CpE213: Project 3
8051 C Programming Based Application

Goals: For your project 3, you will interface a 4×4 KEYPAD to your microcontroller and build a simple calculator. The Hyperterminal (or similar terminal emulator), on any desktop, will be your screen. Figure 1 shows what your keypad looks like. The keys labeled 0-9 will remain keys for numbers 0-9. The remaining keys should be relabeled and represent the operations mentioned in Figure 1.

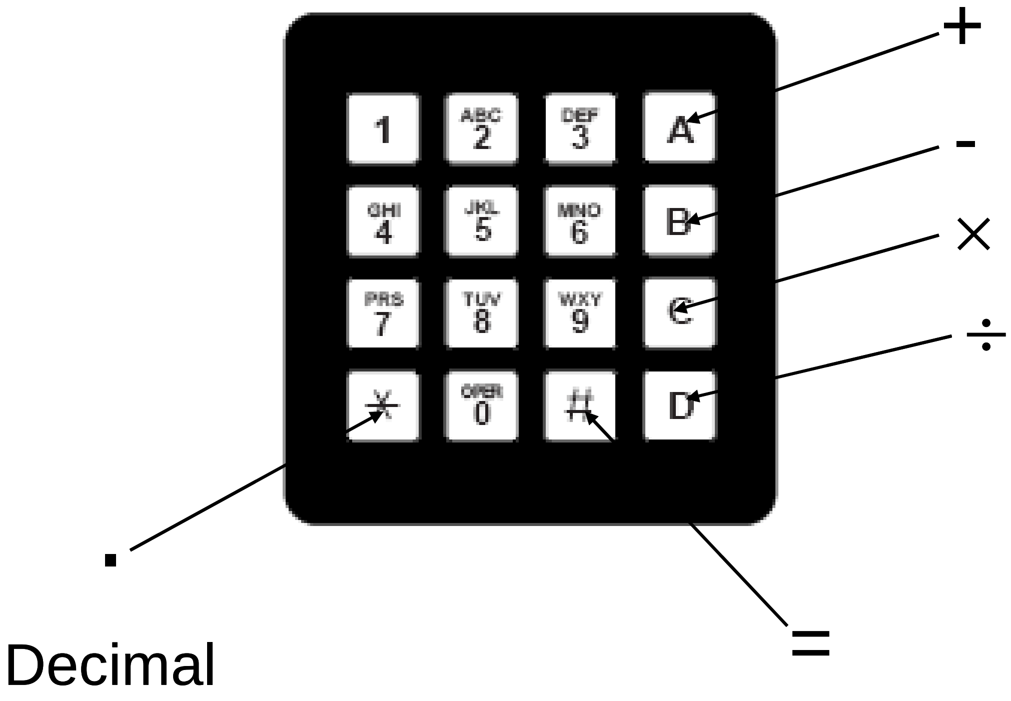


Figure 1: Keypad showing the alternate functions of some of the keys

Alternate Functions:
Key ‘A’: ADD (+) operator
Key ‘B’: SUBTRACT (-) operator
Key ‘C’: MULTIPLY (×) operator
Key ‘D’: DIVISION (÷) operator
Key ‘∗’: DECIMAL (.)
Key ‘#’: EQUAL (=)

