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. If you downloaded the graders for Labs 8-11 after 2/19/2014, you do not need the 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.

PREPARATION

You will need a LaunchPad, a switch, a 10kΩ resistor, an LED, and a 470Ω resistor.

STARTER PROJECT

Lab8_SwitchLEDinterface

PURPOSE

Lab 8 is our first lab requiring you to build circuits on the breadboard and connect them to the LaunchPad. The purpose of this lab is to learn how to interface a switch and an LED. You will perform explicit measurements on the circuits in order to verify they are operational and to improve your understanding of how they work.

SYSTEM REQUIREMENTS

In this lab you will build a switch interface that implements positive logic, and you will build an LED interface that implements positive logic. You will attach this switch and LED to your protoboard (the white piece with all the holes), and interface them to your TM4C123. Overall functionality of this system is similar to Lab 6, with five changes: 1) the pin to which we connect the switch is moved to PE0, 2) you will have to remove the PUR initialization because pull up is no longer needed. 3) the pin to which we connect the LED is moved to PE1, 4) the switch is changed from negative to positive logic, and 5) you should decrease the delay so it flashes about 5 Hz. To flash at 5 Hz means the LED comes on 5 times per second. If the switch is pressed we turn on the LED for 100 ms, turn off the LED for 100 ms, and repeat.
1) Make PE1 an output and make PE0 an input.
2) The system starts with the LED on (make PE1 =1).
3) Wait about 100 ms

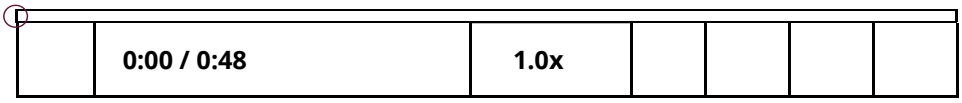
4) If the switch is pressed (PE0 is 1), then toggle the LED once, else turn the LED on.

5) Steps 3 and 4 are repeated over and over.

The grader will activate the PLL so the system runs at 80 MHz, you must not modify this rate.

WORKING LAB 8

Help



DR. JONATHAN VALVANO: Before we begin, let me show you how Lab 8 works.

Here we have a operating Lab 8.

I flipped the launchpad over and connected the wires

into the female sockets of the launchpad.

We have the ground.

We have power.

We have one input and one output.

And over on my protoboard, I have built the LED interface here

and the switch interface there.

I've downloaded the code, and it's running.

And if the switch is not pressed, the LED here should be on continuously.

And if the switch is pressed, the LED should flash.

That's it.

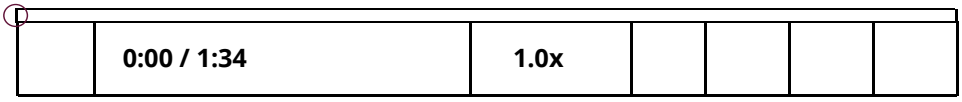
Not pressed, LED on.

Pressed, LED flashes.

And that's what we're going to build here in Lab 8.

HOW TO DEBUG WITH THE LOGIC ANALYZER

Help



DR. RAMESH YERRABALLI: Hi.

In this video, we'll show you the use of a nonintrusive debugging tool, which

is a logic analyzer, which can be used in the Simulation mode in Keil.

We run the project.

Once we run it, we will interact with our program using the TExaS Interface.

We will pull down the logic analyzer.

And that will pop up this window.

We add the things we want to watch.

In Lab 8, the things we want to watch are Port E in 1 and Port E in 0.

We make sure that we're viewing them as bits and not analog,

and we add them into our screen.

And now we can run our program, interact with our program

using the TExaS Interface, and we watch the behavior

and then we can stop it, analyze it.

We can measure things by using the cursor option, or we can zoom in

and then we can click between points of interest

and measure the difference between them.

Jon, what does this say?

DR. JONATHAN VALVANO: 83 milliseconds.

Perfect.

DR. RAMESH YERRABALLI: That's perfect.

So now you try it.

DR. JONATHAN VALVANO: And when you close the compiler,

it will automatically save those settings for next time.

DEBUGGING TIPS IN KEIL

Help

DR. RAMESH YERRABALLI: In this video we'll

show you some basic debugging tools that Keil supports.

DR. JONATHAN VALVANO: These debugging tools work both in the simulator

and on the Real Board.

DR. RAMESH YERRABALLI: So as an example, we'll take a program from