Middle East Technical University

Computer Engineering Department

Design Project

+18 SOFTWARE

Presents proudly

"HitMe"

Final Design Report

18/01/2009

Table of Contents

TABLE OF FIGURES

1 INTRODUCTION

1.1 Project Definition and Goals

Our senior project is a kind of laser tag game in which players use guns and wear special designed cloths. This game gives players the opportunity to play individually or by creating their own teams. As can be guessed, main goal of the game is to hit opponent members in the game area.

All players have their own guns which will be a designed circuit which has infrared LED as firing component. At the gun there will be a lens which is located in front of the infrared LED. This lens is used to collect infrared light beams into a linear infrared light beam. This infrared light which comes from players' guns will be captured by a sensor placed on the players' bodies. Besides guns, players wear special designed cloths on which an LCD is attached used to show health and bullet information of the player and infrared sensors are attached to capture infrared light beam are placed.

Some bonus devices are located throughout the game area to make this game funnier. Players can shoot these bonuses to increase their health or number of bullets on their guns. Another special bonus which locks the opponent members' guns for a predefined time period is decided to increase the importance of bonuses.

A graphical user interface will be designed to control the game at the server side. User who manages the game can initialize game settings and modes with this GUI. Health, scores and teams of the players can be seen in this GUI at the run time after the game starts. Player and bonus devices send information with their RF kits to server and received information is processed at server side to be displayed by GUI. In addition, there is an option to look at the high scores of the players. To decide high scores, this program saves all game data when the game is ended. User can stop the game at any time during the play and then restart the game with the same properties of players and bonuses. With all these features of GUI, user can manage the game easily.

1.2 Purpose of Document

The purpose of this report is to show our final design concepts about project. In this report, we improved some parts of our design according to initial design report. This report is about final design of the following parts of the project.

- software components (server module, microcontroller module and bonus module)
- hardware components (RF device, microcontroller)
- complementary components (optics module for infrared receive and transmit)

2 DESIGN CONSTRAINTS

2.1 Resource Constraints

We need datasheets of the devices and some additional materials about the software component we will be using while coding for microcontroller. Some of the documents were supplied by our teaching assistant and also we will use internet resources whenever we need extra information. Additionally, since we are using PIC18f2553 as a microcontroller and this microcontroller is not a common one, we have some troubles in integrating our code onto microcontroller. Sometimes, this limits our development.

2.2 Power Constraints

Since we are designing real platform game and the players will be carrying their own guns and sensors on their jackets, the power resource must be light enough not to disturb the players. The power supply must also be strong enough to feed the components and there must not be any problem caused by the power at any device during the whole game.

2.3 Time Constraints

The deadline of our project is June 2009 and also we should provide a prototype at the end of this semester. Therefore, especially for an embedded project, time constraint is the most important one. To use our time efficiently we must follow our schedule strictly.

2.4 Manufacturing Constraints

We will design the guns and the cloths for players on our own, but we have no experience on designing this kind of equipments. Integrating our circuits to the guns and to the cloths professionally will require too much effort.

2.5 Performance Constraints

As the number of players increases, there may be a huge traffic of interrupts that must be responded correctly. In such cases some of the interrupts may be lost if coding is not good enough. To avoid that kind of scenarios, our coding must be so efficient and well designed.

2.6 Experience of Members

Lack of experience of the team members on coding for an embedded device is one of the restrictions. Sometimes we may have difficulties with managing unexpected problems and unforeseen details of the project.

3 GAME CONTENT

As described in the project definition part, the aim of this game is to gain points by shooting opponent players. The main elements of the game are players, bonuses and as coordinator Server. Interaction between these elements constitutes the game logic.

Firstly, players are the most important elements of the game, because any interaction is triggered by players' actions. To clarify the actions of the players, electronic equipments that compose the player kit should be explained in detail. These equipments are guns and specially designed cloths. The former one consists of infrared (IR) LED and lens. When the player pushes the trigger IR led will send spread IR light and passing through the lens placed in front of the gun, this spread light will become a straight beam. Hence, the player will be able to shoot other players and the bonus packs. The event of being shot is understood by IR sensors attached to the cloths, and the player will be notified by a sound device, buzzer. The buzzer will also notify the player when she/he is out of bullet. Besides sensors, the health degree will be visible in an LCD placed on players' arms. Player can see his/her health and bullet information in this LCD.

Second element is bonus packs which are electronic devices. They can get shot by a player, giving him/her the specified bonus which can be either health bonus or bullet bonus or locking opponent's guns bonus. Bonus packs will be activated at a certain time defined by the server and will stay active until defined time period expires. If a player shoots a bonus during this time period, the bonus pack will be deactivated. Bonus packs' design is nearly same as the player kit except that the bonus packs will not have IR LED, as they will not shoot.

All these shooting stuff and other processes will be coordinated by a server. The communication between the server, players and the bonuses will be handled by RF devices, and the player will not be aware of this process.

4 SYSTEM ARCHITECTURE

In system architecture, first a general overview of the system will be given, and then architectural design of hardware and software will be explained in detail separately.

