CSCI 4415 Lab # 4

Keyboard

Created: January 30, 2011

Objective:

Learn how to capture serial data and make sense out of the bits. For this lab you will connect a PS/2 keyboard to your Z16 contest kit and write a program that will decode the data stream and display the proper characters and scan codes.

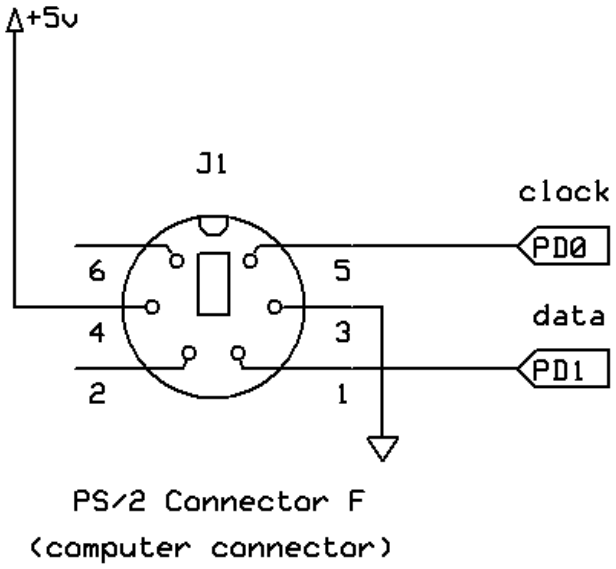
Time Estimate:

10 hours.
You should be prepared to start this after class 6 (Serial Interfaces) and this will be due by Midnight class 10.

Specific Requirements:

For this lab you will connect a PS/2 keyboard to your Z16 contest kit and write a program that will decode the data stream and display the proper characters and scan codes.

1. Connect your Z16 contest kit console port to your computer's serial port (you may need a USB-serial dongle for this). A serial cable is included in the lab kit.
2. I have keyboards if you need one, and in your lab kit is a connector for the keyboard so you can wire that to your Z16 contest kit. Connect the PS/2 connect to the Z16 using the following schematic:



3. Look up the proper pinouts for the keyboard if you need to. Note that there are many different diagrams floating around in the iterwebs for the PS/2 pinout. They are not all the same. Many of them use different (incorrect)pin numbering schemes and some of the schematics you find will refer to the connector on the back of the PC (the female connector) and some refer to the keyboard connector (the male connector) and they don't state which. If you don't know which you are looking at then your pin ordering may be reversed.
4. Modify your CLI from the previous lab to include the following new commands (and appropriate actions).

```
scan
      - will display scancodes from the PS/2 keyboard as the
        key are pressed/released. Will continue to do this
        until any of the buttons (SW1,2,3) are pressed.
type
      - will display characters from the PS/2 keyboard as the
        key are pressed/released. Will continue to do this
        until the escape key pressed, or until any of the
        buttons (SW1,2,3) are pressed (both must work).
```

5. For both CLI modes above, make the speaker play a tone when you start capturing keyboard data and when you end.
6. When displaying scan codes (in the CLI), you will display the numeric codes in hex.
7. When displaying normal typing you should display the name of the key pressed and you should interpret the shift/control keys (so that pressing shift + a results in a capital A, and control-a results in "ctl-a" or "control-a" being displayed.
8. Create a file named readme.txt file that includes any notes about the lab and answers to the questions below.
9. In the main.c file include the following information in a comment block

```
Author: [Your name]
Email:  [Your email]
Class:  [CSCI-4415]
Date:
Lab:    [Lab number]
Description: [short description of program. Include changes you made and
note any enhancements that you made to the lab]
Other files: [list other files that are necessary for this program]
Compile: [provide any special instructions for compiling. Only necessary if
there are special instructions]
Problems: [explaining what you did, what problems you had, how you solved them,
and what you might do differently if you had to do it again]
Comments: [Feel free to provide comments on how this lab went, what you think
is good or bad about it and how it could be improved or anything else you want
to say.]
Enhancements: [describe the enhancements you added to this lab]
```

10. Turn in a zip file (lastname-lab4.zip) of the entire project directory (include the entire ZDSII project directory and the readme.txt). Use BlackBoard to submit assignments (or if thats not working for some reason email the ZIP file).

References

1. This page: http://www.mp3car.com/vbulletin/attachments/hardware-development/27594d1... seems to be the proper numbering (and note that this is the keyboard port so it refers the connector on the computer not the cable on the keyboard, so its the female connector).
2. Use this page for information on how to decode the serial data stream:
    - http://www.beyondlogic.org/keyboard/keybrd.htm
3. Use one of these pages for determining the scancode:
    - http://www.computer-engineering.org/ps2keyboard/
    - http://www.microsoft.com/whdc/device/input/Scancode.mspx

Grading

- On time, in proper format (Zip file), named <lastname>-lab1.zip
- Compiles and runs without problems
- Meets lab requirements
- Reasonable code (partitioning of function, coding standard, readable
- Reasonable comments
- Capture of PS2 signals
- Interpretation of scan codes, tracking of states
- Modular code (no side effects)
- Error detection and handling
- Proper use of Z16
- Answered questions

Enhancements:

- Add command to CLI

```
input
      - will get and display characters from the PS/2 keyboard
        AND interpret them as if they came from the console
        serial port. Will input one command (up to the newline)
        and then return to normal CLI input from the serial
        port.
```

- Send LED status codes back to keyboard to turn on /off keyboa lights for cap-lock, num-lock and scroll-lock.
- Display scan codes on LED display as you type.
- Something you thought of

Questions

Include with your lab submission (in the comments of your main.c file):

1. What technique did you use for watching the data and clock lines (polling or interrupts)? Why did you choose that?
2. Did you use the oscilloscopes in the class room to help debug your lab setup?
3. What difficulties did you have keeping track of when to sample the data line?

4. What did you try that didn't work?
5. If you have problems integrating the LED display library you can exclude that IF YOU NEED TO. Describe the problems you had and what you tried to do to fix things. Make sure you include stubs for the LED library so the CLI will continue to work (just not display scrolling messages).