to the other. Each kit should display the string it received from the other on the first line of its LCD display and the string it transmitted on the second line. Write a program to do the following (possible template given below):

- Write a function `init_serial()` to configure the serial port and enable the corresponding interrupts.
- In `ISR_serial`, if data is received, display it on the LCD. Thus data is received in interrupt mode.
- Write a function `check_switch()` that reads an onboard switch every 500ms. If the recently read value of switch is different from previous value, transmit a 16 bit data to on the `TxD` pin.
- Display the string received on first line of the LCD and display the transmitted string on second line of the LCD.
- Thus, both kits are ready to receive from the beginning, but transmit only when permitted to do so by changing slide switches.

```
// Template for lab work on UART
void  ISR_serial(void) interrupt 4
{
 //ISR for serial interrupt

}

void init_serial()
{
 //Initialize serial communication and interrupts
}
```

```
int check_switch()
{
//function to check switches after every 500ms

}

void transmit_data(unsigned char str)
{
 //function to transmit data over TxD pin.

}

void main()
{
    init_serial();
    while(1)
    {
     //check switch value
     //if unequal
     // transmit data

    }
}
```