4.1 Overview

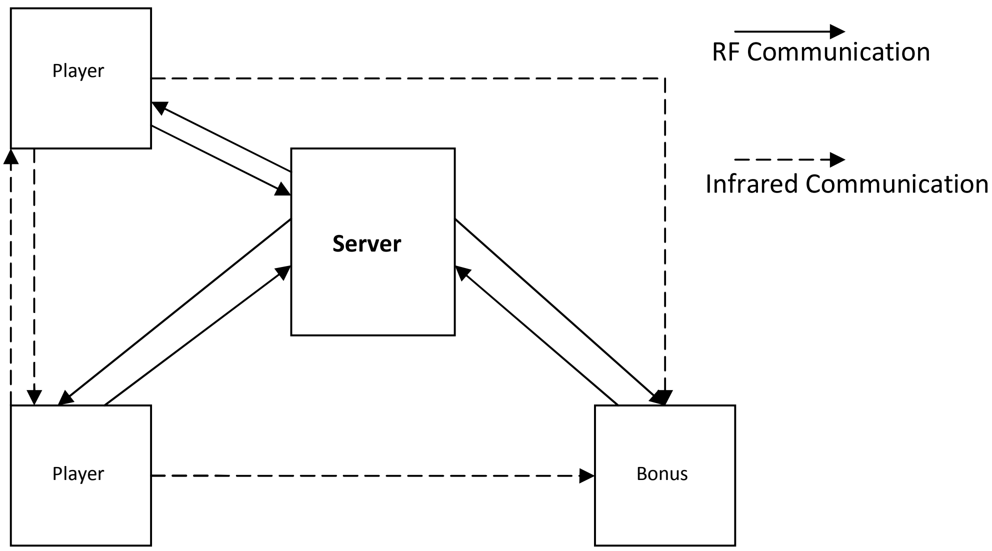

Figure 1: General Overview of the system

Our system consists of 3 main elements which are server, bonus and player. The module based structure of these elements is given below.

- **Server**

Server device consists of bidirectional RF communication module (server to bonus, server to player, player to server, and bonus to server) and a PC.

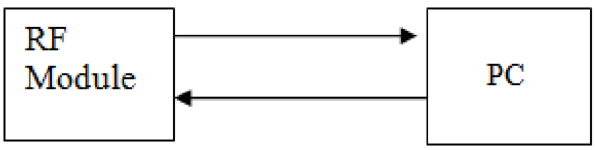

Figure 2: Server Device

- **Player**

Player device consists of bidirectional RF communication module (server to player, and player to server), bidirectional IR communication module (player to player, and player to bonus) and a microcontroller module.

Figure 3: Player Device

- **Bonus**

Bonus device consists of bidirectional RF communication module (server to bonus, and bonus to server), unidirectional IR communication module consisting of only IR receiver (player to bonus), and a microcontroller module.

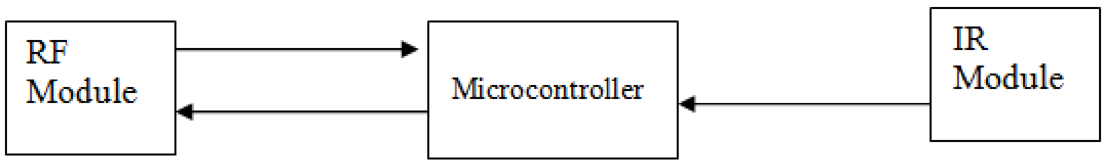

Figure 4: Bonus Device

4.2 Architectural Design

We have decomposed our design to 2 main parts which are hardware architecture and software architecture.

4.2.1 Hardware Architecture

In figure 5 below, the hardware components that each element in the system has can be seen. Also, the interaction between them can be observed.

Hardware architecture is composed of a microcontroller module, PIC18F2553, infrared (IR) communication module (IR LED, and IR sensor), RF communication module (XBee). These modules will be explained in detail in the following sections.

