Spring 2013 - CSE 325: Embedded Microprocessor Systems

Project 5: Let's Play a Game

Points: 10, Due Date: Apr. 08, 2013

Project Requirements

For this project you will use the LED matrix to implement the Game of Life. http://en.wikipedia.org/wiki/Conway's_Game_of_Life you can follow the link for the rules to the game. Essentially it is an animation that can be implemented on the Matrix, and it requires no interaction from the user.

Your program can operate in three modes, idle mode, download mode, and game mode. In Idle Mode, the serial communications module is disabled and no game is playing. When the system is in idle mode, if push button 1 on the microcontroller board is pressed and released, the system shall toggle to Download Mode. When the system is in Download Mode, if push button 1 is pressed and released it shall toggle to Game Mode. When in Game Mode, the system shall return to Idle Mode when push button 1 is pressed and released. If the starting game position array is empty, Game Mode shall immediately be terminated and the system shall skip ahead to Idle Mode.

Upon entering Download Mode, the system shall begin listening on UART1 for a download starting game packet from the host PC. As a starting position is being downloaded, push button 1 shall be ignored, i.e., pressing and releasing it shall do nothing. When a starting position is not being downloaded, but the system is listening for a download packet, pushing and releasing push button 1 shall terminate Download Mode and the system shall return to Idle Mode.

Starting positions are stored on the host PC as a binary file in any format that you choose. On the course website, there is a zip archive named SleazyTerm.zip**.** This archive contains a Java source code file, SleazyTerm.java that you can build and run to download files to the microcontroller. Extract this archive to a working directory and follow the installation instructions (documented in the source code file header comment block) to build it. It uses an open-source serial communications library named RxTx, so you have to make sure that the files RXTXcomm.jar and rxtxSerial.dll are in the same directory as SleazyTerm.class when you run the program.

In the final program that you submit you must have an example file for a starting position for the game of life, and a file explaining how to modify it so I can test your code.

What to submit for grading

Put a comment header block at the top of each source code file that contains: (1) the name of the source code file; (2) the lab project number; (3) your name (and your partner’s name); (4) your email address (and your partner's email address); (5) the course number and name, CSE325 Embedded Microprocessor Systems; and (6) the semester, Spring 2013.

We will export the project to a directory structure. In CodeWarrior, click File | Export on the main menu. In the Export dialog, expand General. Click on File System. Click Next. In the next dialog, click Select All. Enter a destination directory, e.g., C:\Temp. Click Create Directory Structure for Files. Click Finish. The entire project will be exported to C:\Temp\Proj03 (or whatever name you used for your project when you created it).

Zip this entire directory naming the zip archive **cse325-s13-p05-*lastname*.zip** or **cse325-s13-p05-*lastname1-lastname2*.zip** if you worked with a partner.

Upload the zip archive to Blackboard using the project submission link by the deadline. Consult the online syllabus for the late and academic integrity policies.