Department of Engineering

EE 3710 Lab 3

Title: Tug-o-war

Objective: The student should be able to write, assemble, download and test a simple microcontroller program that manipulates data on a bit level.

Parts: 1-C8051FX20-TB Evaluation Board & USB Debug Adapter
Project Board from Lab 2

Software: Silicon Laboratories IDE version 3.50.00 or greater

Preparation: Write the title and a short description of this lab in your lab book. Make sure the page is numbered and make an entry in the table of contents for this lab.

Write an 8051 assembly program (tugowar.asm) that meets the following requirements:

1. When the system is reset (either by powering it on or by pressing the reset button) all LEDs in the bar graph shall be off except for the two in the center, which shall be lit.

2. Each time the left button is pressed (transitions from an open to a closed state), the 2 lit LEDs shall move one space to the left unless the two lit LEDs are at either extreme of the bar graph.

3. Each time the right button is pressed, the 2 lit LEDs shall move one space to the right unless the two lit LEDs are at either extreme of the bar graph.

Create a project file for this lab, Add tugowar.asm to your project and build it. Make sure there are no assembly errors.

Note. Mechanical switches (such as pushbuttons) almost always “bounce”. That is to say that the switch may open and close dozens of times while it transitions from an open to closed state and vice versa. Obviously you won’t want your program to react to each switch bounce. Fortunately, most switches settle out after 20ms or less, a period that is imperceptible to humans. The upshot is that your program can “debounce” switches simply by sampling them every 20 milliseconds or so.

Also note. The C8051F020 has an internal oscillator frequency of about 2MHz (500ns period clock cycle) and DJNZ takes 3 cycles (1.5us). Most instructions on the C8051F020 require the same number of clock cycles as they have bytes. Keep this in mind as you program your 20ms delay.

Create a project using the IDE and build your code as you did in Lab 2. Correct any syntax errors until builds without error.

At the beginning of each lab, you are required to show your preparation work to the lab instructor who will score it based on the rubric given in Lab 1.

Lab Work: Using the IDE, download your program code and run it. Verify that it seems to meet the requirements on the previous page. Debug it if necessary.

Stop the program and reset it. Set a breakpoint right before or after the 20ms delay. This allows you to test the program in “slow motion.” Each time you hit the breakpoint, press or release a pushbutton and “continue” the program. Verify that the lit LEDs (a) move left when the left button is pressed, (b) move to the right if the right button is pressed and (c) remain unchanged if both buttons are pressed.

Remove the breakpoint, rerun your program and challenge your partner to a game of tug-o-war. The winner gets an extra credit point (just kidding).

Demonstrate your program to your lab instructor.

Print a copy of your code and affix it to you lab book.

Write a summary/conclusion in your lab book and present it to your lab instructor for grading. (The rubric is given in Lab 1)