

EE 371 LAB 4 F05
ASCII Data and Serial I/O

Giraffiti: Vandalism spray-painted very, very high

Name _____
Partner _____
Meeting Day: _____ Hr: _____
Demo: _____
Code _____

Schedule: Week of September 26: Do the lab
 Week of Oct 3: Last chance to demo the lab and hand in the assembler listing.

1. Pre-Lab: Come to the lab ready to do the following programming assignment.

Specifications: Our best customer at our embedded systems company, *Two Dot Engineering*, is finding that students are having a tough time understanding ASCII codes and she would like a program that does the following:

- a. Prints out the following text: **The ASCII code for**
- b. Allows the user to type a single character on the keyboard.
- c. Prints out the following text: **is \$**
- d. Prints out the ASCII code (in hexadecimal) for the character typed.
- e. Prints a carriage return, line feed and repeats starting at step a.

Here is what the program would look like if the user typed the letter A:

The ASCII code for A is \$41

The ASCII code for

Our crack Two Dot Engineering design team decides to do this job and goes to the Guru across the hall to ask for some help. Being a nice guy, he provides the following helping hands:

Helping Hand #1: The Guru has some source code files that allow serial I/O to occur between the DB9 connector on the CSM-12C32 module and a PC running Hyperterminal. This code includes the following routines that you will find useful:

Routine	Function	Calling Sequence
initsci	Initialize the serial port to 9600, 8, N, 1. Must be called before any use of the serial port.	jsr initsci
putchar	Outputs ASCII code in register B to the terminal	ldab character jsr putchar
getchar	Gets a character from the serial port (waits until a character is ready) and returns it in the B register	jsr getchar B has the character

He agrees to supply these routines and required include files at N:\EE371\lab04_f05\serial_io.asm

Helping Hand #2: The Guru requires you to write the following function in a subroutine and then to use it in the program you write.:

Subroutine *putstr*: Print a null terminated string. A null terminated string is a string of ASCII characters to be printed on the screen. The very last byte = 0 (null) indicating the end of the characters to print.

Subroutine input: D register contains the starting address of the string to be printed.

Subroutine output: Nothing, except the string is printed on the screen.

Of course, as is the case with all Two Dot Engineering software, the *ECE Department Assembly Language Practice* document will be followed.

2. Demonstration: You are to demonstrate the program to the customer's representative in the lab. Make sure you can show it working for a variety of situations. To demonstrate, hook a serial cable from the switch box to the DB9 connector on the CPU module (not the one of the SLK board.) Launch the Hyperterm terminal emulator (click on the red telephone icon HC12.ht on the desktop.)
3. Grading: Hand in your list file - one list per group. Demo 10 points (**-2 points for each demo not working 100% correctly**), program .LST file 10 points [you will be graded on your use of labels, equates, orgs, comments and other assembler language features - only one listing per group need be submitted]. Memo report - 10 points (see #4).
4. Special Submission: When you have completed your *putstr* subroutine, email me a copy of the source code (fcady@ee.montana.edu). Make sure the names of both lab partners are on the file. I will remove all identification and email it back out to a DIFFERENT student. Each of you will get somebody else's code to look at and to write a memo report (10 points). The memo is to describe how the subroutine works based on the comments that are in the file. If the person who gets your code can't describe its operation due to lack of good comments and good assembly language programming practices, that student and you and your lab partner will get a reduced grade on the list file you have submitted. If you do a great job on commenting and use of assembler language features you will get a good grade even if the student who writes about it isn't able to describe how it works. The files will be distributed on October 5 and the report will be due in class on October 12.