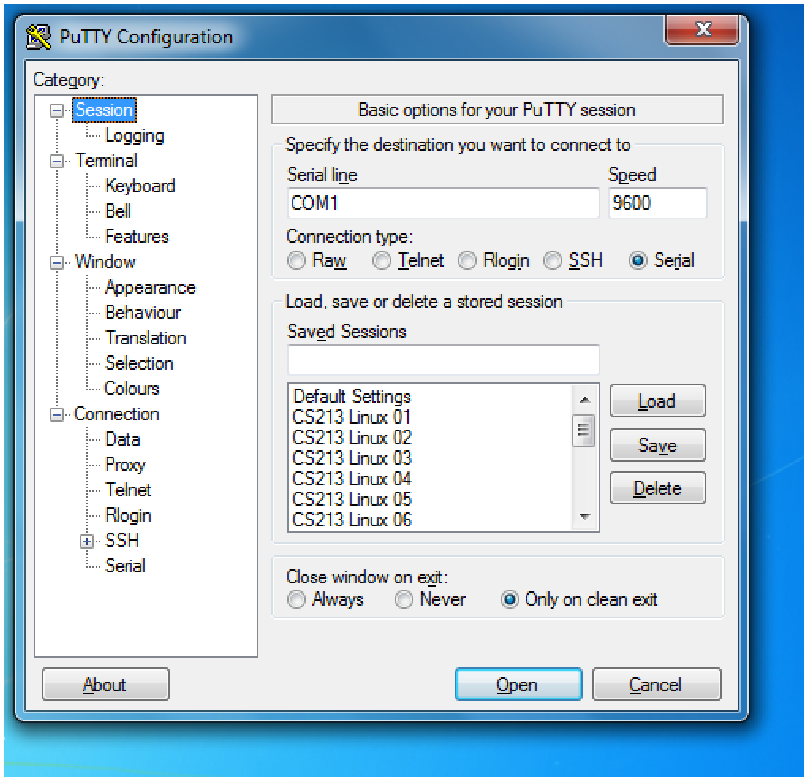
Your program should allow for the following features:

1) Your program should be written in C. If you feel a certain subroutine can be efficiently written in ASM, then you can do so. But the main program has to be in C.
2) The operands and the answers can be of the largest size possible, but is limited to only 2 decimal places. Please note that microcontrollers are not suited to handle floating point numbers very well. After you have the basic program written, you can experiment on what size of number, your program and, finally, your microcontroller can handle. Also, note you have limited available memory.
3) A terminal emulator, such as HyperTerminal (or a similar program), should display all the operands, operators and the final answer.
4) Once the final answer, for a particular operation, is displayed then the program should provide the user with TWO choices.
    a. If the user presses a number, then a new computation begins.
    b. If the user presses an operator (add, subtract, multiplication, or division), then the first operand should be the previous answer.
5) The program should provide error messages for any illegal presses and should restart the operation.
6) If the user wishes to cancel an operation, already started: You can decide how to implement this feature. One way is to interface a switch (from project 2) to a port pin and monitor that pin press for a cancellation command. Be Creative!

Terminal Emulator:

- You should be able to find HyperTerminal on XP machines. Do some research on how to setup HyperTerminal for serial communications
- If you plan to use Windows 7 machines, you can use Putty as an alternative. Do some research on how to obtain and use Putty. Some information is given on the next page.
- You can try other terminal emulators such as TeraTerm
- The serial communication is set to set is 8-bits, no parity, 1 stop bit.

