TEXAS BUG

TExaS versions downloaded prior to 2/19/2014 did not properly simulate bit-specific addressing to I/O Ports A-F. So, if you installed TExaS prior to 2/19/2014 5pm Central Time, please download and install this patch. You will need to first install Keil before installing this update for TExaS. This installation only updates the DLLs in the to **Keil\ARM\BIN** folder, however, it is good practice to backup any Keil projects you have edited prior to installation.
1) **Download the TExaS patch** (http://edx-org-utaustinx.s3.amazonaws.com/UT601x/TExaS_Patch.exe) saving the 11.5M file on your computer.
2) Execute the **TExaS_Patch.exe** file to update all the DLLs.

PROCEDURE

The basic approach to this lab will be to first develop and debug your system using the simulator. You will get a lab grade for this simulation phase of development. After the software is debugged, you will interface actual components to the LaunchPad and run your software on the real microcontroller. You will get a second lab grade for this real-board phase of development.
Part a) You should draw a circuit diagram connecting the switch and LED to the microcontroller, similar to Figure 8.1.

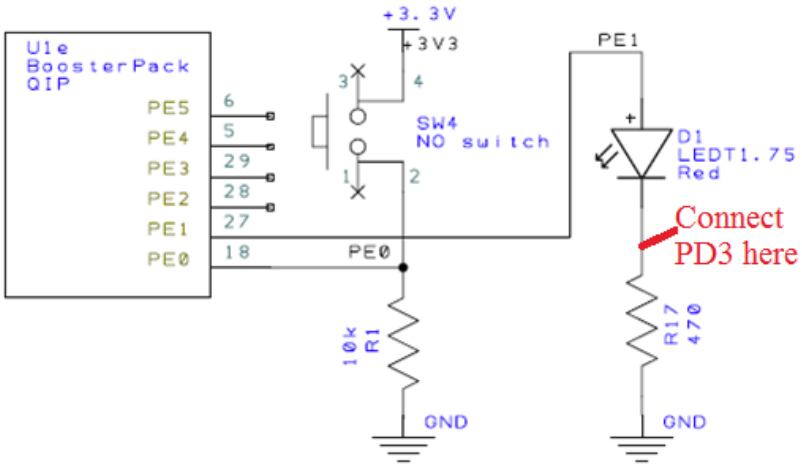

Figure 8.1. One possible Lab 8 interface connecting the input to PE0 and the output to PE1.

The lab describes using PE0 for input and PE1 for output, but Tables 8.1 and 8.2 show other options that the automatic grader can handle. You could connect the output to PA3, PB1 or PE1, and you could connect the input to PA2, PB0, and PE0.

Output	PA3	PB1	PE1

Table 8.1. Possible ports to interface the output (PE1 is default).

Input	PA2	PB0	PE0

Table 8.2. Possible ports to interface the input (PE0 is default).

Notice in Figure 8.1 that the PE0 and PE1 wires cross over each other in the circuit diagram, but they are NOT electrically connected, because there is NO dot at the point of crossing. Conversely, the PE0 signal to the microcontroller, pin2 of the switch and one end of the 10kΩ R1 resistor ARE electrically connected, because there IS a dot at the point of crossing.

Part b) Write the software that satisfies the requirements for this lab. The best approach to time delay will be to use a hardware timer, which we will learn in Lab 9. In this lab, however, we do not expect you to use the hardware timer. Again, use the logic analyzer on the simulator to verify the software operates properly.

Help

GRADING IN SIMULATION MODE

DR. RAMESH YERRABALLI: In this video, we'll

show you how to earn your grade for Lab 8 in Simulation.

We go to edX.

We copy the code that is given to you on the screen, which says Lab 8 Simulation

Grader, which is a four-digit number.

Copy it.

Go to Keil.

Make sure you hit the debug with the simulator.

Build the project, debug it, and we have a TExaS Interface

where we're going to paste the number

we got from edX.

DR. JONATHAN VALVANO: Now everybody's number is different.

DR. RAMESH YERRABALLI: Yes.

And this is nice, because here we see the Logic Analyzer.

We don't really need to see it, but it's a good way

to see how the grader is working.

And now we hit the grade button.

The grade button is going to run through a sequence of tests,

and once all the tests are completed, it's

going to tell you what your grade is.

Now, here's something you can look at in the command window.

It shows you the actual tests being performed, so if you're curious,

you can look at which tests you passed

Help