|  |  |
| --- | --- |
| \\cetus.ece.missouri.edu\users\fischerjd\Desktop\Current Semester\common\logo-clear.png | University of Missouri – Columbia  Department of Computer and Electrical Engineering  *“A Proud Tradition, Engineering the Future!”* |

Simon Says Cross Network   
Multiplayer Game

ECE 4220: Real Time Embedded Computing

March 31st, 2015

— Engineer—

Adam Speichinger, Electrical and Computer Engineering

— Course Instructor —

Luis Rivera

Topic to be addressed:

The topic that will be addressed in this project will be multiplayer gaming across a network. I decided upon this sort of project because I enjoy playing video games and wanted to experiment with a simplified game that could possibly lead to further game development ideas. This simple game will be a form of Simon Says where two players can play against each other in different game modes. The game will be played on an embedded system enabled board that will use server client communication in order to communicate back and forth between both boards. The gaming system will be using serial communication with the user’s controller, which will be a gaming controller plugged into a USB port of each board. The game will also be using a FIFO approach to the commands sent and received in order to check the user’s accuracy of buttons pushed.

Related systems:

For this project, I will try to use the TS-7250 embedded system boards in order to achieve this game. Since this board is supplied through the school it will make things cheaper to implement, and also prove to be a greater challenge since I haven’t worked with the TS-7250 board for very long. Using this board I will be able to achieve the goal of creating the Simon Says game that will communicate to another TS-7250 board and controller via a local area network (LAN).

If for some reason the controllers that I have chosen for this game are not compatible with the TS-7250 boards, then the implemented board will become a Raspberry Pi. Since I’m familiar with a Raspberry Pi board, I don’t want this to be my first choice in order to attempt learning how to use a variety of embedded system boards. The Raspberry Pi will still be able to perform just like the TS-7250 board will and the overall project would still be the same despite using the TS-7250 board or the Raspberry Pi.

Goals:

My goal for this project is to get it completely working on either embedded board using client server communication to communicate to each other.

Expected outcomes:

The expected outcome of this project is to create a working version of Simon Says that will allow two users to play against each other across a network. The first implementation will be that the two users will take turns entering in a series of buttons from their controller and then the opposing player will have to repeat the same buttons in the same time. If the opposing player is unsuccessful then the game will be over, if they are successful then the game will continue with the second player entering in more buttons than the first player could initially. The game will continue in this fashion, switching between players to enter buttons in the same manner that was just described, until a winner is decided.

Constraints:

Constraints to this project will be:

* time to implement the project
* time to work in the lab (if using the TS-7250 boards)

Assumptions:

I am assuming that the gaming controllers that I have chosen to use will be compatible with both the TS-7250 board and the Raspberry Pi board. I have been given the impression that both boards should be able to handle the controller given the correct drivers for the USB ports. Another assumption is that the project will be able to run successfully across a network instead of having to run on one board.

Methodology:

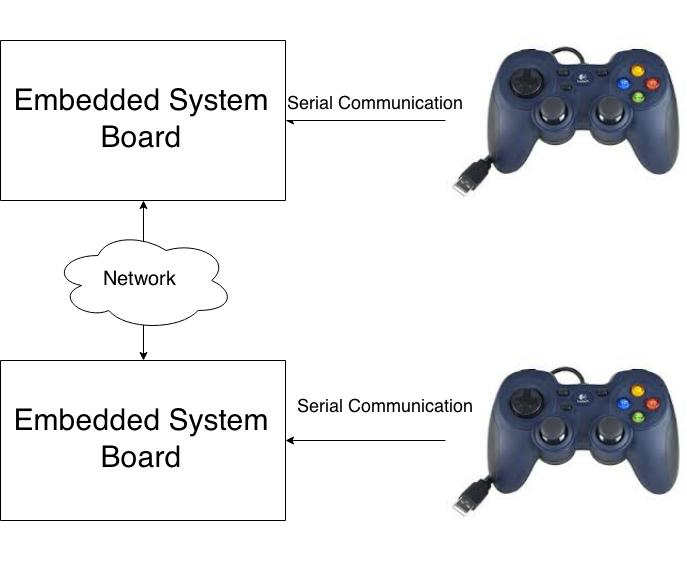


Figure 1: Flowchart of game

The above figure demonstrates how the game will be implemented. Since it is undecided what the board will officially be, for now it is left as just a general square. For the final report, a more appropriate flowchart will be demonstrated.Timeline:

This project will be completed in weekly sprints:

* 1. Sprint 1: April 6th -10th

Determine the board type

Get serial communication working with the USB ports

Start coding the project

* 1. Sprint 2: April 13th - 17th

Set up server client communication

Continue working on project code

* 1. Sprint 3: April 20th -24th

Have code working for single user by this point

Test out single user to verify the game works smoothly for one person

Start coding for two players

* 1. Sprint 4: April 27th - May 1st

Continue with two player code

Finish code for two players

Test out two players

* 1. Sprint 5: May 4th - 8th

Fix any issues with two players

Finalize project

Write up project report