EE459 Lab Assignment 4
Serial I/O

1 Introduction

In this assignment you will add a serial interface to your project board and demonstrate that you can use it to establish a communication link between your board and a command-line interface (terminal) window on the computer. Providing the ability to send and receive text data on your project board can be a valuable tool later in the development of our project when it can be used for debugging purposes.

Most of the information you need for this assignment concerning serial communications is in the document "Using the Serial Communications Interface on the ATmega328P" that is available in the reference library portion of the EE459 website. You are strongly advised to read that document before starting to do the assignment.

2 Components

Each team will be provided with the following components:

- MAX232 IC for translating between TTL and RS-232 signals
- 16-pin socket
- Four 1μF electrolytic capacitors
- Male DB-9 connector
- Two 3/8" spacers

3 Hardware

Install the socket on the board in a place that won't get in the way of other components that may be installed later. Wire the MAX232 and capacitors as show in Fig. 1. Make sure the electrolytic capacitors are installed with the polarity correct. Note that three of them go in one orientation and the fourth is reversed.

The male DB-9 connector needs to be installed on the project board so it can be connected to the serial interface cables attached to the computers in the lab. The connector can be mounted somewhere on your board or you can have it on the end of a three-conductor cable and decide on some method of connecting those wires to your board, either permanently or just when the serial interface is being used. For mounting the connector on the board, one method is to use two 3/8" spacers with 4-40 screws running through them to mount the connector above the board as shown in Fig. 2. For making connections, the wires can be run through the holes in the board to the wiring side.

