High Level Design

Circuit CitEE

Irere Kwihangana
Lauren Mahle
Jaclyn Nord

12/16/2013

Table of Contents

1 Introduction

ADEC is the Association for the Disabled of Elkhart County. It is a program with several campuses and group residences throughout Elkhart and St. Joseph Counties. It is a group that is "working in partnership with all to help people with physical and developmental disabilities, live fuller, richer, more meaningful lives"(www.adecinc.com).

Our group is composed of Irere Romeo Kwihangana, Lauren Mahle, and Jaclyn Nord. Our aim is to help ADEC by creating a cause and effect game they can use in educational and recreational activities.

This documents outlines the problem put forth by ADEC and our proposed solution. It also explains in depth how the Circuit CitEE team intends to build a cause and effect game.

2 Problem Statement and Proposed Solution

ADEC has a need for a game to demonstrate the cause and effect relationship of a physical action. This can be demonstrated with using a game with a matrix of large push-able buttons similar to a Whack-a-Mole game.

The game Circuit CitEE intends to build will feature a sequence of randomly lit buttons that users must push. Failure to push the lit button in a specified amount of time will result in the game ending. The game will have varying levels of difficulty, allowing for different lengths of time before resetting the game or requiring multiple buttons to be pressed at once. The simplicity and the familiarity of the concept makes this solution ideal for ADEC.

3 Requirements

After consulting with ADEC, Circuit CitEE has determined a set of requirements for game. They are broken down by the stakeholders' requirements and the system requirements. The stakeholders are the staff members at ADEC and the players. They will be the ones setting up the system for use. The players are the people who ADEC serve. The requirements are as follows:

- The player shall be able to push a lit button
- The player shall be able to hear a tone when a button has been pushed
- The player or staff member shall be able to select a mode of difficulty (easy, medium, hard, progressive) before starting to play
- The player shall be able to see their score on an LED screen
- The staff member shall be able to turn on the system
- The staff member shall be able to see the status of the system (on or off)
- The staff member shall be able to change the level of the tone on the system
- The staff member shall be able to enter the identity of the player
- The staff member or player shall be able to select the player before starting to play (To Be Confirmed)
- The staff member shall be able to interface with the board game using a custom interface (To Be Determined)
- The system shall be able to be placed on a flat surface

- The system's buttons shall have different colors
- The system shall recognize a button press from the player
- The system shall be able to turn on an LED when button is pressed
- The system shall be able to play tones
- The system shall be able to display to the score of the game session to the player or the staff member
- The system shall be able to record a play session (identity of the player, duration of game, date and time of play, mode of play, and score)
- The system shall be able to run on a modified ATX power supply
- The cost of design and production shall not exceed $500

4 System Block Diagram

4.1 Overall System:

- User Interface:
    - Big Dome Push Button
        - Input
            - User pushes button
        - Output
            - Light (LED)
            - Sound
    - Selection buttons
        - Input
            - User pushes button
        - Output
            - Level Selection (to the program)
            - Sound
    - LCD Display
        - Input
            - Microcontroller sends level and score from the program
        - Output
            - Selected Level
            - User's Score
    - Speakers
        - Input
            - Program prompt
        - Output
            - Sound
- Software:
    - Level Selection
    - Game Generation
    - Score Tracking
- Hardware:

- Power Supply with a voltage regulator
- Board (wood or plastic)
- Microcontroller

A block diagram of the system is shown below in Figure 1.

