Microcontroller Serial Interfacing Lab

Purpose

- To introduce the serial protocol standard
- To develop familiarity with how to connect devices using the serial port
- To reinforce C program standards and coding documentation

Learning Objectives

Upon successful completion of this laboratory experiment, the student will be able to:

1. Identify what a ‘level shift’ chip is used for in RS232 communication.
2. Explain what TTL logic voltage levels are used for RS232 communication.
3. Describe where to find specifications for the RS232 communication protocol.
   a. Explain why the voltage levels for these serial protocols were chosen to be what they are.
   b. Describe the difference between ‘baud’ and ‘bytes per second” and describe why are they not the same.
4. Calculate and set Baud rate for the Atmel ATMEGA 128L processor.
5. Set the clock speed of the Atmel processor.
6. Verify, with the digital oscilloscope, that the bytes being transmitted between the microcontrollers match the data to be sent.
7. Verify the data rate of the transmission using a digital oscilloscope with a captured transmission.
8. Determine what baud rate will consume 30% of the microcontroller resources.
9. Classify the differences between hardware and software implementation of RS232 protocol on a microcontroller, and the advantages and disadvantages of each.
10. Design a circuit that will be able to use the microcontroller to turn an LED on and off based on an incoming transmission.
11. Use provided software codes to transfer serial communications on the STK 501 from one serial port to the second serial port along with code on the second STK501 to receive signals from the primary STK501 and react to these commands, as well as send responses back to the STK501.

Components

<table>
<thead>
<tr>
<th>Qty</th>
<th>Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>STK500</td>
</tr>
<tr>
<td>1</td>
<td>STK501 with Atmega 128L processor</td>
</tr>
<tr>
<td>2</td>
<td>Jumper cables</td>
</tr>
<tr>
<td>2</td>
<td>Serial port cable Male to Female (straight through)</td>
</tr>
<tr>
<td>1</td>
<td>USB to serial cable</td>
</tr>
<tr>
<td>1</td>
<td>10-pin ribbon cable with headers</td>
</tr>
</tbody>
</table>
Lab Preparation

The following pre-lab preparation is required to be able to successfully complete this lab.

- Complete reading “Atmega Microcontroller Summary.” (Pages: 1-7, 13, see course website)
- Complete reading Dr. Du’s handout “Communication Protocols and Interface”.
- Read data sheet for Atmel’s Atmega 128 Manual (Page 90-92, see course website) covering the setting of processor clock speed and calculating scaling bit for serial baud rates.

Lab Tasks & Procedures

Tasks

The Serial lab utilizes the following programs on the PC: Teraterm¹ or HyperTerminal, AVR studio², and a C compiler to generate code for an Atmel chip. The lab procedure includes:

(1) Set the clock and baud rate of the Atmega 128 and the corresponding baud rate of two Teraterm terminal windows.

(2) Write a simple program, with the guidance of lab TA, to pass back to one Teraterm window anything typed into the second Teraterm window.

Both (1) and (2) lab require you to use the primary as well as the secondary RS232 port on the STK501 as well as a USB to serial cable. Fig. 1 shows connection of the serial cable between the computer running Teraterm and the STK 500/501 mounted with an ATMEGA 128 processor.

![Figure 1 Laboratory setup for serial communication lab](image)

(3) Connect the STK501 secondary RS232 port to another STK501, as shown in Fig. 2. You will have to set up the clock speed along with baud rates between all devices to guarantee data integrity. You will download a precompiled hex file to the second STK501. You will send data to the first STK501 that will be forwarded to the second STK501 so that the second STK501 is able to turn on and off an LED as shown in Fig. 2.

¹ Teraterm and Hyper Terminal is a free software terminal emulator (communication program) for MS-Windows. It supports VT100 emulation, telnet connection, serial port connection, and so on.
² AVR Studio is the new professional Integrated Development Environment (IDE) for writing and debugging AVR applications in Windows 9x/NT/2000/XP environments. AVR Studio includes an assembler and a simulator. It supports the following AVR development tools: ICE50, ICE40, JTAGICE mkII, JTAGICE, ICE200, STK500/501/502 and AVRISP.
Procedure

1. Connect Power to the STK501
2. Connect a serial cable from the communication port 1 of the computer to the RS232 CTRL port on the STK500 main board. This will be used to download and debug code for this experiment.
3. Connect a second serial cable from the computer's communication port 2 to the spare RS232 port on the STK500 main board.
4. Place jumper wires from port PE pins 0 & 1 to RX & TX on the STK500. Verify that the wiring is correct. It should be port PE pin 0 to Rx and port PD1 to TX on the main board.
5. Open a HyperTerminal or Termaterm session configure it to Communication on port 2 setting set to baud rate to 9600, 8, 0, none.
6. Download program “Com1 test” to the ATMEGA128. If more information is needed on how to do this refer to “Introduction to the Atmel ATmega 128 Microcontroller”
7. Press STK500 reset button, after the board resets a message should appear in the HyperTerminal window.
8. Connect the third serial cable from the USB port to the RS232 port on the STK501 board on the STK501
9. Place jumper wire from port PD pins 2 & 3 to RX & TX on the STK500 and STK501 respectively. Verify that the wiring is correct. It should be port PD pin 2 to Rx and port PD3 to TX on the main board.
10. Open a HyperTerminal or Termaterm session configure it to Communication on port 3 setting set to baud rate to 9600, 8, 0, none.
11. Download program “Com2 test” to the ATMEGA128. If more information is needed on how to do this refer to “Introduction to the Atmel ATmega 128 Microcontroller”
12. Press STK500 reset button, after the board resets a message should appear in the HyperTerminal window.
13. Compile the following code with include files.
14. Download the code through the serial port on communication port 1 to the device.
15. Verify that you have two Hyper Terminals open, one for port 2 and the other for port 3. Review code for baud rates, parity bit and stop bit.
16. Press STK500 reset button, after the board resets the two serial ports are now setup as pass through configuration.
17. Establish communication from one terminal program through the STK501/500 back to the second terminal window.
18. Verify connection by typing a message in one terminal window and the message should appear in the other.
19. Now verify communication in the reverse direction by typing in the second window and see the message appear in the first window.

Questions (considering the following questions while doing the lab)

1) What effect, if any, would a resistance in the ground connection cause in RS232 communication?
2) Should there be current (e.g., maximum amperage) specifications associated with these protocols?
3) Identify as many things about the circuitry that will limit the maximum rate at which data can be reliably transmitted, and explain why. (e.g., what effect might the length of the wires have? What effect might capacitance between data transmission and ground have?)
4) Is it possible that the power supply could have an effect, and if so, what?
5) If these protocols are called ‘serial,’ give a simple explanation of what a ‘parallel’ protocol would be like.
6) What are the strong points of serial and parallel protocols compared with each other?