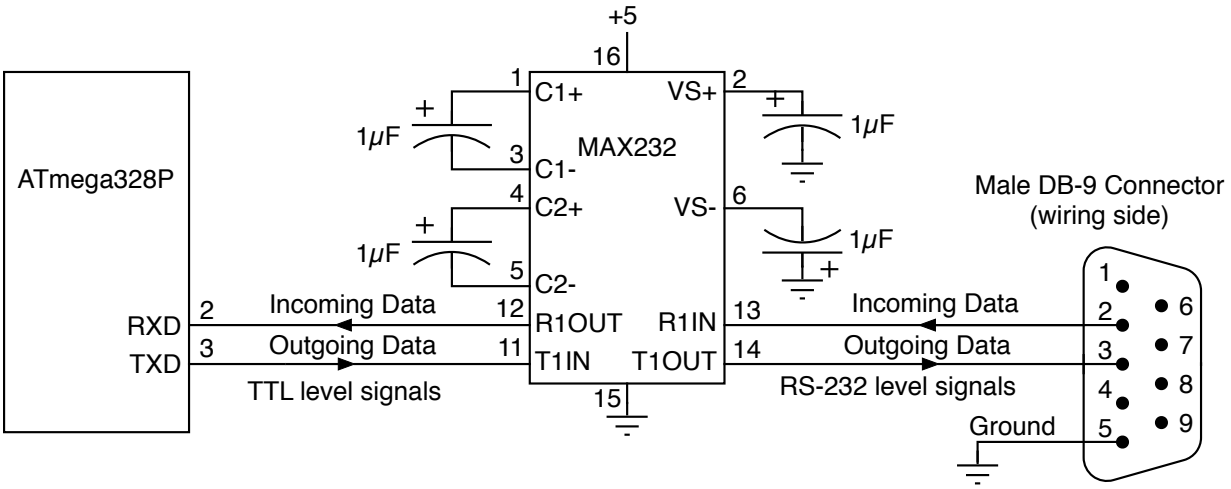

Figure 1: RS-232 to TTL converter circuit

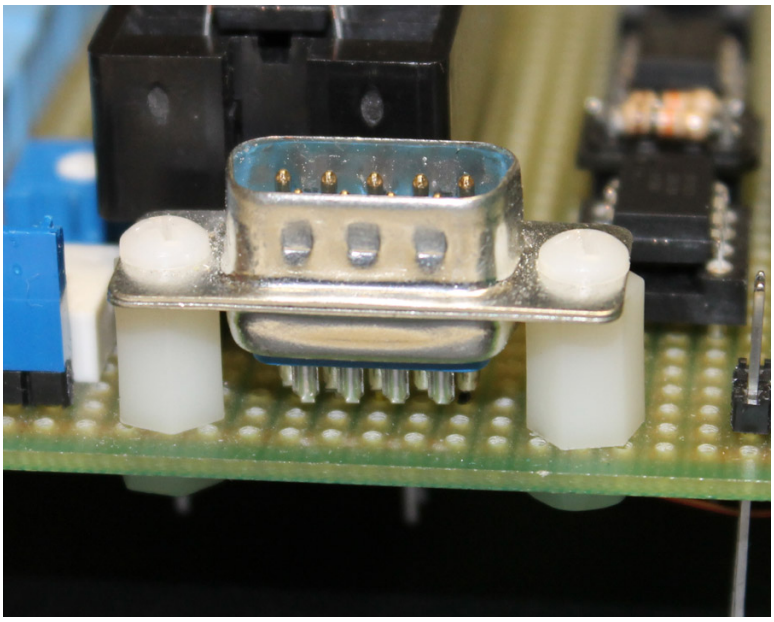

Figure 2: A male DB9 connector mounted on a project board using the two 3/8" spacers

4 Software on the Microcontroller

Information on the software needed on the ATmega328P microcontroller to use the serial port is available from the EE459 class website. At a minimum for this assignment you just need routines to initialize the serial port, write a byte out the port, and read an incoming byte.

5 Software on the Mac/PC/Linux System

Note: The following is not meant to be a definitive description of how to get the software working, nor is it the only way to do it. If you know of a better way, go ahead (and let us know so we can improve our notes.)

In order to use the USB serial adapters it will probably be necessary to install the USB drivers for it. Windows systems may be able to find the drivers and install them once the adapter is plugged into the system. For Mac, Linux and Windows systems, drivers can be downloaded from

```
http://www.ftdichip.com/Drivers/VCP.htm
```

Once the drivers are installed and the adapter plugged into the computer, the following can be used to open a command-line window and establish a communications link.

macOS

Find the device name of the USB serial adapter by opening a Terminal window, and with the adapter **not** plugged in, use the command "`ls /dev/tty.cu*`" to find all the files in the `/dev` directory that have names that start with `tty.cu`. Then plug in the USB serial adapter, run the command again and note what new files have appeared. The adapter's device file should now be present with a name like `/dev/cu.usbserial-A901C419`. From the Terminal window enter the command

```
screen name-of-device 9600
```

where "`name-of-device`" is the device filename you found above for the adapter device file. At this point anything you type in the window goes out the serial adapter.

When you are done with the serial connection, the cleanest way to terminate the process is to type control-a control-\ (holding down the Control key, type an 'a' and then type a '\').

Windows

Find the name of the port used by the adapter from the Device Manager (Start→Control Panel→System and Maintenance→System→Device Manager). Expand the Ports item, look for the USB Serial Port and make a note of the port name (e.g. COM4).

You'll need some sort of terminal program. Hyperterminal should work but may not be included in recent releases of Windows. A free alternative is PuTTY, available from

```
http://www.chiark.greenend.org.uk/~sgtatham/putty/
```

When PuTTY is started it puts up a configuration box where you can select the connection type. Select "Serial", fill in the port name from above, set the baud rate to 9600 and click on "Open". A terminal windows should open and anything you type in the window goes out the serial adapter.

Linux

Install a package like "`picocom`" or "`screen`" for doing serial communications. These should be available from the software repository for whatever Linux distribution you are using.