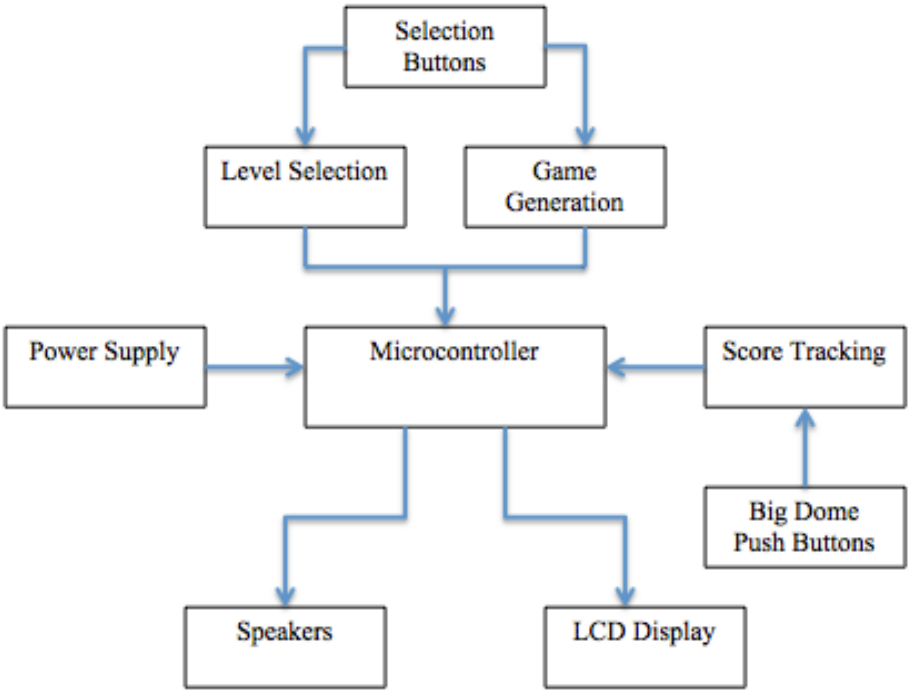

Figure 1. A system block diagram for the cause and effect game.

4.2 Subsystem and Interface Requirements:

Listed below are the requirements of the subsystems and user interface:

- User Interface:
  - Big Dome Push Button
    - Input
      - User pushes button
        - The press of a button will act as the input for a running program. If the user selects the lit button, the input will return the command to continue running the program. If the user selects an unlit button, the input will return the command to move to end game sequence. If the use does not select the

lit button in the allotted time, the input will return the command to move to end game sequence.

- Output
  - Light (LED)
    - If the selected program is running, a button will be lit by a light (LED) inside the Big Dome Push Button, indicating the button to press for the next input sequence. The Big Dome Push Buttons will be different colors.
  - Sound
    - Upon correct button press, a sound will notify the user that he or she hit the correct button. Upon incorrect button press, a different sound will notify the user that the game has ended.

- Selection buttons
  - Input
    - Button press
      - Upon pushing a selection button for a level (easy, medium, hard, progressive), the program will receive the selected level.
  - Output
    - Program will initialize corresponding level of the game.
    - Program will send level selection to microcontroller to output to LCD.
    - A unique sound will indicate which level has been selected.
- LCD Display
  - Input
    - Inside the user selected level, the program will keep track of the user's score. The program will pass user's selected level and current score to the LCD Display.
  - Output
    - The LCD Display will show the user selected level sent from the microcontroller.
    - The LCD Display will show the user's score sent from the microcontroller.
- Speakers
  - Input
    - The program will send a signal to the microcontroller when the user selects a level, pushes the correct button, pushes an incorrect button, thus ending the game, or achieves a new high score.
  - Output
    - The speakers will emit a unique sound when the user selects a level, pushes the correct button, pushes an incorrect button, thus ending the game, or achieves a new high score.

- Software:
  - Level Selection
    - Will require input from the user in the form of a button push. The selection will determine which of four levels the program will run (easy, medium, hard, progressive)
  - Game Generation
    - Will require a generation of random sequences with each button press. The frequency and duration of each random generation will be determined by the level chosen by the user. Must be able to maintain until the user fails to press the correct button, or beats the game.
  - Score Tracking
    - Program will need to track the number of correct button presses the user makes until the game is over. This number will then need to be displayed, and stored for an individual user.
- Hardware:
  - Power Supply with a voltage regulator
    - The board will need to be compatible with a standard wall outlet (120V), and will need to be able to convert this voltage into a usable level for various parts of the hardware, including the microchip and the lights in the push buttons. To accomplish this we will be using a modified ATX power supply.[1] The voltage regulator will allow us to control exactly how much voltage is sent to the board.
  - Lights (LEDs)
    - There will be a light (LED) inside of each Big Dome Push Button in order to illuminate it when appropriate. The LED will be a white or yellow.
  - Board (wood or plastic)
    - The buttons will be part of a board that will either be made of wood or plastic. This board will allow the user to place it on a flat surface.
  - Microcontroller
    - The microcontroller will be compatible with the voltage of the board. It will also have the necessary processing power.

