Serial Data Communication

Using RS-232

Birzeit University

Information Technology Faculty

Computer Systems Engineering Department

Abstract

This experiment aims at understanding and expanding the concept of Serial data communication over the Universal Asynchronous Receiver / Transmitter Controller (UART 16550). Besides, a main part of the experiment focuses on studying the RS232 protocol.

PART I Theoretical Introduction

In this introduction general point are sorted out about main topics related the serial communication.

Serial Communication:

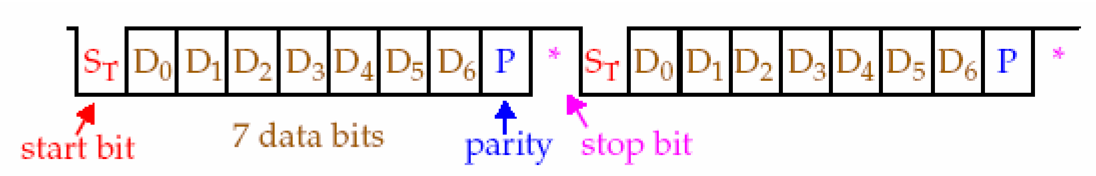


Figure 1 Serial Communication

- The data coming in at the receiving end of the data line in a serial data transfer bit by bit (Figure 1).
- It is difficult to make sense of the data unless the sender and receiver agree on a set of rules, a protocol, on how the data is packed, how many bits constitute a character, and when the data begins and ends.
- To allow compatibility among data communication equipment made by various manufacturers, an interfacing standard called **RS232** was set by the Electronics Industries Association (EIA) in 1960.

Data Transfer Rate:

- The baud rate and "**bps**" are the same, and for this reason the terms **bps** and baud rate interchangeably.
- The data transfer rate of a given computer system depends on communication ports incorporated into that system.

  For example, the early IBM PC/XT could transfer data at the rate of 100 to 9600 bps. However in recent years, PCs transfer data at rates as high as 19,200 bps.
- It must be noticed that in asynchronous serial data communication, the baud rate is generally limited to 100,000 bps.

RS232 and Other Serial I/O Standards:

- Since the standard was set long before the advent of the TTL logic family, the input and output voltage levels are not TTL compatible.

- In the RS232 a 1 is represented by -3 to -25 V, while the 0 bit is +3 to +25V, making -3V to +3 V undefined (Figure 2).

- For this reason, to connect any RS232 to a microprocessor-based system we must use voltage converters such as MAX232 to convert the TTL logic levels to the RS232 voltage level, and vice versa.