Find the device name of the USB serial adapter by opening a Terminal window, and with the adapter **not** plugged in, use the command "`ls /dev/ttyUSB*`" to find all the files in the `/dev` directory that have names that start with `ttyUSB`. Then plug in the USB serial adapter, run the command again and note what new files have appeared. The adapter's device file should now be present with a name like `/dev/ttyUSB0`. From the Terminal window enter a command like one of the following depending on which package you installed.

```
picocom --baud 9600 name-of-device
screen name-of-device 9600
```

where "`name-of-device`" is the device filename you found above for the adapter device file. At this point anything you type in the window goes out the serial adapter. When you want to terminate the connection, type control-a control-x (holding down the Control key, type an 'a' and then type an 'x').

6 Test Programs

Use the ATmega328P sample code from the serial communications handout to write the following short test programs. These can be written as separate programs or as one program with different parts commented out.

The tests below that involve sending data between the project board and a computer will require installation of the drivers for the USB serial adapters, and installation of a program for establishing a communications link using the adapter. See Appendix A for information on installing the necessary software.

6.1 Voltage Level Test

For the first test all you need to do is initialize the serial port of the microcontroller. To do this you'll have to determine the correct initialization parameters to pass to the `serial_init` routine. Assume you need to communicate at 9600 baud. In the program call the `serial_init` routine to initialize the serial port using the arguments you determined. Initializing the port will ensure that the microcontroller's serial out pin is acting as an output to drive the MAX232 chip.

Use a multimeter or scope to measure the voltage levels at the DB-9 connector on the transmitted data, (pin 3) and received data (pin 2) lines. The transmitted data pin should be showing a negative voltage of around -8 volts. The received data pin should be close to zero volts. If you are not seeing voltages close to these then something is wrong with the circuit or the program. Don't bother going on to the next parts until the above tests come out correct.

6.2 Transmit Test

For the next test write a program that loops sending some character followed by a delay of a few milliseconds.

```
while (1) {
    serial_out('A');
    _delay_ms(5);
}
```

This will create an output stream that can be observed on the oscilloscope. Download this program to the board and then use the oscilloscope to look at the signal in multiple places to see if any data is being transmitted.

- At pin 3 of the microcontroller you should see the bits of the 'A' character being transmitted with the signal levels going between 0 and 5 volts. This same signal should be present on pin 11 of the MAX232 IC.
- On pin 14 of the MAX232 you should see the bits of the 'A' character but the voltage levels should now be going between around -8 and $+8$ volts.
- Lastly, check the signal on pin 3 of the DB-9 connector. It should be the same as what you observed in the previous step on pin 14 of the MAX232.

Besides testing for the presence of the signal and voltage levels, this test can be used to confirm that the data rate is correct. On the scope look for the width of one of the bits being transmitted. If the data rate is correctly set to 9600 bits/second, one bit should be $104\mu s$ wide on the scope. If it is some other width, then the data rate is wrong and this needs to be fixed before proceeding to the following tests.

6.3 Loopback Test

Now write a program that loops echoing the character received from the computer back to the computer. You can use the input and output routines from the serial communications document for this.

From the EE 459 supply cabinet get one of the USB serial adapters. These have a USB connector on one end of a cable and a female DB-9 on the other. Plug the adapter's USB connector into one of the USB ports on the computer. With the DB-9 connector of the USB serial adapter **not** connected to the project board, start the communications program as described above for the computer. Once it is running, anything you type goes out the serial port.

Try typing some characters. With the adapter not connected to the board you should not see the characters echoed to the screen. Now connect the serial adapter cable to the project board and type some characters. If the hardware and software are working correctly you should see them echoed in the terminal window.

6.4 String Output Test

Write code to read a character from the computer and then output one of two strings depending on whether the character was a consonant or a vowel ('a', 'e', 'i', 'o', or 'u'). The strings should contain the character you typed. For example

```
You entered the vowel "a"
That was the consonant "m"
```

You can assume the input character is in lower case. The `sprintf` function can be used to format the strings. The strings will have to include both a carriage return and a line feed character at the end ("`\r\n`") in order for them to be properly displayed in the terminal window.