4.3 Future Enhancement Requirements

A possible future enhancement is the addition of a stored high score system. This system would allow each individual user to have a high score stored inside the program, which could be accessed when they begin to play the game. The addition of this program would need to include a method to add user names, and later access them when a player returns to the game. This is also addressed in the section 6 (Open Questions).

5 High Level Design Decisions

Below are the decision for the high level system requirements. They are broken down by interface and subsystem:

- User Interface:
    - Big Dome Push Button
        - Input
            - User pushes button
                - The press of a button will act as the input for a running program. If the user selects the lit button, the input will return the command to continue running the program. If the user selects an unlit button, the input will return the command to move to end game sequence. If the user does not select the lit button in the allotted time, the input will return the command to move to end game sequence.
                    - The software will run as a case statement combined with sets of if/else statements. The input for a Big Dome Button press will be entered into an if/else set of statements. If the button that is pressed is the correct one, it will run the code to randomly select which button will be illuminated next and increment the user's score. If the button pressed is not the illuminated one, it will run the code to end the game, display the final score, compare the final score to the user's high score, and store the information. If the final score is higher than the user's current high score, a signal will be sent to the microcontroller to be output the sound from the speakers. The new high score will be saved in the system under the user's information.

        - Output
            - Light (LED)
                - If the selected program is running, a button will be lit by a light (LED) inside the Big Dome Push Button, indicating the button to press.
                    - The selection for this button (and the light inside) is determined by the random generator for a given program. The light will be powered through the board itself, which will be connected to the button.
            - Sound

        - Upon correct button press, a sound will notify the user that he or she hit the correct button. Upon incorrect button press, a different sound will notify the user that the game has ended. Upon selection of a level, a sound will notify the user that a level has been chosen. Upon achieving a new high score, a sound will notify the user.
            - An if statement will be embedded inside the if statements for deciding if the button press was correct or not. There will be software that will run the program to send a signal to the microcontroller to emit a sound from the speakers. Which sound played will be dependent upon which part of the if statement the button press goes through (the correct or incorrect aspect of the program). The sound will last for 0.5 seconds for a correct answer, and 3 seconds for an incorrect answer, thereby signaling the end of the game. A high score will have a separate 3 second tone to alert the user to the new high score.
- Selection buttons
    - Input
        - Button press
            - Upon pushing a selection button for a level (easy, medium, hard, progressive), the program will receive the selected level.
                - The selection button acts as the input for a case statement. The case statement will have cases for each of the four levels(easy, medium, hard, progressive). Within each of the case statements will be the commands necessary to run the respective program.
    - Output
        - Program will initialize corresponding level of the game.
            - The program will move to the correct case in the case statement, which will give the command to run the program for the selected level.
        - A unique sound will indicate which level has been selected.
            - Within the case statement that will determine which level will run will also be the software code necessary to produce a sound that is unique for each selection button. Each sound will run for 0.5 seconds.
- LCD Display

        - Input
            - Inside the user selected level, the program will keep track of the user's score. The program will pass user's selected level and current score to the microcontroller.
                - With each correctly pressed button, a counter will increment the score within the program for the selected level. (At the beginning of each game the counter will reset itself to 0). The information from the counter will be sent to the output program for the LCD, where it will be displayed on the board.
        - Output
            - The microcontroller will output the selected level and user's score to the LCD Display.
                - Within each case for the level selected, the program will keep track of the user's score. The score will increment with every correctly pushed button until the game is ended by the user pushing an incorrect button. At that time, the program will send the value of the level, based on the case that was run, and the total score at the end of that case. These values will be sent from the microcontroller to the LCD Display, so that the user will know what score they have reached for the level selected.
    - Speakers
        - Input
            - The program will send signals to the microcontroller when a sound needs to be played. This will happen when a level is selected, when the user presses the correct button, when the game is over, or the user has achieved a high score.
                - The program will track when any of these instance occur and send a signal to the microcontroller dictating the sound and duration.
        - Output
            - The microcontroller will send the sound and its duration to the speakers to be played based on the appropriate occurrence. A unique sound will be emitted when a level is selected, when the user presses the correct button, when the game is over, or the user has achieved a high score.
- Software:
    - Level Selection
        - Will require input from the user in the form of a button push. The selection will determine which of four levels the program will run (easy, medium, hard, progressive)
            - Level selection will run as a switch case with the input coming from the button selection. When a selection button is pressed, the program receives that input value as being

