Lab 8 – RS232
October 22, 2015

In this lab you will use the Serial Communications Interface (SCI) system on the HCS12 microcontroller to send and receive characters using the RS232 signal format. You will also observe the transmission of RS232 characters on the oscilloscope.

1 Prelab Questions (complete prior to coming to lab)

The standard 9-pin RS-232 connector has two rows of pins. The male connector on the PC is shown in Figure 1(left) – this is the view looking at the pins. The female connector on the cable that plugs into the male connector is shown in Figure 1(right).

![DB-9 RS-232 Connectors](image)

Answer the following questions:
1. We will use the RS-232 signals called Tx (Transmitted data), Rx (Received data), and GND (Signal ground). What pins are they on the standard 9-pin serial connector?
2. What are the ASCII codes for the first letters of the last names of the people on your team?
3. Write C statements to initialize the SCI system for 9600 baud (using the SCIBD register) and turn on the transmitter and receiver (using the SCICR2 register).

2 RS232 transmission

Get a serial cable with a 9-pin connector from the instructor. You will need a cable with a connector on one end and bare wires on the other end; so that you can connect the oscilloscope leads to those wires.

Find the wires corresponding to Tx and GND, and attach the oscilloscope leads to these wires.

Plug the end with the connector into your PC’s COM port and run the application called “PuTTY”. Select “Serial” and a speed equal to 9600 bits per second.
1. For each person on your team, take the first letter of their last name. Transmit that letter continuously from the PC. Observe the transmitted signal on the oscilloscope. Capture an oscilloscope image of what you observe. Be sure to label the time values and voltage values. Also label where the start bit, stop bit, and data bits are located. Indicate the logic 1 or 0 values of the data bits and how they relate to the ASCII code for that letter.

2. By looking at the transmitted signal, determine what ASCII character is transmitted when you press the ENTER key on the keyboard. Determine what ASCII character is transmitted when you press the BACKSPACE key on the keyboard. Determine what ASCII characters (there are more than one) are transmitted when you press the UP ARROW key on the keyboard.

**Sign-Off 1:** Demonstrate your program and data waveforms to the instructor. Discuss the ENTER, BACKSPACE and UP ARROW waveforms.

### 3 Heartbeat Indicator

As shown in the lecture slides, implement a program that uses the RTI system to flash an LED at a rate of about 1 per second, using interrupts (i.e., it turns on the LED for one second, then turns off for 1 second, and so forth). This is kind of a “heartbeat” indicator by which you can tell your program is running. Keep the heartbeat indicator in place, while you are adding code to do RS232 transmission and reception in parts 5 and 6.

**Sign-Off 2:** Demonstrate your program to the instructor.
4 “Run” mode

In order for your program to use the serial communication interface (SCI) system on the microcontroller, you have to be in “Run” mode. This is done by changing switch SW1 on the Nanocore module from the “Load” position to the “Run” position.

You will first download your program as usual using the CodeWarrior debugger. Once it is downloaded you will then put switch SW1 into the “Run” position and hit the reset button on the SSMI board. That will cause your program to start running instead of the monitor program, and your program is free to use the serial port.

When in “Load” mode, one of the things that the monitor program does is to change the E-clock frequency from 8 MHz to 24 MHz. So if you are in “Run” mode, you will have to do that yourself at the beginning of your program. Just incorporate a call to the “initializePLL()” routine that follows.

```c
/*******************************************
Call this function to initialize the PLL, if the
RUN/LOAD switch is in RUN position. If you don't do
this, your clock will be at 8 MHz rather than 24 MHz.
It doesn’t even hurt to call it even if you are not going
to use RUN mode; you will just set the clock (again) to 24 MHz.
*******************************************/
void initializePLL(void) {
    CLKSEL_PLLSEL = 0;  // turn off PLL for source clock
    PLLCTL_PLLON = 1;   // turn on PLL

    // Assuming OscFreq = $8000, we mult by synr+1=3 (to get 24MHz)
    SYN_R SYN = 2;       // set PLL multiplier
    REFDV_REFDV = 0;     // set PLL divider (divide by REFDV+1)

    CRGFLG_LOCKIF = 1;  // clear the lock interrupt flag
    while(!CRGFLG_LOCK){} // wait for the PLL to lock before continuing
    CLKSEL_PLLSEL = 1;  // select the PLL for source clock
}
```

5 Sending a character

Write a C program that initializes the SCI system to 9600 baud and turns on the transmitter and receiver. Then your program should continuously transmit a character by repeatedly calling the function “putcharSCI”, which is given below:
void putcharSCI(char c) {
    // wait till transmit data register is empty
    while (!(SCISR1 & 0x80)) ;
    SCIDRL = c;      // send character
}

Download the program onto the SSMI board, and then switch to “Run” mode. Close the CodeWarrior debugger and start up the PuTTY application. Hit the reset button on the SSMI board to cause your program to start running. Verify that you receive the characters on the PuTTY terminal screen.

**Sign-Off 3**: Demonstrate the program to the instructor.

### 6 Send and receive

Write a C function called “getcharSCI” that reads a character from the SCI port and returns it. This function needs to wait until the receive data register is full; then it should return the contents of the SCI data register.

Also write a C function “putstringSCI” that writes a character string to the SCI port. The function takes a string as an input argument (which is just an array of characters). This function should call “putcharSCI”, for each character in the string. Recall that in C, character strings are terminated by the “null” character (i.e., a zero), so you should keep outputting characters from the array until you reach a zero.

Finally, have your main program continuously read a character from the PC, and then print it back to the PC in the form of a string, such as “Character received is xx (ASCII value yy)”, where xx is the received character, and yy is its ASCII numeric code. You will have to use the "sprintf" function from the LCD lab.

**Sign-Off 4**: Demonstrate the program to the instructor.

Include in your report:

1. The C language program, along with a corresponding flowchart or pseudo code.
2. An explanation of the circuit and program.
3. Why is the period of the “heartbeat indicator” unaffected by the processing time needed to handle the RS232 communications?

---

1 If the received characters are wrong, try starting the PuTTY program before running the HCS12 program. If you really understand RS232 communications, you will know why this will fix it (if not, ask the instructor!).
## Lab 8: RS232

Name: ________________________________Name: ________________________________

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>Initials</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sign-Off 1</td>
<td>RS232 Transmission</td>
<td></td>
</tr>
<tr>
<td>Sign-Off 2</td>
<td>Heartbeat Indicator</td>
<td></td>
</tr>
<tr>
<td>Sign-Off 3</td>
<td>Sending a Character</td>
<td></td>
</tr>
<tr>
<td>Sign-Off 4</td>
<td>Send and Receive</td>
<td></td>
</tr>
</tbody>
</table>

### 7 Rubric

<table>
<thead>
<tr>
<th>Deliverables</th>
<th>20 pts</th>
<th>Pre-Lab</th>
<th>5 pts</th>
</tr>
</thead>
<tbody>
<tr>
<td>C code with flowchart/pseudocode</td>
<td>/ 10</td>
<td>Questions</td>
<td>/ 5</td>
</tr>
<tr>
<td>Circuit and program Explanation</td>
<td>/ 10</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Composition</strong></td>
<td>5 pts</td>
<td>Sign-Off 1: RS232 Transmission</td>
<td>/ 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sign-Off 2: Heartbeat Indicator</td>
<td>/ 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sign-Off 3: Sending a Character</td>
<td>/ 7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sign-Off 4: Send and Receive</td>
<td>/ 8</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>/ 50 pts</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>