

Final Project Proposal

To: A. Cano

From: Group 4 (Steven Eisinger, Kim Bhagwan)

Date: 7/12/16

Proposal

We would like to create one or several small arcade games using the LCD and input devices provided on the board such as the RPG and button, what we can switch between using external interrupts.

Overview

Creating a game would use the PIC's ability to define custom characters and create animations on the LCD (Chapter 7), use the rotary pulse generator (RPG) (Chapter 8) or potentiometer with A/D conversion (Chapter 10) as a controller and use external interrupts to switch between games (Chapter 9). Using the RPG and creating custom symbols on the display requires new reading that hasn't been explored in previous labs. The two game ideas we would like to implement are a space shooter game and a simple racing game. If this would take too much time, only one of the games will be implemented. These games can be implemented on the small 8x2 LCD provided with the QwikFlash board, but could be much more intricate on a larger LCD if one is available.

Tentative Task Outline

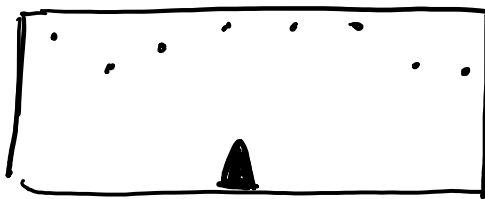
1. *Create custom game icons and display and animate them on the LCD.*
Read Chapter 7.7 to find out how to create custom icons for the LCD and display them.
To animate them on the display properly, use a subroutine such as LoopTime to control how often the display is updated with new icons. We must animate enemy ships for the space shooter and oncoming obstacles for the racing game.
2. *Make the game icons react to input from the button and RPG.*
For the space shooter, the ship should move left and right depending on the rotation of the RPG and should shoot when the button is pressed. For the racing game, this means the car must switch lanes (top and bottom rows of LCD) when the button is pressed and increase the frequency of oncoming obstacles when the RPG is rotated.
3. *Handle collision detection between objects on the LCD and handle game over states.*
For the space shooter, this means making an enemy explode after the player shoots it by pressing the button. In the racing game, this would mean having the car explode when the car crashes. The space shooter would display game over if an enemy ship reaches the player ship, while the racer would display game over once the player crashes. Display a score on the screen once this occurs based on the player's performance.
4. *Create a game select menu and use interrupts to go return to it.*
Using interrupts, display a game select screen where the player can choose which game to play by rotating the RPG then selecting it with the button. The player should be able to access this screen at any time while playing the games.
5. *(Optional) Generate sound effects for the games using the SPI.*

Using the SPI, we can generate simple waveforms which, when connected to a speaker, can serve as the sound effects for our games.

Notes:

- Due to the complexity of writing a game, the code should be written in a modular fashion.
- Each game and the menu should have its own subroutine, but games can share other subroutines with each other where appropriate (such as LoopTime).
- Below are some mockups of how the games should work.

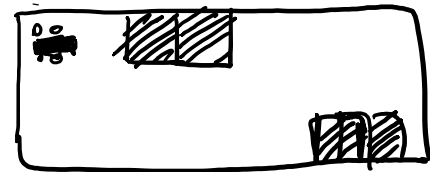
PIC Shooter



RPG: Move left/right
Button: Shoot

- Enemies fall from top of LCD.
- Player ship on bottom of LCD.

PIC Racer



2 players

- RPG controls frequency of obstacles.
- Button switches car lane.