EECE 344 Digital Systems Design

Lab 2: GPIO, Part 2

1 Pre-Lab

1. Review the lecture notes about writing device driver code for the LCD and the keypad.
2. Review the lecture about GPIO and the interfacing circuits for the LCD and the keypad.

2 Objective

The lab is intended to:

1. Learn the typical code structure for device driver in embedded C.
2. Learn and apply the logic operation in C to **extract bit fields**.
3. Build the interfacing circuit for the GPIO port with an LCD display.
4. Learn and apply **flow controls in C** to scan external input sources (multiple keys).

3 Required Parts

Required parts: the Tiva Launchpad for TM4C123GH6PM, an LCD display, a keypad, jumpers, ADALM 2000 (as the +5V power source), and a breadboard.

You are strongly encouraged to have your LCD **soldered** with header pins **before your lab. Work in pairs** for this lab assignment.

4 System Requirements

This section explains the system's behavior requirements.

4.1 Interfacing the LCD in the 4-bit Mode.

In this part, we will interface an LCD to be the output. The LCD we studied in the class supports two operational modes: the 8-bit mode and the 4-bit mode. In this lab, we will interface the LCD in the 4-bit mode. You may use Port B for the data bus interface, as explained in the lecture. You may proceed as follows.

1. Use the template provided on Canvas to start the design of your code. Start with the low-level "device drivers", the functions that configure the LCD hardware and provide APIs to your main function. The device drivers include the following:
   (1) void LCD_4Bits_Init(void); // initialize the GPIO and the LCD.
   (2) void LCD_Write4Bits(unsigned char data, unsigned char control); // write information to the LCD using the 4-bit mode.
   (3) void LCD4Bits_Cmd(unsigned char command); // send a command to the LCD by calling the "LCD_Write4Bits" function.
   (4) void LCD4Bits_Data(unsigned char data); // // send a data (character, number, symbol) to the LCD by calling the "LCD_Write4Bits" function.
   (5) void delay_micro(int n); // pseudo delay for micro seconds, provided to you.

(6) void delay_milli(int n); // pseudo delay for milli seconds, provided to you.

2. Design and implement your main() function to display characters or numbers or symbols on the LCD screen.
3. Compile and debug your code to make it error-free. You may use the logic analyzer or ADALM2000 to help your debugging. Make sure your code has the correct logic behaviors.
4. Build your circuit, connecting the three types of pins of LCD: power, data, control.
5. Test the performance of your system.

Please keep the circuit and code for your LCD. We will use it and add a keypad to your system. We might also use the LCD in a future lab toward the end of the semester.

4.2 Interfacing with the LCD and the Keypad.

Now let us add a 16-key keypad to your circuit. You may interface it to the Lauchpad on Port C and Port E, as explained in the lecture. Follow the following steps:

1. Use the template from Canvas to finish the device drivers for the keypad:
   (1) void KEYPAD_Init(void); // initialize GPIO C and E that interface to your keypad.
   (2) unsigned char getKey(void); // design this function to obtain user key stroke.
2. Modify your main() function so that it allows user to type keys on the keypad and display the typed number, letter, or symbol on the LCD screen. Note: since in Section 4.1, you might have only enabled the clock to Ports A and B, you need to use friendly code to add clocks to Ports C and E.
3. Compile and debug your code to make it error-free. You may use the logic analyzer or ADALM2000 to help your debugging. Make sure your code has the correct logic behaviors.
4. Construct your circuit to connect the keypad to the Launchpad.
5. Test the performance of your system.

5 Grading

The grading of your lab comprises the following components. Refer to the rubrics about the ranking of each component.

5.1 Performance of the Lab

Show the performance of your built system. Explain the interfacing circuit to your instructor.

5.2 Demonstration

When demonstrating the program, each student in the team will be asked a different question to demonstrate his/her understanding of the project at hand. You are expected to explain the design and implementation of your code if asked.

5.3 Deliverables

Please indicate who your lab partner is in your deliverables. Submit your deliverables on Canvas to “Lab 2: GPIO, Part II” under “Assignments” by the due time and date. Refer to the schedule shown on Canvas. For this lab, your deliverables are:

1. Provide a flowchart of the program you designed. Use software to create the chart, as any hand-drawn chart will be classified as at best second ranking or lower according to the rubrics.

2. Your interfacing circuit drawn by a professional software, such as PCB Artist (available at https://www.4pcb.com/free-pcb-design-software.html). Any hand-drawn chart will be classified as at best second ranking or lower according to the rubrics.

5.4 Code

Submit your code on Canvas to “Lab 2: GPIO, Part II” under “Assignments” by the due time and date. Your project folder should contain all C code, compressed into a zip file. Include comments and indentation in your code. Points will be deducted for sloppy code lacking documentation.

5.5 Lab Report

You may choose to write a lab report for this lab. Recall that you are required to submit three lab reports this semester, and you have done the mandatory first report for Lab 1. If you choose to write a report for Lab 2, submit your report on Canvas to “Second Lab Report” under “Assignments” by the due time and date. Follow the instructions in the section on "Lab Report Writing Guidelines" in “EECE 344 Digital Systems Design Lab Policy.” In the body of your report, make sure you include the following:

1. A flowchart of the program you designed. You may use the flowchart you prepared for Deliverables (5.3 above). Use software to create the chart, as any hand-drawn chart will be classified as at best second ranking or lower according to the rubrics.
2. Your interfacing circuit drawn by professional software. You may use the flowchart you prepared for Deliverables (5.3 above). Any hand-drawn chart will be classified as at best second ranking or lower according to the rubrics.
3. A photo of the circuit on the breadboard for the system.
4. Technical writing explaining your design and the flowcharts of your code. Describe any challenges in your experiment and how you solved the problems. Draw a conclusion about programming the digital I/O with the Launchpad.