Settings for Putty: Running putty.exe will bring up this main window



PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Serial line: COM1

Speed: 9600

Connection type: ○ Raw ○ Telnet ○ Rlogin ○ SSH ● Serial

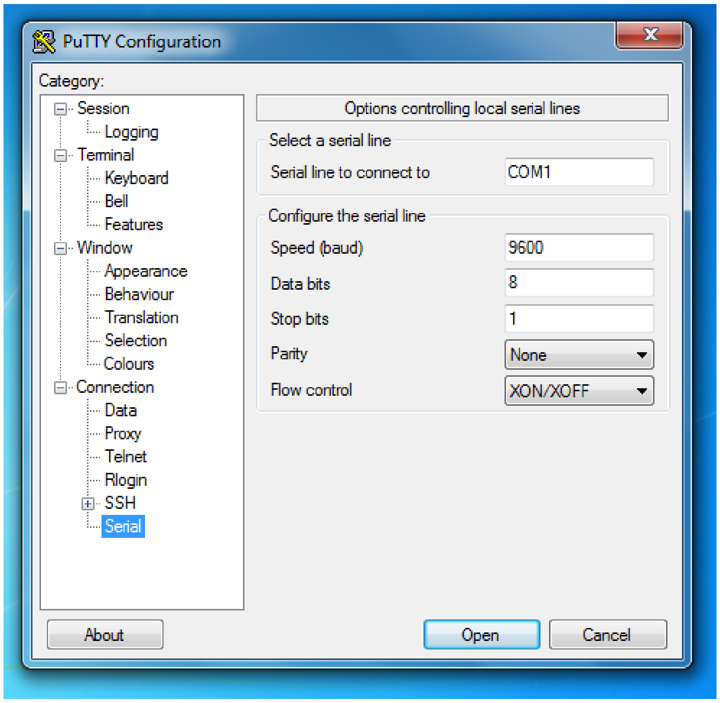
Load, save or delete a stored session

Saved Sessions

- Default Settings
- CS213 Linux 01
- CS213 Linux 02
- CS213 Linux 03
- CS213 Linux 04
- CS213 Linux 05
- CS213 Linux 06

Close window on exit: ○ Always ○ Never ● Only on clean exit

The Serial Line: Can be COM1 or COM2



PuTTY Configuration

Options controlling local serial lines

Select a serial line

Serial line to connect to: COM1

Configure the serial line

Speed (baud)	9600
Data bits	8
Stop bits	1
Parity	None
Flow control	XON/XOFF

Settings for Serial Communications. You may have to vary the baud rate to get your communications working!

Project Advice: In addition to the advice given for project 1, here are additional tips:

- Use the settings for serial communications as outlined in sets 11 and 12 regarding serial communication and interrupts.
- Keypad interface: You will have to follow the algorithm explained in class. The P89LPC932 microcontroller has a separate Keypad interrupt that can be used efficiently. It can be found in chapter 14 of the users manual of the microcontroller. You can also, find code for Keypad key-detection on the web.
- First get your serial communication to work. See if you can send characters to the HyperTerminal. Then develop the Keypad program. Finally perform the mathematical operations.

DELIVERABLES AND DEADLINES:

- **Project demonstration:** Prove to me that your project works. "Cool" extras will be viewed favorably when grading your project. **Note: start debugging your project ahead of time or you seriously risk missing your deadline.**
  - Project demonstrations must be scheduled outside normal class hours. Please work with me to schedule a demonstration time. Saturdays are also possible!
  - Entire project folder: Please e-mail your entire project folder. I should be able to compile and download your program, on the Simon2 board and run it.
- **Project code:** Due with project. During your project demonstration I will ask you to: show me your code, compile your code, email me a copy of your code, download the code to your device, and show me your working device. Your code does not have to be commented at this point. Having comments in your ASM code will help me understand your code better.
- **Short report**. Submit a well-written, but brief, summary (in MS Word or Adobe PDF format) of what you did in your project, what problems you encountered, and how you got around them. Email your report to rdua@mst.edu at the time of project demonstration. Your report should include:
  - **Title and team members**
  - **Summary.** Give a quick summary of what you did in your project, the problems you encountered, and how you got around them. All mathematical calculations performed to get the desired settings.

- **Explanation.** If your code didn’t work, explain what you might do to fix it.
- **Future work.** Explain what you might do to improve your project or the way you went about completing your project (timeline, etc.).
- **Project code.** Well documented code is essential. Code that makes appropriate use of segments, variables, labels, and the like, is far more valuable.
- **Work effort distribution.** List each person in your group. Explain what their job was and the total percentage effort they contributed to the completion of the project.

Projects must be completed, demonstrated, and submitted by

Grading Policy:

- Project Demonstration: 50%
- Project Creativity: 30%
- Project Report: 20%

Note: For the project submission to be valid the project must be demonstrated and a report must be submitted. Failure to do so will result in a reduced overall score for project 3.