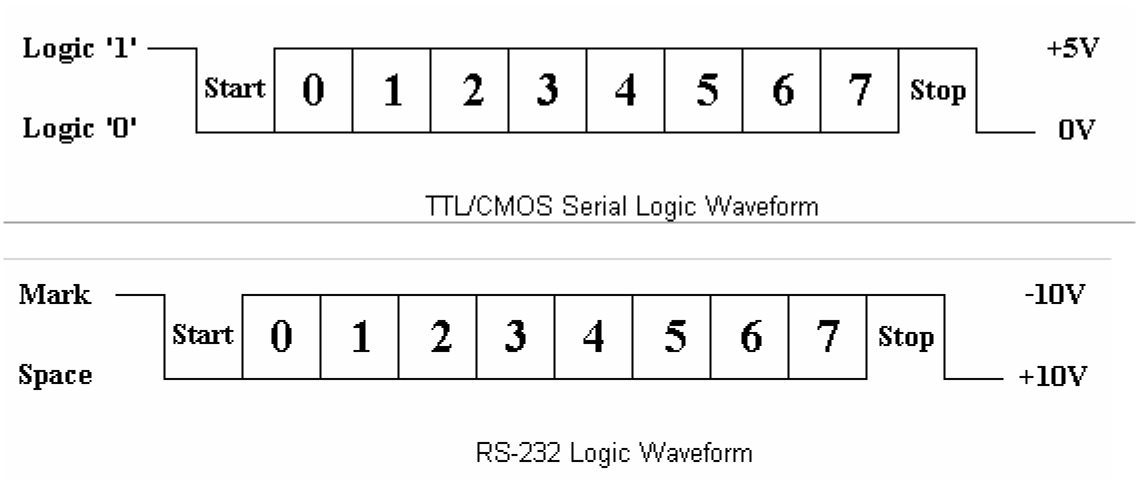


Figure 2 A Comparisons between RS-232 and TT/CMOS Waveform

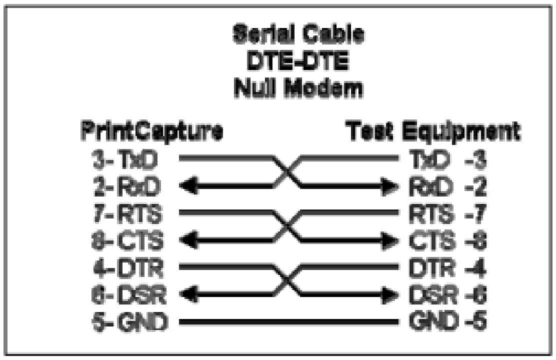
DTE and DCE

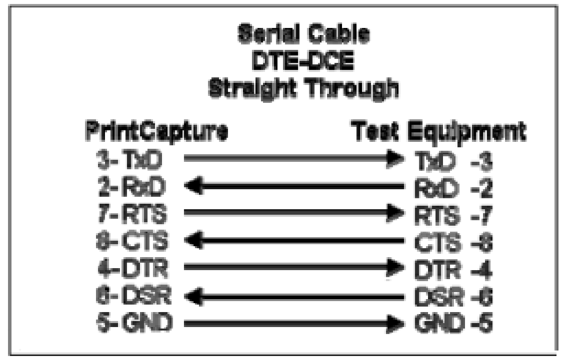
Two terms you should be familiar with are **DTE** and **DCE**. **DTE** stands for **Data Terminal Equipment**, and **DCE** stands for **Data Communications Equipment**. These terms are used to indicate the **pin-out** for the connectors on a device and the direction of the signals on the pins. Your computer is a DTE device, while most other devices are usually DCE devices. ()

- If you have trouble keeping the two straight then replace the term "DTE device" with "your PC" and DCE with “modems”

- DTE refers to terminals and computers that send and receive data.

- DCE refers to communication equipment, such as modems, that are responsible for transferring the data.





Programmable Communications Interface (16550)

In PC's there is a serial port uses RS-232 protocol decoded as Shown in (Figure 3) Note that the base address is **3F8h**.

