EE4524 Introductory Lab 2 (Runs for 2 weeks)

(Ver 1.2, 22 Jan 2020)

The aim of this project is to gain familiarity with programming a microcontroller and its ports and to begin using interrupts.

This lab is divided into three successively more challenging tasks.

For each task, you must program the ATMega328P microcontroller in an Arduino Board and demonstrate a working program running with the switches and LEDs shield.

Task 1

You will find 1 C file in SULIS in the folder Introductory Lab 2: Lab2Task1.c. This implements a slightly modified version of the Cylon program we used in the introductory lab session. It uses a different delay loop to delay a fixed time and it is necessary to set the F_CPU parameter to reflect the microcontroller clock frequency for this delay to provide an accurate time.

The program flashes 8 LEDs attached to PORTD, where a HIGH or 1 turns on an LED and a LOW or 0 turns it off. The LEDS turn ON and OFF in Cylon pattern.

Modify the program while (1) loop as follows:

If PINB **bit 4** = 1, Set a variable, called polarity to 1.

If PINB **bit 4** = 0, Set a variable, called polarity to 0. (This is the ELSE case).

If PINB **bit 5** = 1, set a variable called delay_type to 1.

If PINB **bit 5** = 0, set a variable called delay_type to 0. (This is the ELSE case).

On the Arduino add-on board, the Push Buttons are connected to PINB bits 5 & 4.

Modify the move_leds() function so that it takes two parameters, polarity and delay_type.

In the move_leds function test the passed-in parameters as follows:

If polarity == 1, the LED output shifts ONE LED ON at a time (ie shift turns one LED ON, others OFF)

If polarity == 0, the LED output shifts ONE LED OFF at a time (ie shift turns one LED OFF, others ON)

If delay_type == 1, LEDs ON/OFF time is set at 0.25 seconds (ie Delay time is set to 250000 microseconds or 0.25s)

If delay_type == 0, LEDs ON/OFF time is set at 1 seconds (ie Delay time is set to 1000000 microseconds or 1.0s)

Your main program should consist of the following components:

An initialisation section, that sets all the bits of PORTD to outputs and all the bits of PORTB to inputs.

C Code to read PINB and set two variables based on the settings of PINB bits 5 and 4: called *polarity* and *delay_type*. You will have to read PINB in the while loop.

Note that *polarity* and *delay_type* should be declared in main.

A call in main() to a function move_leds(polarity, delay_type). You will need to modify move_leds so that it takes two parameters. It's up to you to decide what type of variables you are going to pass. (Hint: either int or unsigned char types will work).

Note: PORTB is an input port here, so use the PINB register to get its value (not the PORTB register).

You may notice that changes on PINB do not change the pattern immediately – PINB is only sampled after the current pattern has completed its sequence.

Question 1: How do you think that this behaviour could be changed so that PINB would have an immediate effect?

Question 2: As part of the initialisation, after making PORTB into inputs, the following line has been included:

PORTB = 0b00110000;

Since PORTB bits are all inputs, what is the purpose of this line? (You will need to consult the ATMega328P data sheet to answer this question).

Task 2

Modify your program so that polarity and delay_type are **declared as integers**, but passed to the move_leds as **pointers**. You will also have to modify move_leds() and its prototypes again as a result.

Task 3

Modify your program so that changes on PINB affect the LED flashing immediately, but use interrupts to detect and read the changes on PINB. You will need to use the External Interrupts – Pin Change Interrupts to do this. Again you will need to modify move_leds() among other things. We will cover these points in lectures or tutorials over the next few weeks.

Submit the final version of your programs using Sulis.

Submit a separate set of files for each task.