**Computing Machinery II**

**Assignment 3**

**Basic GPIO and Interrupts**

Create a program that uses the Raspberry Pi GPIO pins to drive 3 LEDs and respond to 2 pushbutton switches that are located on the supplied breadboard. Use the following GPIO pins for the specified devices:

|  |  |  |
| --- | --- | --- |
| **GPIO pin** | **Input or Output** | **Device** |
| 17 | output | LED 1 |
| 27 | output | LED 2 |
| 22 | output | LED 3 |
| 23 | input | Switch A (pulled high) |
| 24 | input | Switch B (pulled low) |

The program will have 2 states:

* State 1: LEDs are illuminated one after the other in the pattern: 1, 2, 3, 1, 2, 3, etc. Each will be on for about 0.5 second.
* State 2: LEDs will be turned on one after the other in the pattern: 3, 2, 1, 3, 2, 1, etc. Each will be lit for about 0.25 second. In other words, the pattern will be twice as fast and in reverse order compared to State 1.

Your program will start in State 1. Configure interrupts so that Switch A triggers a falling-edge interrupt, and Switch B triggers a rising-edge interrupt. When Switch A is pushed, an interrupt service routine (ISR) should put your program into State 2. When Switch B is pushed, the ISR should put your program into State 1.

You may write your program in C or in A64 assembly language, or a combination of both. You can freely use the following files located on D2L as part of your code base: gpio.h, irq.h, link.ld, Makefile, startV2.s, uart.c, and uart.h. You should use “busy loops” to create any delays needed in your program.

**Demonstration**

You will demo your program to your TA in one of the tutorials shortly after the due date. Your TA will set up a schedule for this in class before the due date.

**Bonus (10% if fully implemented)**

Use the System Timer peripheral or the Timer (ARM side), plus interrupts to create any delays needed in your program. In other words, replace your “busy loops” with code that uses one of the hardware timers to delay the appropriate amount of time.

**New Skills Needed for this Assignment:**

* Ability to do cross compilation for the Pi
* Ability to program GPIO pins for input and output
* Ability to configure interrupts
* Ability to create an interrupt service routine

**Submit the following on D2L:**

1. All files needed to compile your program. This will include the files listed above, plus any custom code that you have created. Your TA will compile and run your program to make sure it is working. If you are doing the bonus part of the assignment, then also include a README file that indicates this.

**Computing Machinery II**

**Assignment 3 Grading**

**Student:**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Configure GPIO output pins 6 \_\_\_\_\_\_

Configure GPIO input pins 4 \_\_\_\_\_\_

Configure interrupts 4 \_\_\_\_\_\_

Interrupt Service Routine(s) 4 \_\_\_\_\_\_

Correct behavior for State 1 3 \_\_\_\_\_\_

Correct behavior for State 2 3 \_\_\_\_\_\_

Correct behavior for pushbuttons 2 \_\_\_\_\_\_

Code formatting and documentation 2 \_\_\_\_\_\_

Design quality 2 \_\_\_\_\_\_

**Total 30 \_\_\_\_\_\_ \_\_\_\_\_%**

**Bonus (10% if fully implemented) \_\_\_\_\_%**

**Assignment Grade \_\_\_\_\_%**