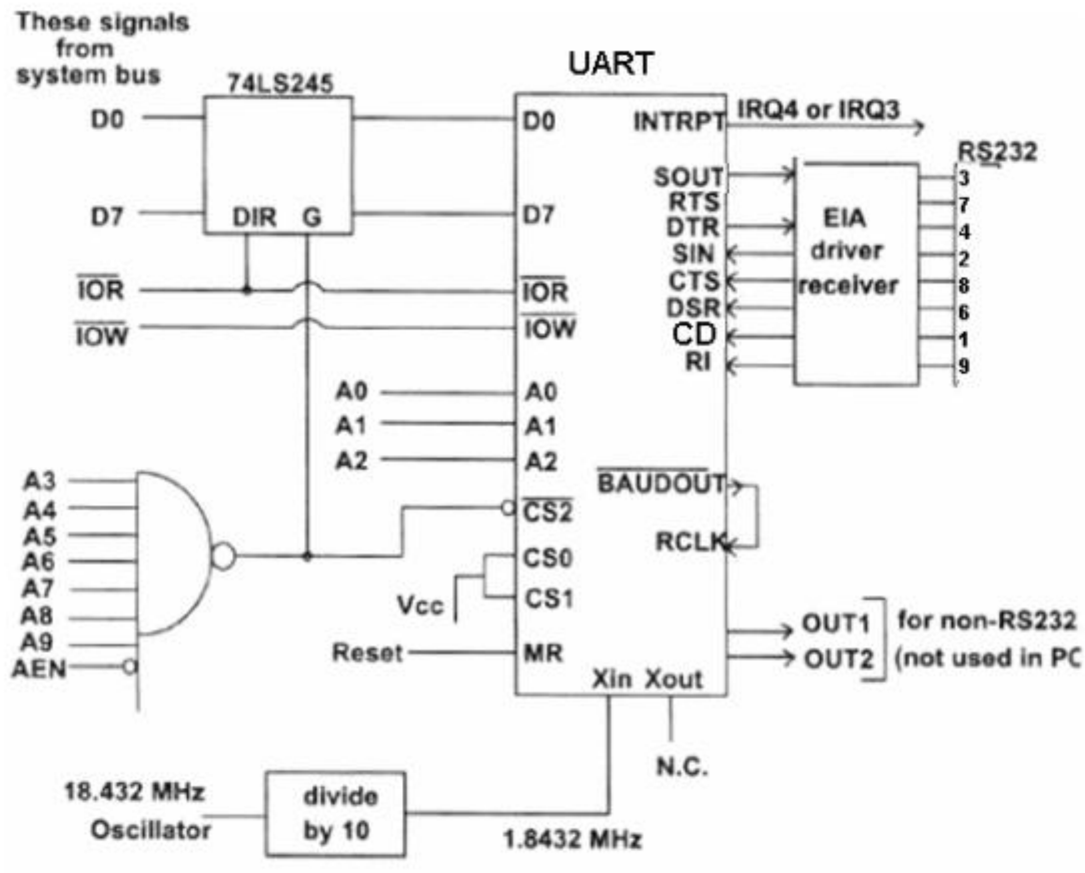


Figure 3 UART (16550) Interfacing Circuit in Personal Computers

Table 1 UART Addresses in PC

Name	Address	IRQ
COM 1	3F8	4
COM 2	2F8	3

Table 1 shows the UART address in personal computers.

From Figure 3 we note that A0, A1 and A2 select an internal register for programming and data transfer. For example, Line control register sets the number of data bits, number of stop bits and the parity, addressed at location 011. Notice that it will be at address (3F8+3=3FB). For the list of all internal registers see Table 2.

Table 2 UART Internal Registers

A1	A2	A3	Register Name
0	0	0	Receiver buffer (read) and transmitter holding (write)
0	0	1	Interrupt enable
0	1	0	Interrupt identification (read) and FIFO control (write)
0	1	1	Line control
1	0	0	Modem control
1	0	1	Line Status
1	1	0	Modem Status
1	1	1	Scratch

Programming Baud Rate

Baud rate generator is programmed with a divisor that sets baud rate of transmitter. Baud rate generator is programmed at 000 and 001. Port 000 used to hold least significant byte, 001 most significant.

For example; if we want the baud rate to be 9600 we use the following formula:

$$\text{Baud rate} = 1.8432M / (16 * \text{count})$$

$$9600 = 1.8432M / (16 * \text{count})$$

From the equation count= 12 = 000Ch so we will send 0C (least significant)on 3F8 and 00 (most significant) 3F9.

The following figures show some of the main internal registers of UART 16650.

