

Programming The PIC Micro Controller

Computer Organization
CS 140

The Lab - Overview

In Lab01, you learned how to program the PIC Microcontroller running on the Development Board. You could write a bit of new code, and you could explain what various Assembly instructions do. You also had a general idea of how to read the hardware specification.

In this lab, you will have one very specific task.

Write a program with the following characteristics:

1. Get a random number
2. Get a second random number
3. Perform and display the logical AND of the two numbers
4. Perform and display the logical OR of the two numbers
5. Perform and display the logical XOR of the two numbers

Functional Specification of This Program

A “Functional Specification” is a description of the behavior of a program as seen from an outside point of view. This view know nothing about the details – it understands the program only from its function or external behavior. The “Design” is written from a programmer’s point of view; it understands the details of HOW the program goes about performing the functions.

1. The program flashes all 4 LEDs when the program starts. While this is happening,
2. The program “chooses” a random number based on when the switch is depressed.
3. This number is displayed on the LEDs.
4. A second random number is chosen.
5. This second number replaces the first that has been displayed on the LEDs.
6. Time = 5 seconds elapses while the second number is displayed.
7. The logical AND of the two random numbers is displayed for 5 seconds.
8. The logical OR of the two random numbers is displayed for 5 seconds.
9. The logical XOR of the two random numbers is displayed for 5 seconds.
10. The program continues forever and now goes to the “all flashing” mode in the first step.

Designing Your First Piece Of Code

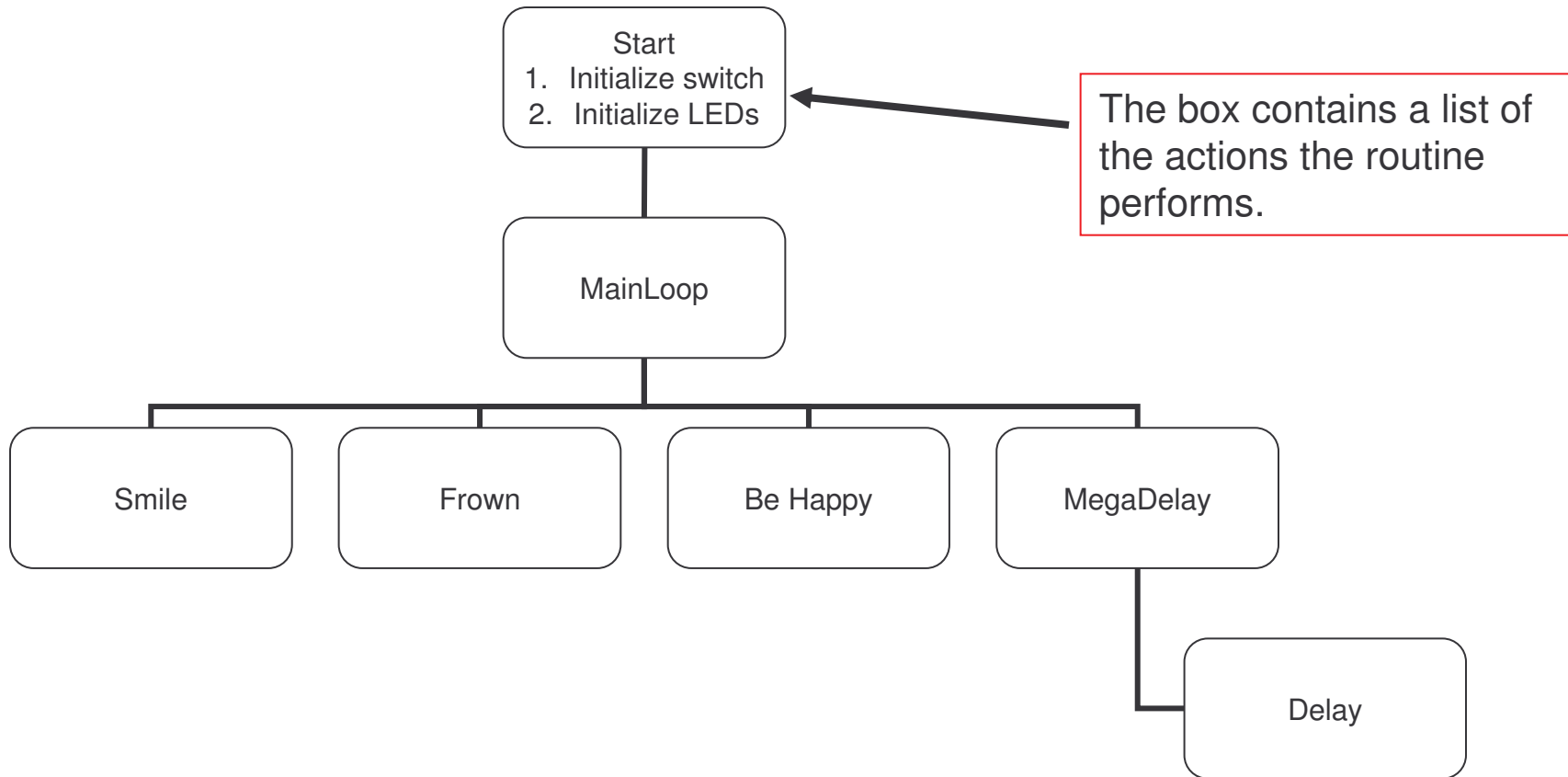
This program is pretty simple really. You could write it in java in about 5 minutes. But it will take MUCH MORE code in PIC Assembler, and will have lots of new learning involved. It's frequently difficult to do numerous tasks at the same time and that is the case here; you're coping with a new language, a new piece of hardware and a new IDE.

