EE 335 Advanced Microcontroller Engineering
Oregon Tech Portland, Winter 2014
Lab Assignment #3 — Serial Communications Interface
Due January 30

Objective:
The student will demonstrate using the serial communication interface (RS232) using interrupt routines.

Equipment and Software needed:
- Your Dragon12 board. You can use the Simulator for initial development.
- Heartbeat routine from lab assignment 1.

General Instructions:
This assignment is to implement a strip recorder. A stream of data values are sent out the serial port. There is one value per line of output, and the line consists of a sequence of space characters (proportional to the value) followed by the character "X". We will consider values from 0 to 63 being 0 through 63 space characters followed by the "X". Values larger than 63 won't be allowed, nor values less than 0. The recorder is to run at the rate of five lines ("X"es) per second.

DBUG-12 uses serial port 0 for communication and prohibits application programs from setting up an interrupt service routine. In this assignment, use serial port 1 to avoid confusion between DBUG-12 and program communication. You might want to check your program first using the simulator. When you are using both serial ports you will need to connect to port 0 to download your program and give the "g" command to DBUG-12. Then you will need to change jumper J42 (while the program is running) to select serial port 1 for the USB connection.

Remember that an interrupt routine can only process one "chunk" of information at a time. In this case that means one interrupt per character transmitted. We need to implement a state machine within the serial port interrupt routine where the value in *count* represents our state and instructs what to do. The values in count are:
- Values >0 mean decrement count and transmit a space.
- 0 means decrement count and transmit an 'X'.
- -1 means decrement count and transmit a CR.
- -2 means transmit an LF, and turn off the transmitter interrupt enable.

At each interrupt we look at the value and take the appropriate action. Each action ends with transmitting a single character so that another interrupt will occur, except for the last character to transmit, which disables the interrupt.

The data we record needs to come from the analog to digital converter. Use the input that connects to the potentiometer so that varying the potentiometer position will vary the voltage and therefore the chart value. Configure the ATD so that it runs

continuously (scan mode) but don’t enable interrupts for it. Have the program every 200 milliseconds take the 8 existing ADC readings, average them, shift right 4 bits to get the value in the range 0 to 63, store in *count*, and enable the SCI1 transmitter interrupt so the line gets written to the strip recorder.

Don't forget the heartbeat monitor interrupt.

To turn in:
- Commented program listings
- Captured program output.
- Discussion of your results