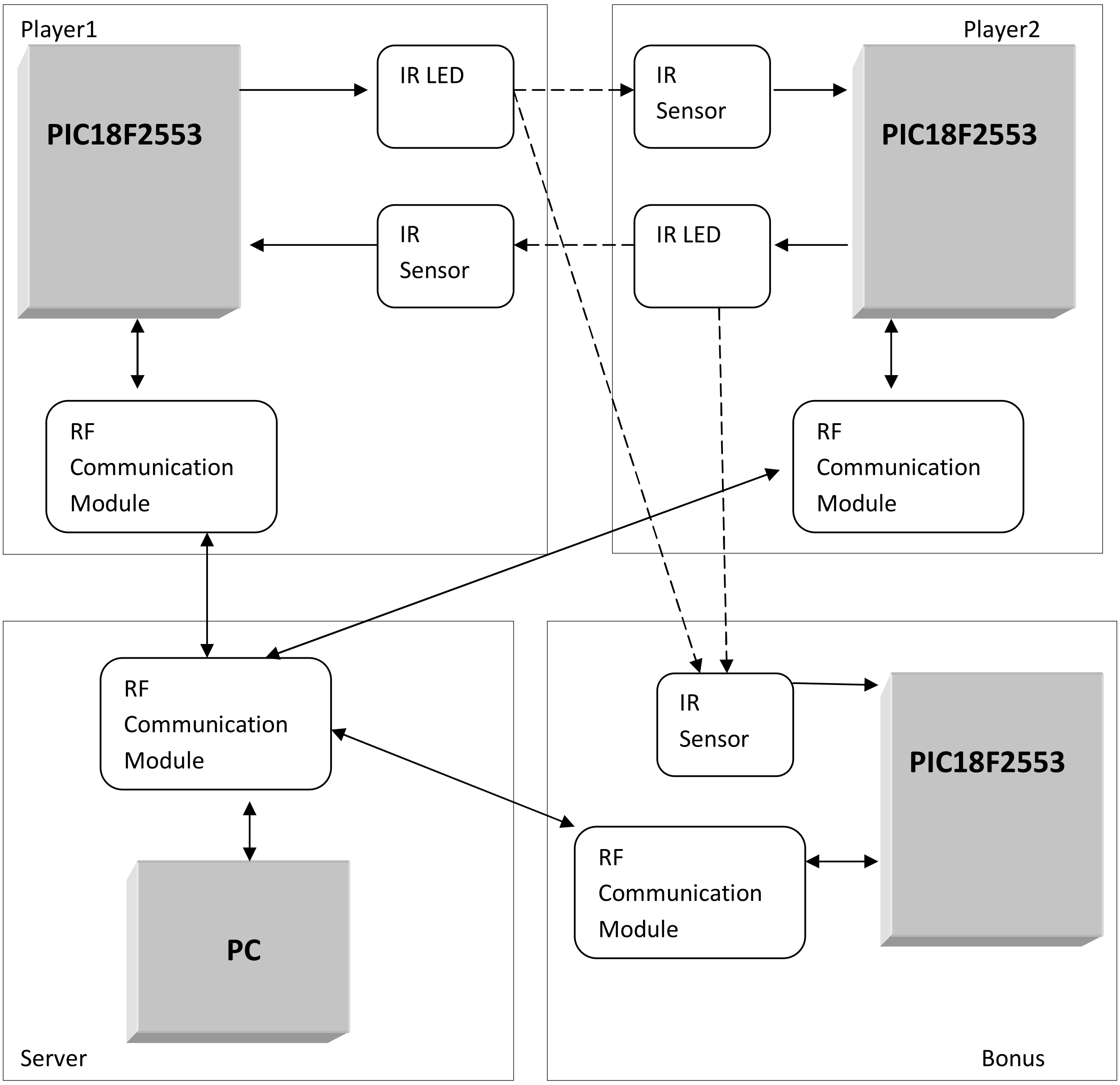

Figure 5: Overview of the Hardware System

4.2.1.1 *Microcontroller Module*

The main part of the hardware system is the microcontroller module. As microcontroller, we use PIC18F2553, and as a development board, we use USB Bit Whacker - 18F2553 PTH Kit which is easy to use and assemble. The assembled PIC board looks like the figure 6.

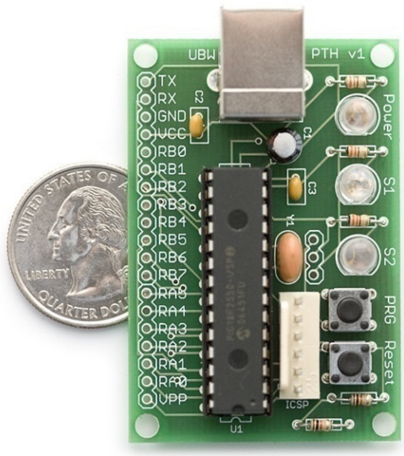

Figure 6: USB Bit Whacker - 18F2553 PTH Kit [1]

The following hardware components are connected to the microcontroller:

1- Health and Bullet Indicator LCD (as Bonus indicator in bonuses)
2- RF communication
3- IR LED (not in bonuses)
4- IR Sensor
5- Pushbutton (not in bonuses)
6- Sound Indicator(buzzer) (not in bonuses)

The user is informed about his/her health and bullet condition by an LCD screen which is placed on players' arm. This LCD is connected to the PIC by RA0, RA1, RA2, RA3, RA5 and RB7 pins whose detailed information will be given in "User Interaction Hardware Modules" part.

The RF communication will be handled by XBee device (XBee Pro Series 2.5). This module uses serial communication channel of PIC18F4553 and is connected to microcontroller's TX/RX pins.

The infrared LED and sensor connections are made via RB1 and RA4 pins, respectively. The sensor uses the timer0 interrupt for infrared data receive. When infrared light is sensed by IR sensor, timer0 interrupt fires and microcontroller starts to process data. Detailed data protocol is explained in data protocol part of the document, and the hardware properties of IR system are focused in "IR Communication Module" part.

One pushbutton has been added to the system for activating the IR LED, which is the trigger of the gun. When user presses the button, the IR LED will start sending fixed length data package which is the shooting part of the game.

A sound indicator is connected to microcontroller in player kit and will "beep" when health or ammo decreases below some critical limit. This is important, because the user should be informed about her/his health and ammo condition. The sound indicator is attached to one of the digital outputs of the PIC18F2553. Information about pushbutton and sound indicator will be given in "User Interaction Hardware Modules" part.

The schematic and layout of the system is shown in figure 7 and figure 8. Connector explanations are given below figure 8.