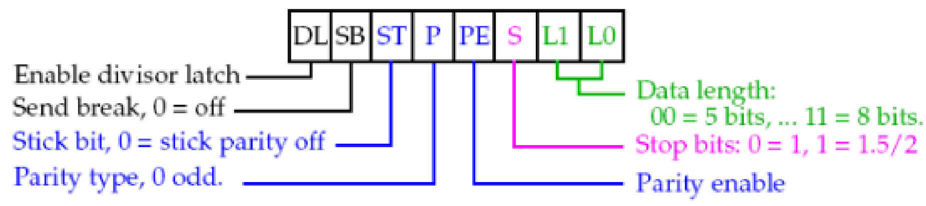


Figure 4 Line control register

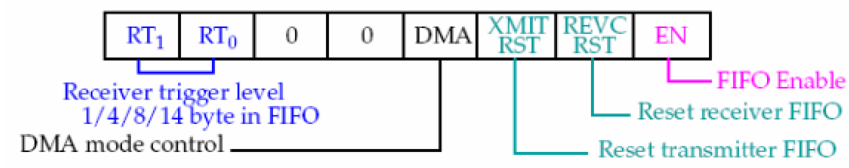


Figure 5 FIFO control register

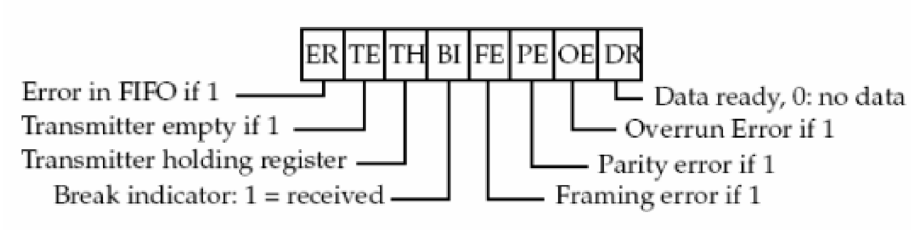


Figure 6 Status line register

Status line register (Figure 4) gives information about error conditions and state of the transmitter. This register needs to be tested in software routines designed to use the 16550 to transmit / receive data. Suppose a program wants to send data out. It needs to test the TH bit to determine if transmitter is ready to receive data. To receive information, the DR bit is tested.

In personal computers the connectors used are either DB9 or DB25, but new computers uses DB9 as shown in Table 3.

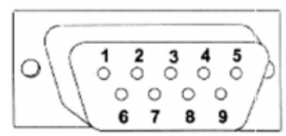


Table 3 D-Type-9 Pin Assignment

D-Type-9 Pin	Abbreviation	Full Name
Pin 1	CD	Carrier Detect
Pin 2	RD	Receive Data
Pin 3	TD	Transmit Data
Pin 4	DTR	Data Terminal Ready
Pin 5	SG	Signal Ground
Pin 6	DSR	Data Set Ready
Pin 7	RTS	Request To Send
Pin 8	CTS	Clear To Send
Pin 9	RI	Ring Indicator

PART II Pre-Lab

1. From Figure 3 how can you explain why the address of COM1 is 3F8h?
2. What are the values of 3F8 and 3F9 registers, when programming the UART to operate using baud rate to be 4800bps?

PART III Practices

3.1 PRACTIC I: Sending Data Over Serial Port

Step 1: Initialization

- Connect the oscilloscope to pin #3 (why pin #3?)
- Run debug and write the following commands

```
3fb 80
3f8 0c
3f9 00
3fb 3
3fa 7
```

- Comment each of these commands?
- What is the result of theses commands?

Step 2: After you initialize the serial port you can transmit your data
Write the following commands on the debug:

```
3f8 a
3f8 0
3f8 ff
```

Step 3: Try to adjust the oscilloscope till you can see the signal.

TASKS

1. Using the oscilloscope find the bit rate (1/bit transmission time).
2. Try to understand how the data is transited. (Analyze the data sent).
3. Repeat the previous steps changing the second instruction to be (O 3f8 18). What will change?
4. Again using oscilloscope find the bit rate (bit transmission time) and baud rate.
5. Repeat steps changing the instructions so that you use 7 bit data, odd parity, 1 stop bit, 4800 bps.

3.2 PRACTICE II: Coding

TASKS:

1. Write an assembly program that Initialize the serial port and send each character you press on the keyboard to the serial port. Also, it will read the data from the serial port if there is any. (Always check if the transmitter and receiver are ready)
    - Test your program using oscilloscope
    - Connect the TD (pin#3) to RD (pin#2) in the COM port. What is the result?

2. Rewrite the program in Task 1 in C/C++ language. (Hint: Use inportb, and outportb).
3. Connect two computers together through serial port (as in the figure 7) and run the program in Task 2 in the two Computers what is the result? Change the baud rate on one computer to be 2400 and in the other to be 9600. What is the result? Why? (for simplicity you can do this task using one computer, how?)

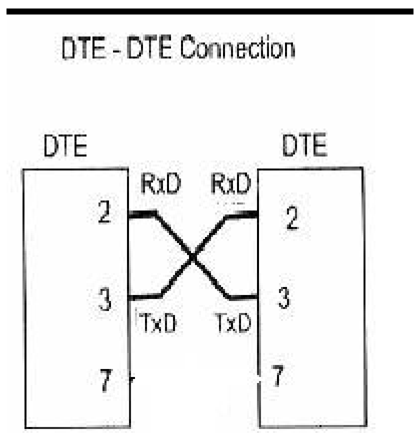


Figure 7 DTE – DTE Connection