

Embedded Controller Programming I

Homework #4, Serial Port and MOVX

This homework requires you to become familiar with using the internal serial port on the 8051 processor. As we mentioned in class, the SDK has two serial ports; the external serial port (on SDK serial connector P5) which we have been using to communicate with the SDK via the monitor program, and the 8051's own internal serial port (on connector P1) which we will be using for this homework assignment. DO NOT use the external serial port for this homework assignment – that is only for extra credit after you've got it working with SBUF.

First, a quick reminder about accessing the 8051's internal serial port. This port is accessed via the P1 connector (the DB-9 Female connector labeled P1 for Plug 1) on the SDK. To use this port, you just need to connect an ordinary serial cable (NOT a null modem cable) between this port and your computer. To test the homework, you need to either have two PC's running two copies of the terminal program (one for loading the code like we have been doing all along, and a second for communicating with the internal serial port), or just one PC running the terminal program while you switch serial cables. To do the cable switching option, assemble and load your program, then use the MEM command to place the RAM in external data space (see SDK manual) and start it running by typing the E command while connected to connector P5. After doing this, switch your serial port cable to the DB-9F connector labeled P1 on the SDK so you are now communicating with the internal serial port. Then type in a string terminated by the Enter (carriage return) key. You can try this with the demo programs (serial1, 2, and 3.asm) to make sure both serial ports are working correctly. In addition, you can use these demo programs as examples of how to read and send data using the internal serial port. Your homework program should do the following;

- 1) Read in a string of characters from the internal serial port as they are typed in to your PC's terminal program and store them in a buffer area in the EXTERNAL memory space (i.e. the MOVX memory). The serial port should be setup to MODE 1 and 9600 BAUD.
- 2) The buffer should be able to handle at least 80 characters, however, you can make it as large as you want within the limits of the external memory.
- 3) As each character is typed, the SDK should store the character in the external memory buffer and continue storing characters until it encounters the CR character. (Which get sent when you hit the enter key).
- 4) After receiving the CR character, (the easiest way to check for this is to check for either a CR (ASCII 0Dh), the program should stop loading characters in the buffer, and then,
- 5) Send back the entire contents of the buffer (which would be all the characters entered up to the point of the CR/LF) over the internal serial port.
- 6) Return to 1) above and wait for another string of characters to come in...

To test the program, just run it as described above. As you type characters on the keyboard, you should not see anything on the screen until you hit the enter key, at which time the SDK should send back the entire string which has been stored in external memory.

Keep in mind that you need to check the state of the serial buffer (SBUF) before reading to or writing from it by inspecting the TI and RI bits. You may want to use a string termination character to mark the end of your buffer in external memory to simplify the process of finding the end of the string when sending the characters back. You could also keep track of the number of characters in the buffer by creating a buffer count variable.