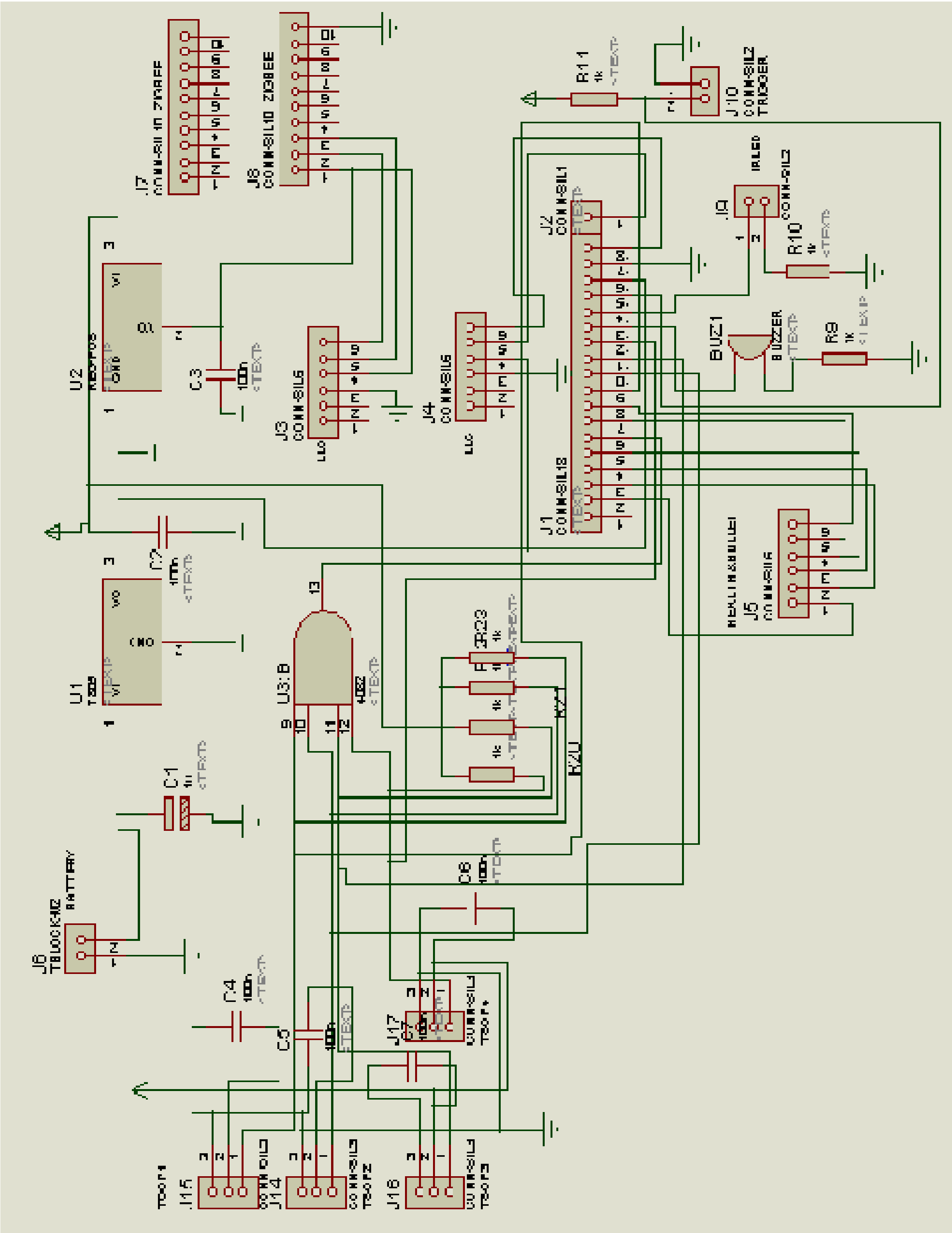

Figure 7 : Schematic of the System

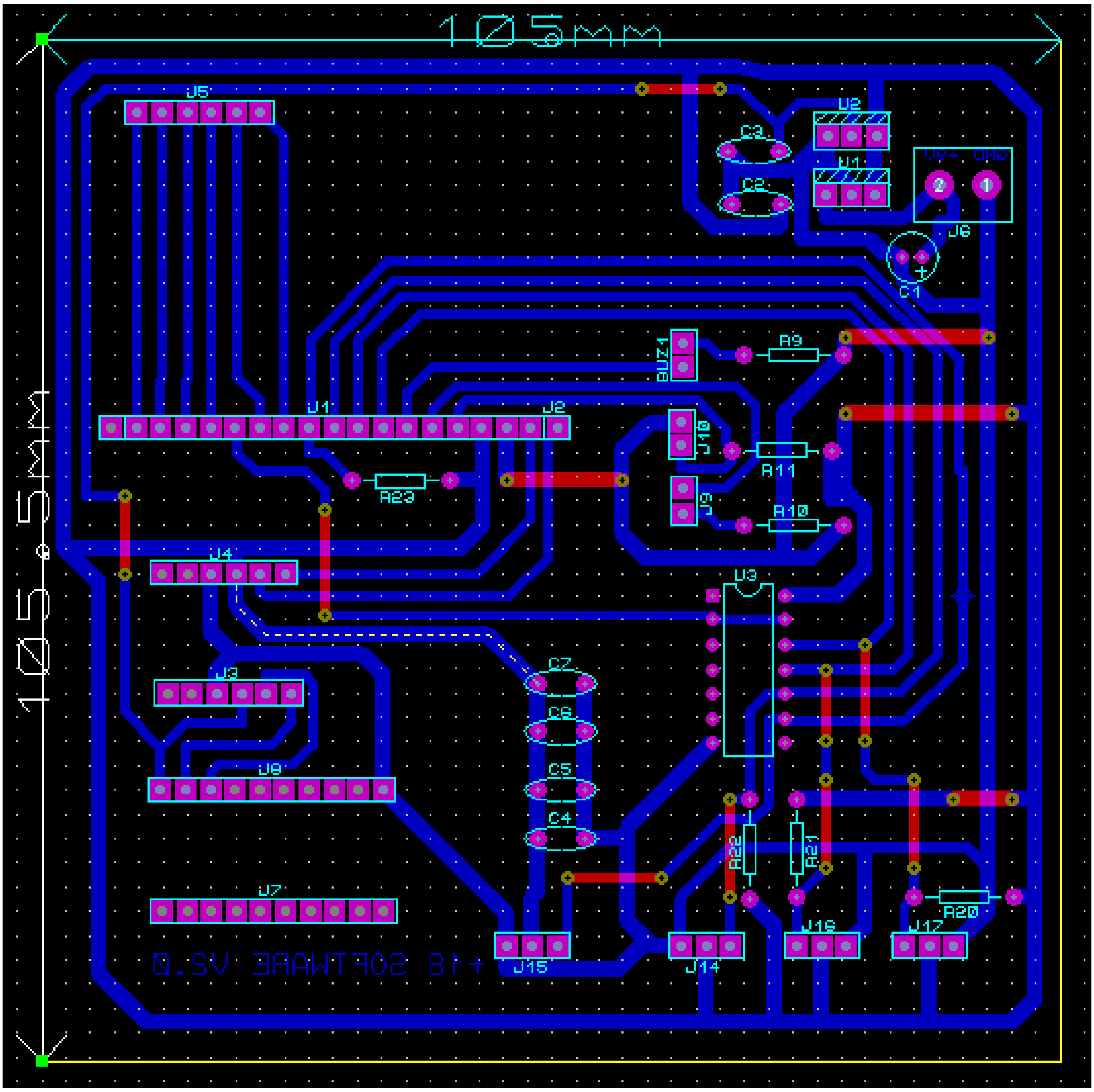

Figure 8: Layout of the System

Connector List:

Connector Name	Explanation
J1, J2	1 X 19 Pin connector for the USB Bit Whacker - 18F2553 PTH Kit
J3, J4	2 X 6 Pin connectors for the Logic Level Converter that will be used with XBee module(will be explainaned later).
J5	1 X 6 Pin connector for health and bullet indicator LCD (bonus indicator in bonuses)
J7, J8	2 X 10 Pin connectors for the XBee module
J14, J15, J16, J17	4 X 3 Pin connectors for the IR body sensors.
J9	1 X 2 connector for IR Transmitter (IR LED)
J10	1 X 2 connector for pushbutton(trigger)
BUZ1	1 X 2 connector for buzzer

Final lookout of the designed circuit is visualized in figure 9.

