(https://www.edx.org)

UTAustinX: UT.6.01x Embedded Systems - Shape the World

KarenWest (/dashboard) ▼

Courseware (/courses/UTAustinX/UT.6.01x/1T2014/courseware) **Course Info (/courses/UTAustinX/UT.6.01x/1T2014/info)**

Discussion (/courses/UTAustinX/UT.6.01x/1T2014/discussion/forum) **Progress (/courses/UTAustinX/UT.6.01x/1T2014/progress)**

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

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

Embedded Systems Community (/courses/UTAustinX/UT.6.01x/1T2014/e3df91316c544d3e8e21944fde3ed46c/)

Help

The entire game can be debugged in simulation using the LaunchPadDLL version 1.0.0.6 or higher. Unfortunately the speed of the game will be much slower than running on the real board. However, just like the other labs, debugging in simulation first allows you to debug software without worrying if the hardware works. The next video shows me running my game in the simulator. This program runs about 10 times slower than it does on the real board. So once I got it to run in the simulator, I had to re-tune all the timings for the game to run on the real board.

LAB15A RUNNING ON THE SIMULATOR

PROFESSOR JONATHAN VALVANO: Hi.

Let me show you how to debug the game in the simulator.

Here is the project, and in order to share your project with other students,

you will not be able to change the configuration of this project.

In particular, you will be able to add and subtract software

to this file called SpaceInvaders.c, and that will let the rest of the class

play your game.

We're going to run in the simulator, so things run very, very slowly.

And so I'm going to reduce the delay here

so that I don't have to wait a long time.

OK.

Let's run in the simulator.

So we'll check in the Debug option that we have the simulator.

This is LaunchPadDLL version 6.0 or bigger.

OK.

Build, Debug, OK.

Help

So all of the complements of the game exist here in the simulator.

Here we have the Nokia.

Over here is the game console.

There's the DAC on this one.

And so let me single step across the starter function.

This will set the PLL to 80 megahertz.

You see the PLL is now 80 megahertz.

There I'm initializing the random number generator, initializing the Nokia.

Clear buffer, if you remember, just clears a RAM buffer.

In a manner similar to Lab 14, we suggest you develop and test Lab 15 in a modular fashion. For examples, each of the modules could be independently developed and tested, before you begin combining modules into the system.

- 11kHz interrupts with DAC to play sounds
- 30 Hz interrupt with ADC input and semaphore link to main program
- Switch input
- LED output
- Low level Nokia LCD
- Drawing sprites

One approach is the develop and test each module on the simulator. Once the modules are tested you can put them together. After the game runs completely on the simulator, you switch over and debug on the real board.

A second approach is to develop and test a module in the simulator. Once the module works on the simulator, test that module on the real board. Either way we suggest you add only 5 to 10 lines of code and then test.

Press (https://www.edx.org/press) FAQ (https://www.edx.org/student-faq)
Contact (https://www.edx.org/contact)

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.

Help

(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)

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