

EET363 Introduction to Microcontrollers

OIT Portland West, Fall 2010

Lab Assignment #3 – General Purpose I/O
Due November 11

Objective: The student will use general purpose I/O pins.

Project Description: Lots of people have talked about how Star Trek technology has in some cases been implemented (talking to computers, communicators/cell phones, even Transparent Aluminum has been made). In the remaining lab assignments we will implement Star Trek technology on the Dragon12-Plus board!



In the episode “The Menagerie” the former captain of the Enterprise, Captain Christopher Pike, has been horribly disfigured from radiation burns saving other Starfleet members lives, and now moves around in a motorized chair and life support system. Unable to speak, he communicates by flashing a lamp mounted on his vehicle. One flash for yes, two for no.

Your assignment is to implement that technology of the future, today. Whenever button SW5 (connected to PH0) is pressed, the LED at the top of the board (described below) is to turn on for one-half second, then turn off. This is the “yes” signal. Whenever button SW4 (connected to PH1) is pressed, the LED connected to PBO is to turn on for one-half second, turn off for one-half second, turn on a second time for one-half second, then turn off. This is the two-flash “no” signal.

Equipment and Software needed:

- Everything you used in Lab Assignment 2
- While you can develop your program using the simulator, use the Dragon12-Plus for final testing.
- You may use the C compiler, if you wish (and know C).

General Instructions:

The Dragon12+USB board has an RGB LED that is perfect for this assignment. Captain Pike’s indicator light glows yellow, which you can obtain by illuminating the Red and Green LEDs in the RGB LED simultaneously. The Red, Green, and Blue LEDs are connected to Port P pins 4, 5, and 6 (PP4, PP5, PP6), but as Wytec gets these

parts with different pinouts you will have to experiment to find the right pins to drive high. Put a tissue in front of the LED to diffuse the light to make it look better.

If you have an older Dragon12+ board you will need to use one of the LEDs connected to port B. In order to enable the LEDs you need to configure PJ1 as an output and drive it to logic 0.

Make sure the all the sliders in SW1 (the DIP switch) are in the UP (which is OFF) position so they don't interfere with the push buttons.

You will need a way to create a half-second delay. I suggest a program loop that generates a delay. A loop that takes 10 microseconds to complete can be placed inside another loop that iterates 50,000 times to create a half-second delay. The delay code can be made into a subroutine that can be called from the several places the delay is needed.

Writing in assembler, you should be able to calculate the number of iterations necessary for the inner loop based on the time it takes to execute each instruction. If you are using C you will have to experiment with the delay. In C, make the variable global and volatile to prevent the loop from being optimized away by the compiler.

Make sure your program will allow answering multiple questions, not just one. But at this point assume the button will not be held down, and only test its state while not flashing the LED.