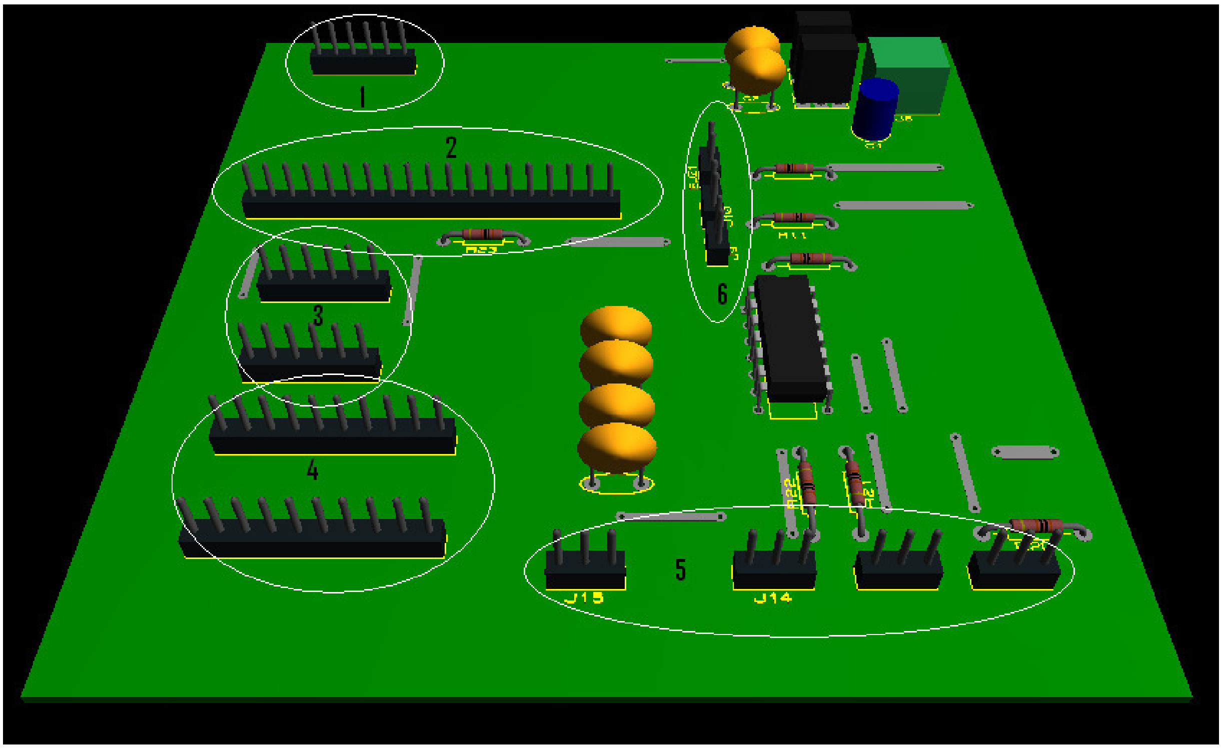

Figure 9: Final Lookout of the Microcontroller Module

The connector explanations of final lookout are given below.

1- 1 X 6 Pin connector for health and bullet indicator LCD (bonus indicator in bonuses)
2- 1 X 19 Pin connector for the USB Bit Whacker - 18F2553 PTH Kit
3- 2 X 6 Pin connectors for the Logic Level Converter that will be used with XBee module.
4- 2 X 10 Pin connectors for the XBee module
5- 4 X 3 Pin connectors for the IR body sensors.
6- 3 X 2 Pin connectors for push button (trigger), IR Transmitter (IR LED) and beeper.

4.2.1.2 *Infrared Communication Module*

There are 2 main components in Infrared communication module.

1. TSOP1138 38 kHz receiver.
2. TSALxxxx IR LED, 5VDC, 850-950 nm wave length.

TSOP1138 IR receiver, shown in figure 10, is connected to RA4 pin of PIC18F2553, as said before. This device receives data packages in 38 kHz frequency.

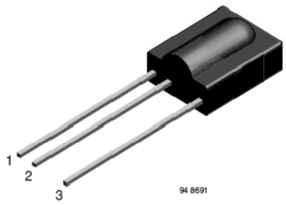

Figure 10: TSOP1138 [2]

TSAL1138 IR LED, shown in figure 11, is connected to RB2 pin of PIC18F2553 which emits IR bursts, fixed length data package, representing the bullets fired from the gun. IR receiver captures the bursts and sends to microcontroller via RA4 port to be handled. The event of sending and receiving fixed length data package is the responsibility of microcontroller software. This will be explained in software architecture part.

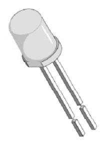

Figure 11: TSAL1138 [3]

The devices described above cannot be used alone need some complementary optics material in order to work properly.

First, IR LED has some limits. It cannot emit light as beams. Hence, some complementary material should be used to collect the light which is convex lens. The most proper convex lens and IR LED couple is found after some research. That is:

Infrared LED: TSAL 1138

LEDs beam angle: 20 Deg

Lens diameter: ~4.8 cm

Lens focal Length: ~10.8 cm

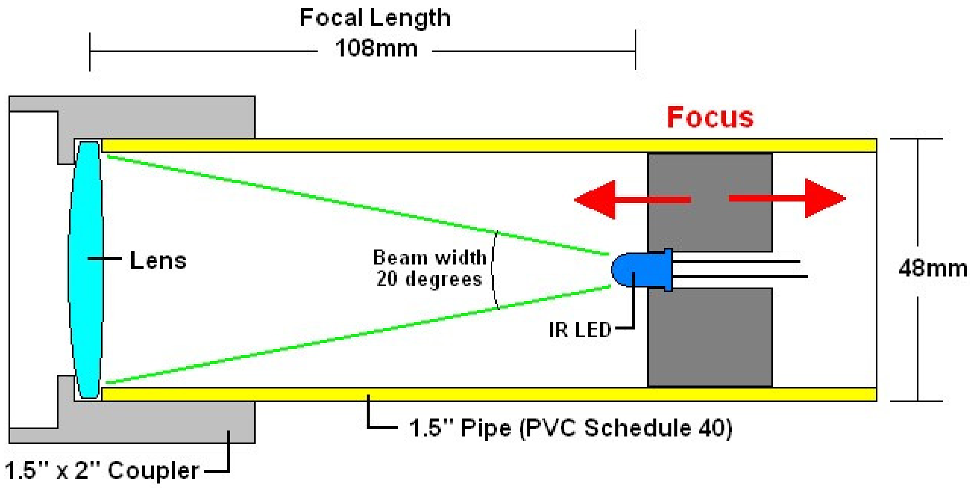

Figure 12: Usage of the convex Lens [4]

This lens will be in a gun-shaped material together with the IR LED. Also, in this gun-shaped material there will be a pushbutton, which will be used as a "trigger". This gun will have a look like a real pistol.

Secondly, there will be a complementary optic material for IR sensor. This optic material spreads straight light burst which will increase hit rate. A model for completed IR sensor is shown in the figure 13.

