Department of Electrical and Computer Engineering

EE 3710 Lab 5

Title: BCD Stopwatch with Serial Interface.

Objective: The student should be able to write, debug and test a program that uses timer and the serial port interrupts without busy/wait loops.

Parts: Project from Lab 4
9-pin D-sub cable

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 (stopwtch.asm) that meets the following requirements:

1. The system shall maintain an 8-bit BCD counter that represents elapsed time in the range 0.0 seconds to 9.9 seconds.

2. The LED bar graph shall display the 8-bit BCD counter value. A lit LED denotes a 1.

3. The system shall have two operating modes, run and stop. When in run mode, the counter shall increment at 10Hz. When in stop mode, the counter shall not change.

4. When the system is reset (either by power-on or by pressing the reset button) the system shall be in stop mode and the 8-bit counter shall be reset to 0.

5. The system shall have a serial interface configured for 9600 baud with 8 data bits and one stop bit. The serial interface shall be used to accept commands and report the time as shown below:

Command	Description	Response
R	Put system into run mode	none
S	Put system into stop mode	none
C	Clear the counter	none
T	Report Time	n.n CR LF

All other characters received on the serial port shall be ignored. The time report consists of the time (in ASCII) currently displayed on the LED followed by a carriage return (0DH) and a line feed (0AH).

6. The system shall have a start/stop button and a clear button. Pressing the start/stop button shall cause the system to toggle from run to stop mode or vise-versa. Pressing the clear button shall cause

the counter to be reset to 0. Buttons shall be debounced by sampling them periodically.

7. The program shall be written such that all work is performed in interrupt service routines (ISRs) and that there are no busy/wait loops in those routines. Once the system is configured and interrupts are enabled, the main program shall consist of an infinite loop that does nothing.

Create a project file for this lab, Add stopwtch.asm to your project and build it. Remember to include code to debounce your buttons. Make sure there are no assembly errors.

Note: You will need to use the 22.1184MHz crystal for this lab. You may use the sample code in echo.asm to enable the crystal and configure the serial port.

Note: The 10 Hz count rate is so slow you will not be able to use a timer (even Timer 2) to obtain it. Instead, configure a faster interrupt rate, say 100Hz (which is ideal for button sampling), and divide by ten in software (i.e. take action every tenth interrupt) to obtain the necessary 10Hz time base.

Lab Work: Connect your project and the serial cable as you did in lab 4 and download your program. Start a terminal emulator as you did in lab 4 as well. Start your program and verify that on reset that all LEDs are unlit.

Press the start/stop button. Verify that the bar graph increments 10 times per second. Press the clear button and verify that the LEDs start over from 0.0. Press the start/stop button again and verify that counting stops. Press the clear button. Verify that the count is reset to zero.

Type each of the commands listed above and verify that each performs the function specified. .

With the system in run mode, use the IDE to stop your program (red button). It should almost always stop in the infinite loop. If not, restart it (green button) and try again.

Print a copy of your assembly code and affix it to your lab book. Also include a summary/conclusion. Demonstrate your working program to the lab instructor. Be sure to stop your program with the IDE and demonstrate that the program stops in the infinite loop.