EE 478 Capstone Final Report
RFID Interaction Suite

Alyanna Castillo
Patrick Ma
Ryan McDaniels

Contents

List of Tables

List of Figures

1 ABSTRACT

2 INTRODUCTION

These are some example of how to cite and use bullet points A reference to Table 1 and one to the Design Specification, Section 3.

- Overall summary description of the module - 2-3 paragraphs maximum (explanation of use cases goes here)
  - Specification of the public interface to the module
    - Inputs
    - Outputs
    - Side effects
  - Psuedo English description of algorithms, functions, or procedures
  - Timing constraints
  - Error handling

3 DESIGN SPECIFICATION

3.0.1 Design Overview

This system is a RFID-based gaming system. A user is able to play alone against a computer, or can play with other users owning a system using a multiplayer connection feature. Additionally, users can create their own customized cards using a third built-in function of the device.

3.1 Design Requirements

3.1.1 Environmental Requirements

The device must operate in an indoor/outdoor environment with relative humidity consistent with desert and tropical environments. It must be durable enough to withstand various types of users, especially small children. The unit is to be portable and battery operated.

3.1.2 System Input and Output Requirements

The system must be capable of accepting several different types of signals and inputs:

- Standard frequencies used for close-range or Near-Field Communication Radio Frequency Identification (RFID) devices
- Standard frequencies used for commercial wireless communication standards such as wifi networks
- User input from a keypad to navigate menus, enter commands, and provide other alphanumeric information

- When in multiplayer mode, communication and commands from other players must be accepted and responded to.

The system must be capable of providing the following outputs:

- Display information about the current game state through an LCD screen
- Provide status information of individual cards placed on the game board
- Send commands to other systems when in multiplayer mode
- Send wireless signals to program RFID-enabled cards

3.1.3 User Interface

The system must also have the following buttons, switches or interface devices:

- 16-button keypad with alphanumeric character labels
- Reset button that when pressed causes a soft reset of the system
- Power button that turns the system on and off
- An LCD screen viewable from several angles. The screen must be viewable indoors and in overcast conditions

An example User interface is shown in Figure 1.

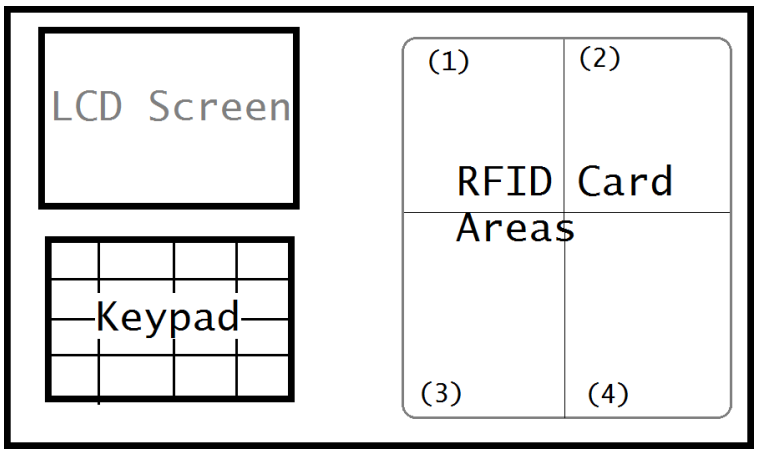

Figure 1: An example front panel layout for the system

3.1.4 Functional Requirement

The system must support the following modes of operation:

Single Player: Allows the user to select from, and play, single player games. Upon entering Single Player mode, the system prompts the user to select a game to play from those stored in memory. Selecting a valid game loads the game from memory and begins game execution.

Multiplayer: Allows the user to select and play multiplayer games. Upon entering Multiplayer mode, the system activates wireless communication and attempts to connect to other users in Multiplayer mode. Once a connection is established, the users select a multiplayer game on both systems and gameplay begins.

Build Card: Allows the user to create custom cards for a game. To create a custom card, the user will input the data via the keypad. The data stored will vary based on game requirements. All cards must contain unique card serial numbers or identifing ID numbers, and a code that specifies which games the card is valid for.