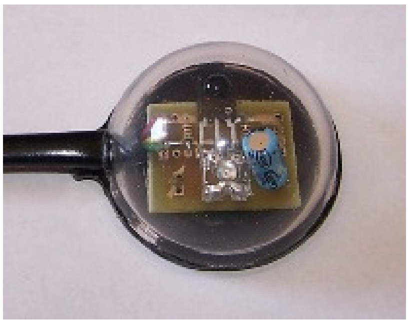

Figure 13: Representative photo of the sensor system [5]

4.2.1.3 RF Communication Module

As can be seen from the figure 5, the communication between PIC18F2553 and server is established by RF module. As RF communication device, we will use XBee Pro Series 2.5.

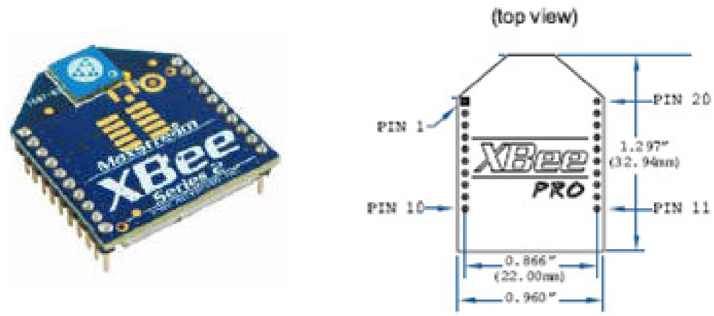

Figure 14: XBee PRO ZNet 2.5 Module [6]

This device can give the data to a host device through a logic-level asynchronous serial port. Through its serial port, the module can communicate with any logic and voltage compatible UART; or through a level translator to any serial device [6]. We are planning to make the data flow via serial communication ports. This would make the system easier to develop and debug. The communication logic of the device can be seen in the figure 15.

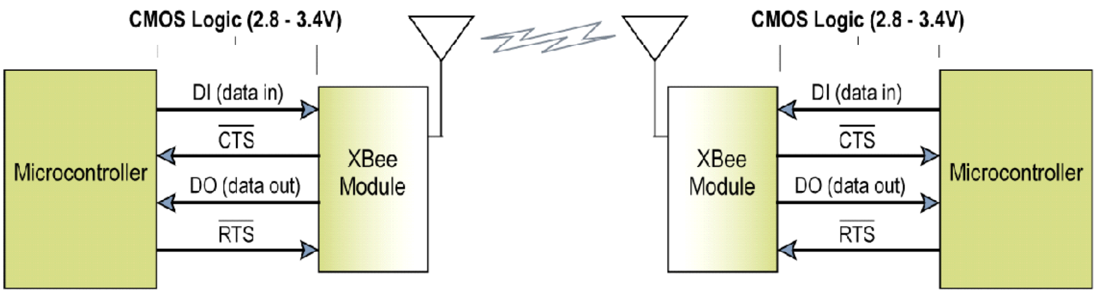

Figure 15: XBee Communication Diagram [7]

XBee is connected to microcontroller via TX/RX pins which is included in player and bonus kits. The data will be sent to server via TX pin and data reception from server is done via RX pin. But the signal voltages of PIC18F2553 and XBee device are not same (5V and 3.3). A complementary device for the interaction between microcontroller and XBee module (namely Logic Level Converter) has been added to the system. This device can be seen in figure 16.

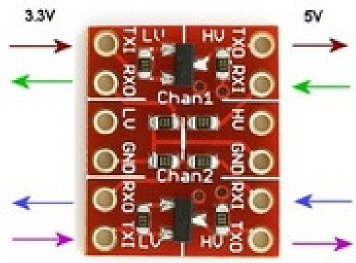

Figure 16 : Logic Level Converter [8]

Also this device has an internal antenna, so no additions are required. Further RF communication works will be done via software, which will be described later.

4.2.1.4 User Interaction Modules

The interaction between the hardware and the users, namely players, is in three parts:

1- Health and Bullet Indicator LCD (as Bonus indicator in bonuses)

2- Push button (trigger) in players' guns

3- Buzzer in player kits and bonuses

As health and bullet Indicator LCD, our system has a basic 16X2 character LCD (red on black). The representative picture of LCD is in figure 17.

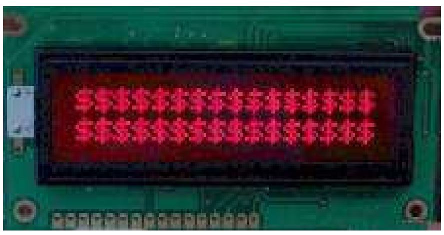

Figure 17 : Basic 16X2 character red on black LCD [9]

The LCD is on player's body (arm) and on the bonus. The information will be regularly updated and also it will show the bullet and health condition of players. On bonus devices, the LCD will give information of the status of the bonus (active/passive, type of bonus etc.).

As a trigger in players' guns, the system has a standard button. This button is connected to RB0 pin of PIC18F2553, and uses external interrupt.

The buzzer will be PC mount 12 mm 2.048 kHz device. It will be on players' bodies and in bonus devices. It will give different alarms in special conditions which are player get shot, player has critical health/bullet, bonus is activated etc. The device can be seen on figure 18.

