Department of Engineering

EE 3710 Lab 4

Title: Magic 8-Ball

Objective: The student should be able to write, compile or assemble, download and test a program that uses timers and the serial port.

Parts: Project from Lab 2
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 program in assembly 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.

2. The serial port shall be configured for 9600 baud with 8 data bits and one stop bit. Immediately after reset, a carriage return (ASCII 0DH) and a line feed (ASCII 0AH) shall be sent via the serial port.

3. Each time a button is pressed or any character is received on the serial port, exactly one of the 10 LEDs shall be randomly lit (or re-lit) and a corresponding message shall be transmitted via the serial port. The 10 messages are:

It is certain
You may rely on it
Without a doubt
Yes
Most likely
Reply hazy, try again
Concentrate and ask again
Don’t count on it
Very doubtful
My reply is no

Each message shall be terminated by a carriage return and a line feed.

Create a project file for this lab, Add your program to your project, build it and make sure there are no errors.

Note: An easy way to get a random number generator is to configure timer 0 in auto-reload mode and to decrement a count on each timer overflow, e.g.:

```
     jnb  tf0, continue   ; put this in main loop
     clr  tf0
     djnz random, continue
     mov  random,#10      ; random in range 1..10
continue:
```

Then, whenever the button is pressed or character is received, the value of random will be a random number from 1 to 10.

Note: The C8051F020 has two serial ports, so you will need to use registers SCON0 and SBUF0 instead of SCON and SBUF.

Note: After reset, the C8051F020 uses an internal 2 MHz system clock that is not suitable for generating an accurate 9600 baud rate. Fortunately, the microcontroller board contains a 22.1184 MHz crystal that can be used as the system clock, but it is tricky to turn on. First you must enable the external clock (register OSCXCN), wait for 1 millisecond, wait for the oscillator to become stable (i,e. wait for bit XTLVLD to become 1), then select the external oscillator as the system clock (bit CLKSL in OSCICN). Example code showing this procedure is included on the course website (echo.asm)

Lab Work: Connect your project as you did in lab 3 and download your program.

Use the oscilloscope (in single sweep mode) to observe the transmitted data on serial port 0 (available on P0.0). Verify that the carriage return and line feed are transmitted immediately after the program starts. Measure the duration of the first start bit and verify that it is 1/9600 second (104 µsec). Using the same technique, verify that 8/9600 seconds (833 µsec) elapse between the end of the start bit and the beginning of the stop bit.

Show the trace on your scope to the lab instructor and demonstrate how you made the measurements above.

Connect the serial port on your evaluation board to the serial port on your computer. Start a terminal emulator such as Hyperterminal or puTTY on your computer and configure it for 8 data bits, one stop bit and 9600 baud. (Note, if you are unsure you have configured your

terminal emulator correctly, you can check it by downloading, assembling and running the 9600 baud echo program from the course website.)

Start your program and Verify that on reset the LEDs are unlit.

Press the button and verify that one LED lights and that one of the 10 messages above is displayed on the terminal emulator screen. If not, use the debugging tools in the IDE to find and correct the problem.

Repeat the test multiple times and verify that the LED selection appears random and that the appropriate message is sent for each LED as it is lit.

Type any key on the terminal emulator and verify that one LED lights and that one of the 10 messages above is displayed on the terminal emulator screen.

Remove the USB adapter from your circuit and cycle the power on the board. Verify that it continues to function.

Print a copy of your code and affix it to your lab book, then add a summary/conclusion. Demonstrate your working program to the lab instructor.