Experiment
On
Serial communication

Objectives: To study serial data communication with RS-232.

After completing this experiment, you should know how systems communicate with each other with the RS-232 serial communication standard, and get familiar with technical terms such as data terminal equipment (DTE), data communication equipment (DCE), RTS signal, CTS signal, start-bit, stop-bit, Baud rate and etc.

Software: The SCom emulation package

Apparatus: Oscilloscope, 9-pin serial communication cables


Background
Serial transmission is one of the commonest and easiest forms of computer communication. The venerable RS232 standard is the most popular serial standard although there are many others used. RS232C has been used for several decades and is still viable. Serial communication requires that both ends of the link are set to the same speed and synchronized because information is interpreted in terms of digital fixed-length pulses. RS232 communication is also asynchronous, that is, each character can arrive at any time. In order to signal the start and end of a character, special start and stop bits are sent. There are a variety of standards for these bits. With modern computer equipment we commonly use one start bit and one stop bit. Speed in serial communications is measured in baud. A baud is an information bit per second. Thus sending 2400 baud, eight-bit ASCII codes with one start and one stop bit will send 2400/(8+1+1)=240 characters/second.
There are some basic terms used in RS232 standard. They are briefly introduced as follows.

**DTE, DCE**

Acronyms originating from days of yore; they expand to Data Terminal Equipment and Data Communication Equipment, respectively. DTE originally meant computer terminal and DCE meant a modem, but the definitions have been generously "expanded" (or warped) to include a huge variety of equipment that uses serial communication. A particular device may be either DCE or DTE, depending on its wiring, and it is possible (and sometimes necessary) to change the "sex" of a device from DTE to DCE, or vice versa. The connection of the signal pins between DTE and DCE is different from that between DTE and DTE.

**Handshaking**

Means of data flow control. Software handshaking involves embedding control characters in transmitted data. For example, XON/XOFF flow control works by enclosing a transmitted message between the two control characters XON and XOFF. Hardware handshaking uses voltages on physical wires to control data flow. The RTS and CTS lines of the RS-232 interface are frequently used for this purpose. Most lab equipment uses hardware handshaking.

**Baud Rate, Data Bits, Parity, and All That**

Serial communication using RS-232 requires that you specify four parameters: the baud rate of the transmission, the number of data bits encoding a character, the sense of the optional parity bit, and the number of stop bits. Each transmitted character is packaged in a character frame that consists of a single start bit followed by the data bits, the optional parity bit, and the stop bit or bits. Figure 1 shows a typical character frame encoding the letter "m" is shown here.

![Character frame diagram](image)

**Figure 1. An example of a serial data unit**
Method and details

In this lab, you will study how computers communicate with each other with their serial ports with an emulation package called SCom. Both computers act as DTEs in this lab. Figure 2 shows the graphical user interface (GUI) of the emulator. Through this GUI, you can select and configure a serial port (COM1 or COM2) of your computer system. You can type characters in the textbox provided and press the ‘Send’ button to send them out through the selected serial port. The serial port can also receive data from outside and store them in an internal buffer. Data in the buffer can be displayed by pressing the ‘Display’ button.

Procedures

1. Run SCom. Select your desired configuration setting of the serial port being tested. Type a character and send it out. Use an oscilloscope to inspect the output waveform of the serial port. Sketch the observed waveform on your logbook. Check it against with your configuration setting. With Reference to the ASCII table at Appendix, verify if the sent character is the character you typed in the textbox.

2. Quit and run SCom again. Select the default configuration this time. SCom will repeatedly send a string out. Use the oscilloscope to inspect the output waveform and sketch the waveform on your logbook. Guess the configuration setting of the serial port. Tell the
composition of a data frame, the parity scheme and the string.

3. Quit and run **SCom** again. Select your desired configuration and try the following.

3.1 Work with your neighbor group, connect your PIN 2 of RS-232 to your neighbor group PIN 2 and also PIN 3 to PIN 3. Set baud rate to 9600bps. Then type a character and send it out. Can you receive any character from your neighbor group? Why is that?

3.2 Again, work with your neighbor group, but this time connect your PIN 2 to your neighbor group PIN 3 and also PIN 3 to PIN 2. Set baud rate to 9600bps. Then type a character and send it out. Can you receive any character from your neighbor group? Why is that?

3.3 Hold the setting of your neighbor group. Change your baud rate to 2400bps. Then type a character and send it out. Can you receive any character from your neighbor group? Are they what you have sent? Why is that?

4. Based on the observation you have had in step 3, answer the following question.

Q1. What conditions should be satisfied, so that can make two DTE communicate successfully?

- END -