

[Courseware \(/courses/UTAustinX/UT.6.01x/1T2014/courseware\)](/courses/UTAustinX/UT.6.01x/1T2014/courseware)

[Course Info \(/courses/UTAustinX/UT.6.01x/1T2014/info\)](/courses/UTAustinX/UT.6.01x/1T2014/info)

[Discussion \(/courses/UTAustinX/UT.6.01x/1T2014/discussion/forum\)](/courses/UTAustinX/UT.6.01x/1T2014/discussion/forum)

[Progress \(/courses/UTAustinX/UT.6.01x/1T2014/progress\)](/courses/UTAustinX/UT.6.01x/1T2014/progress)

[Questions \(/courses/UTAustinX/UT.6.01x/1T2014/a3da417940af4ec49a9c02b3eae3460b/\)](/courses/UTAustinX/UT.6.01x/1T2014/a3da417940af4ec49a9c02b3eae3460b/)

[Syllabus \(/courses/UTAustinX/UT.6.01x/1T2014/a827a8b3cc204927b6efaa49580170d1/\)](/courses/UTAustinX/UT.6.01x/1T2014/a827a8b3cc204927b6efaa49580170d1/)

PREPARATION

You will need a LaunchPad and access to TM4C123_LaunchPadUsersManual.pdf (/c4x/UTAustinX/UT.6.01x/asset/TM4C123_LaunchPadUsersManual.pdf).

STARTER PROJECT

Lab6_BranchingFunctionsDelays

PURPOSE

The purpose of this lab is to learn simple programming structures in C. You will also learn how to estimate how long it takes to run software, and use this estimation to create a time delay function. You learn how to use the oscilloscope to measure time delay. C software skills you will learn include masking, toggling, if-then, subroutines, and looping.

SYSTEM REQUIREMENTS

The system has one input switch and one output LED. Figure 6.1 shows the system when simulated as the switch is touched. A negative logic switch means the PF4 signal will be 1 (high, 3.3V) if the switch is not pressed, and the PF4 signal will be 0 (low, +0V) if the switch is pressed. A positive logic blue LED interface means if the software outputs a 1 to PF2 (high, +3.3V) the LED will turn ON, and if the software outputs a 0 to PF2 (low, 0V) the blue LED will be OFF. Here in Lab 6, you first debug in simulation and then run on the real board, but no external components on the protoboard will be required. The switch and LED are already built into the LaunchPad. However in Lab 8, you will attach a real switch and a real LED to your protoboard (solderless breadboard) and interface them to your microcontroller. Overall functionality of this system is described in the following rules.

- 1) Make PF2 an output and make PF4 an input (enable PUR for PF4).
- 2) The system starts with the LED ON (make PF2 =1).
- 3) Delay for about 100 ms
- 4) If the switch is pressed (PF4 is 0), then toggle the LED once, else turn the LED ON.
- 5) Repeat steps 3 and 4 over and over.

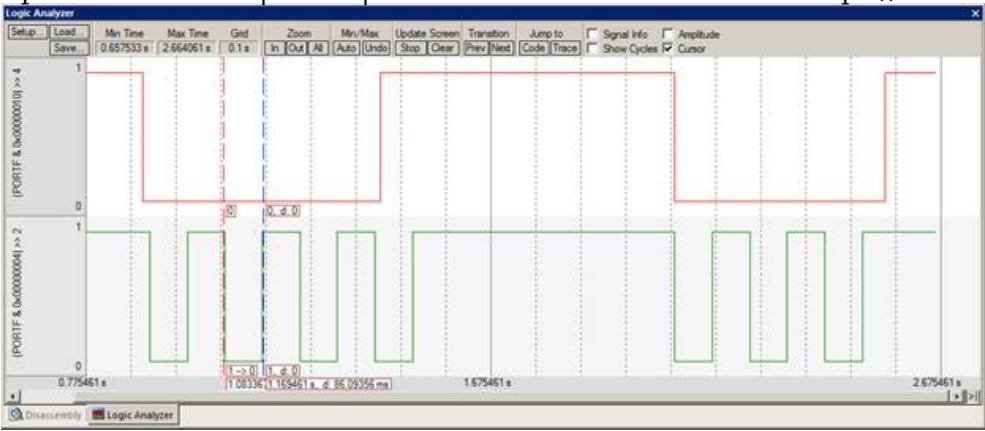


Figure 6.1. Example screenshot in simulation mode. Notice the output toggles at about 86ms when switch is pressed (PF4=0) and the output is high when the switch is not pressed. (the 86ms time in the simulator will result in 100 ms on the real board, because of a bug in the Keil uVision).

LAB 6

Help



PROFESSOR JONATHAN VALVANO: Let me show you what Lab 6A looks like when you're done.

To run the program, we have to make sure that we're set up so that we're using the Lab 6 real board here in the Options menu.

I build it, which is to compile.

I download, which is to burn the program on to the board.

I let me show you what it looks like when

	0:00 / 0:53	1.0x				
--	-------------	------	--	--	--	--





EdX is a non-profit created by founding partners Harvard and MIT whose mission is to bring the best of higher education to students of all ages anywhere in the world, wherever there is Internet access. EdX's free online MOOCs are interactive and subjects include computer science, public health, and artificial intelligence.



<https://courses.edx.org/courses/UTAustinX/UT...>
(<http://www.meetup.com/edx-Global-Community/>)



(<http://www.facebook.com/EdxOnline>)



(<https://twitter.com/edXOnline>)



(<https://plus.google.com/108235383044095082735/posts>)



(<http://youtube.com/user/edxonline>)

© 2014 edX, some rights reserved.

[Terms of Service and Honor Code](#) -
[Privacy Policy \(https://www.edx.org/edx-privacy-policy\)](https://www.edx.org/edx-privacy-policy)