Your design can be on paper or electronic. It should include boxes that represent methods/subroutines that will be performed by your code. Each box should have the name of the routine, as well as the tasks to be performed by that routine.

I've started a design on the next slide to show what I mean.

I'm REALLY emphasizing using separate methods/subroutines for various actions as a way of organizing your thinking.

Writing Your First Piece Of Code



This says:

Start calls MainLoop:

MainLoop in turn calls these various routines – perhaps not every time around the loop, but calls to all these other routines might occur at some time.

Programming Hints

Iterative Development:

What most of you do is write some code and then see if it works. Let's formalize this as a good thing to do. But I'd like you to follow this cycle:

1. Write or modify the design
2. Code a little piece of the design (I'd recommend for this project you work first on the Start code and then on the Delay Code. Never chew on more than 30 lines at a time.
3. Assemble the code.
4. Test the code using the simulator, MPSIM – it should work before going on.
5. GoTo Step 1 – yes, change the design if necessary.

Coding In Chunks:

The use of methods/subroutines is a true wonder in Assembler. Use calls and returns liberally.

Too much assembler code is a nest of intertwined goto's with no way of ever figuring out a problem when disaster strikes.

Programming Hints

Random Number Generation:

The easiest way to do this is have the program running in a loop. Every time it goes thru the loop it increments a counter. It also checks if the switch has been pushed. When it sees a push, then the counter at that time is the random number to use. This actually is quite random since it depends on a human (who knows nothing about how fast the counter is going) pushing a button at an arbitrary moment.

Timers:

The spec asks that you keep values in the LEDs for 5 seconds. You will need to develop a timer to do this. I'd recommend that you use the timer already developed in Lab01, and call it repeatedly many times in order to achieve the total time desired.

Code Formatting and Commenting:

Formatting is mostly about indentation. I don't care what you do as long as you're consistent. Also be consistent about using upper or lower case for mnemonics. Commenting should give the "big picture" of what's happening. A comment like "moving from accumulator to file" is NOT very useful – that's what the instruction already tells us. But if you say "Moving accumulator to PORTC to display on lights", that tells *why* the instruction is there.

Show and Tell Evaluation Sheet

Lab02:

Your Name: _____

Design

- There IS a design written out electronically or on paper – show me.

Examination of the code.

- Code is formatted correctly.
- Code has comments and is legible.

Code execution

- Program is assembled from source and can be loaded onto the Board.
- Five Second Delay works
- Random Numbers generated correctly.
- Logic performed correctly.
- Program behaves according to the specification given in this write-up.