EECE 344 Digital Systems Design

Lab 1: GPIO, Part 1

1 Pre-Lab

1. Read Sections 2.7, 2.8, 4.1, and 4.2 of the textbook.
2. Review lecture contents about GPIO and the interfacing circuits.

2 Objective

The lab is intended to:

1. Familiarize the students with the software development steps for the microcontroller.
2. Learn the typical code structure in embedded C.
3. Learn and apply the array data structure in C.
4. Build the interfacing circuit for the GPIO port with a push button, an LED, and 7-segment LED displays.
5. Learn and apply the concept of negative logic for a switch/push button.
6. Learn and apply logic operations in C to control a switch/push button.
7. Learn and apply transistors as selectors for DC peripherals.
8. Write a program to achieve desired behaviors for the system.

3 Required Parts

The Tiva Launchpad for TM4C123, one push-button switch, several resistors within the range of 220-1K Ohms, an LED, two 7-segment displays, jumpers, two transistors, and a breadboard. You are going to build a system using the above components. The behaviors of the system are explained next.

4 System Requirements

This section explains the system's behavior requirements.

4.1 Switch Input and LED output.

Pins 2 through 6 on Port A are available for configuration. You may review the data sheet to pick an available pin on Port A to connect the push button. Interface the push button in negative logic to GPIO Port A.

Interface the single LED to another pin on Port A so that the microcontroller source current to the LED. Remember to calculate and choose a resistor for your LED circuit. This circuit should give you an idea about interfacing the 7-seg display described in Section 4.2 of this handout, which is essentially a group of LEDs.

The behavior requirement of this part is this: when the user presses the push button, the LED turns on; when the user releases the push button, the LED turns off.

4.2 Interfacing with the 7-seg LED Displays.

The 7-segment LED display is a simple yet useful display peripheral for digital systems. Many elevators still use 7-segment displays to show floor information. The following figure is an example of using four 7-seg displays on an older model of Instant Pot.

The TM4C123GH6PM on the Tiva Launchpad has a limited number of pins for interfacing peripherals. If we were to interface each display using completely different pins, we would have used 32 pins for 4 LED displays. It becomes a scalability issue. Instead, we will allow all displays to use the same pins, but we will need "selectors" to decide at which time instance which peripheral (the display) uses the GPIO pins.

In this part, let us practice with two displays using the same port. Interface two 7-seg displays to Port B, so that one display shows the ten's place and the other shows the one's place. Review the lecture about interfacing 7-seg displays, especially the common cathode mode.

Since we have 7 segments and a dot on the display, we need to connect them to Port B which has 8 pins. We also need to use two pins on another port, such as Port A, to select the display. That is, each display is controlled by a selector pin. The interfacing for the selection is through transistors. Review the lecture to apply transistors as switches.

The pseudo delay function is provided to you in the project template. In your code, define an array variable `NumberPattern` to hold the values to be displayed on the7-seg LED. Choose two distinct numbers from `NumberPattern` to display on your 7-seg displays. Follow the following steps:

1. Configure Port B as the output port to drive the 7-seg displays.
2. Configure two pins on Port A to select the displays.
3. Write the pattern of the number you chose from `NumberPattern` for the **ten's place** to Port B.
4. Write one bit of Port A to activate the **ten's** display.
5. Delay for some time (shorter than 8 milliseconds) using the pseudo delay function.
6. Write the pattern of the number you chose from `NumberPattern` for the **one's place** to Port B.
7. Write one bit of Port A to activate the **one's** display.
8. Delay for some time (shorter than 8 milliseconds) using the pseudo delay function.
9. Repeat Steps 3 through 8.

Utilize the debugger and its logic analyzer or ADALM2000 to debug your program.

4 Grading

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

4.1 Performance of the Lab

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

4.2 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.

4.3 Deliverables

Submit your deliverables on Canvas to “Lab 1: GPIO, Part I” 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.

4.4 Code

Submit your code on Canvas to “Lab 1: GPIO, Part I” under “Assignments” by the due time and date. Your project folder containing all C code, compressed into a zip file. Include comments and indentation in your code. Points will be deducted for sloppy code lacking documentation.

4.5 First Lab Report

A lab report is required for this lab. Submit your deliverables on Canvas to “First 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 (4.3 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. A snapshot of your simulation results in displaying the code on the logic analyzer.
3. Your interfacing circuit drawn by professional software. You may use the flowchart you prepared for Deliverables (4.3 above). Any hand-drawn chart will be classified as at best second ranking or lower according to the rubrics.
4. A photo of the circuit on the breadboard for the system.