true and the corresponding case will be initiated. The program will run the case for that particular level selection button pressed.

- Game Generation
  - Will require a generation of random sequences after each button press. The frequency and duration of each random generation will be determined by the level chosen by the user. Must be able to maintain until the user fails to press the correct button, or beats the game.
    - The game is generated once the user has selected which level he or she is going to play (see Level Selection). Within each case there will be code to generate the random sequence, update the counter (score), output the power to light up a button, accept the input from a button press, determine the outcome of a button press, and run through the if/else statements.
- Score Tracking
  - Program will need to track the number of correct button presses the user makes until the game is over. This number will then need to be displayed, and stored for an individual user.
    - Software for each level will include a counter that will track the number of correct button presses the user makes. Each correct press will add 1 count to the counter. The counter will be sent after each button press to the program code for the LCD output, and will be displayed on the screen. When the user chooses an incorrect button, the current counter number is the final score for the game. The final score will be compared to the high score for the user. If the final score is higher than the current high score, the program for a new high score will be run.
- Hardware:
  - Power Supply with a voltage regulator
    - The board will need to be compatible with a standard wall outlet (120V), and will need to be able to convert this voltage into a usable level for various parts of the hardware, including the microchip and the lights in the push buttons. To accomplish this we will be using a modified ATX power supply.[1] The voltage regulator will allow us to control exactly how much voltage is sent to the board.
  - Lights (LEDs)
    - There will be a light (LED) inside of each Big Dome Push Button in order to illuminate it when appropriate. The LED will be a white or yellow.
  - Board (wood or plastic)

        - The buttons, speakers, and LCD Display will be part of a board that will either be made of wood or plastic. This board will allow the user to place it on a flat surface.
- Microcontroller
    - The microcontroller will be compatible with the voltage of the board. It will also have the necessary processing power.

6 Open Questions

The random number generator is an important aspect of the program we are creating, because it is necessary to ensure that there is no a predictable pattern during the game. Determining whether or not the program is actually generating a random sequence is an important part of the project. The software could use a built in random sequence generator. A hand programed one could also be built into the software by using a prime number sequence.

Another important part of the project is to provide the staff members the ability to enter new players and select the player. One way such a function could be achieved would be to add a raspberry pi to enable serial communication with the board. Adding such a device would give the staff a wide range of possible device to enter names, such as USB keyboard and other devices that can naturally interface with it. Also, adding the raspberry pi could enable automatic logging and analysis of the data collected by the game. Such function could be implemented using a Python program, which Raspberry pi can run.

7 Major Component Costs

Big Dome Push Buttons (9 units) - $9.95 each
Speakers (2) - ranging $2-$20
Small buttons to select mode[2] - $4.95 total
Board (size is to be determined)
Machining cost (will depend on size)
Power supply[3] $12.99
Microcontroller board

8 Conclusions

While there are several very different aspects of engineering that will be involved in this senior design project, we are confident that we will be able to master and employ them all. This particular project is important to us because it has a real world application

and a real world customer. At the end of this project, we plan on giving our finished product to ADEC in the hopes that they will be able to use it in their programs.

References

1 http://www.robotshop.com/media/files/PDF/hacking-atx-power-supply-bb-atxra.pdf
2 https://www.sparkfun.com/products/10302
3 http://www.newegg.com/Product/Product.aspx?Item=N82E16817170012