In all three modes, the user will be instructed by the programmed game to place a card in the RFID reading area to interact with the game and control events.

3.1.5 Operating Requirements

The system must operate in a standard commercial or household environment. The system must be portable, wearable on the user's arm, and operate on an internal power supply.

3.1.6 Reliablility and Safety Requirements

The system must be compliance with communication and electromagnetic radiation standards including those from the U.S. Federal Communication Committee, and applicable state and federal child safety laws.

The biggest problem with the final version of this project that was present at demo time was the fact that the multiplayer features were not implemented. There was test code that correctly configured the Xbee modules for multiplayer, but they were not completely implemented with the game. The reason for this was because when the I^2C system was implemented, the entire game had to be modified to run within the scheduler, when it was just a single function before. I^2C communication is controlled by the system's interrupt handler, and certain flags are set depending on whether data is being sent or received. Those flags have to be processed, and a game that is running in a function and taking control of the entire system would not allow for those flags to be processed. The time it took to port the game over to running completely within the scheduler made it impossible to get the multiplayer functions completely implemented and working in time.

Another problem at the time of the demo was that card reading was not 100% functional. The system had four distinct sets of data that could be successfully written to a card using the "Build Cards" option from the system's main menu, but the data was not being displayed properly in the singleplayer game. Data coming from the card over the I^2C connection was confirmed to be correct, but the game was not interpreting and displaying the information correctly to the user. This was also a matter of running out of time. For the same reasons as above (converting the game to be run within the flag-based scheduler) the RFID reading still had some kinks to work out at the time of the demo. Those problems were that:

- Monster type was being read incorrectly
- Monster name was being displayed incorrectly
- Complete attack list was not implemented

For demo purposes, only the first attack would be read and it would be copied to all three attack slots. Only the "FAIL" attack ID was programmed to be read, and if the ID did not match the "FAIL" attack's ID, then it was interpreted as a "SCRATCH" attack. This function worked correctly, but the rest of the attack IDs were not implemented. The monster's level was correctly read as well.

Finally, there was an error when finishing a singleplayer game. When the game was complete, the main menu was not being properly displayed. Once again, the problem was time, and the main menu's controls were still functional. Another menu could be loaded by navigating the invisible menu and choosing an option, letting that screen load, and then pressing the "B" key to return back to the main menu and redraw it.

6.4 Analysis of Implementation Issues and Workarounds

7 TEST PLAN

7.1 Test Specification

7.2 Test Cases

8 SUMMARY AND CONCLUSION

8.1 Final Summary

8.2 Project Conclusions

A BREAKDOWN OF LAB PERSON-HOURS (ESTIMATED)

Person	Design Hrs	Code Hrs	Test/Debug Hrs	Documentation Hrs
Patrick	x	x	x	x
Alyanna	x	x	x	x
Ryan	x	x	x	x

By initializing/signing above, I attest that I did in fact work the estimated number of hours stated. I also attest, under penalty of shame, that the work produced during the lab and contained herein is actually my own (as far as I know to be true). If special considerations or dispensations are due others or myself, I have indicated them below.

B BILL OF MATERIALS

Table 1 lists the bill of materials for the construction of one system.

Bill of Materials

Item	Unit Cost	Quantity	Total Cost
TI HI-Plus RFID Tags	0.91	20	18.20
DLP-RFID2 RFID Reader/Writer	50	3	150
Xbee S1 Wireless Chips	30	2	60
PLA Makerbot Filament	48	1	48
GAL22V10D	3.5	4	14
PICKit 3	45	2	90
CY7C128A SRAM	4	2	8
3.3 Volt Linear Regulator	3.22	2	6.44
Lever Switch, micro SPDT, momentary	2.5	6	15
16-key Numeric Keypad	7.5	2	15
128x169 Color LCD	17	2	34
PIC18F46K22 Microcontrollers	7.7	4	30.8
RGB LED, Common Cathode	4	8	12
(EXTRA)			
(EXTRA)			
(EXTRA)			
Total Cost			

Table 1