Figure 18 : Buzzer [10]

4.2.1.5 Server Module

The server module is the "Coordinator" of the system. Basically, the server consists of a single PC and an RF module connected to it via USB port. Although we connect the pins of XBee directly to TX/RX pins of PIC18F2553, we need a complementary device for the communication between PC and XBee module, namely XBee Explorer USB. This device, which can be seen in figure 19, gives XBee module the chance to interact with PC easily. XBee module is placed on this board and with the USB port on the board we will connect it directly to PC. PC will see this device on a COM port, so the communication works will be done easily.

Figure 19 : XBee Explorer USB from SparkFun Electronics [11]

This PC will have the server side software running, and will send and receive the proper information. This information consists of the initialization of the devices, logging the game flow etc.

4.2.2 Software Architecture

We designed software in 3 main modules. These are Server Module, Player PIC Module, and bonus PIC module. Each module is responsible for sending data using RF and getting data from RF which provides the data transfer between the modules, also each module has its own internal operations. In the player and bonus PIC modules, there are also IR processes which are sending data and receiving data via IR. The functions of the parts of each module are explained in detail below separately.

4.2.2.1 Server Module

The functions that server module should have are listed with their explanations.

Sending Data With RF Module	
ActivateTheGame ()	Activates the game at the beginning of the game. Activation means initializing the guns with the bullets and giving every player health degrees.
ActivateBonus (bonusType, bonus_ID, interval)	Activates the bonus with the given type (which can be health bonus, bullet bonus, etc.) and bonus_ID. Activation will be on

	up to the given interval.
GiveBonusToPlayer (playerID, bonus_type)	Sends the bonus of the given type to the player who got it(with the given player_ID).
DeactivateTheGuns ()	Turns off the guns when the game is over.
ReactivateTheGuns ()	Reactivates the guns with its previous values.

Getting Data via RF Module	
WhoShotWhom (what_shot_id, who_shoot_id)	Analyzes whether the player shoots a bonus or a player. Calls GiveScore(who_shoot_ID) and Decrease_Health(what_shot_ID) method if a player is shot. Calls LockTheGuns() method if the player shoots a bonus and the type of the bonus is locking guns of the other team's guns, and GiveBonusToPlayer(who_shoot_ID, bonus_type) method to give the bonus to the player.

Internal Methods	
DescreaseHealth (player_ID)	Decreases the health degree of the player with the given player_ID.
GiveScore (player_ID)	Give points to the player with the given player_ID to be assigned in the score table.
PauseGame()	Pauses the game with calling DeactivateGuns() method.
EndGame()	Calls SaveGameData() and UpdateHighScores() methods to finalize the game.
ContinueGame()	After some pause, calls ReactivateGuns() method to continue the game from the point it was paused.

4.2.2.2 Player PIC Module

Sending Data With RF Module	
WhoShotMe (player_ID1,player_ID2)	Sends the ids of player that shoots(player_ID1) and player's own id(player_ID2). This method also calls DecreaseMyHealth() function to decrease health value.

Getting Data via RF Module	
GetRFDataFromServer (dataPackage)	Analyses the data package and calls the appropriate function. If the data is about bonuses , GetBonus(bonusType) function is called. If the data is about activation of the gun, it calls one of the functions; ActivateMyGun(), DeactivateMyGun(), ReactivateMyGun().

Internal Methods	
DescreaseMyHealth ()	Decreases the health degree of the player.
IncreaseMyHealth ()	Increases the health degree of the player.
DescreaseMyAmmo ()	Decreases the ammo of the player.
IncreaseMyAmmo ()	Increases the ammo of the player.
LockMyGun()	Locks the gun of the player for a predefined time period.
GetBonus(bonusType)	Calls the appropriate function according to bonusType which are IncreaseMyHealth(), IncreaseMyAmmo(), LockMyGun().
ActivateMyGun ()	Activates the player's gun when the game is started.
DeactivateMyGun ()	Turns off the player's gun.
ReactivateMyGun ()	Reactivates the player's gun with its previous values.
NotifyMe()	Notifys the player when she/he is dead or out of ammo.

Getting Data Via IR	Data reception will be implemented with interrupts on RA4 pin. Using the interrupt, the data package will be caught and analyzed so that WhoShotMe() function is called with the ids of the players that shoots and is shot.
Sending Data Via IR	Data will be sen via RB2 pin. When the player shoots, the player's own id will be sent. Also DecreaseMyAmmo() function will be called.

4.2.2.3 Bonus PIC Module

Sending Data With RF Module	
WhoShotMe (player_ID,bonus_ID)	Sends the id of player that shoots(player_ID) and bonus_ID to server. This method also calls DeactivateMe() function.

Getting Data With RF Module	
ActivateMe(bonusType,interval)	Activates itself with the given bonus type , and also sets its activation inteval.

Internal Methods	
DeactivateMe()	Deactivates itself when a player shoots it.
Getting Data Via IR	Data reception will be implemented with interrupts on RA4 pin. Using the interrupt, the data package will be caught and analyzed so that WhoShotMe() function is called with the ids of the players that shoots and is shot.

5 MODELING

5.1 Functional Modeling

5.1.1 Data Flow Diagrams

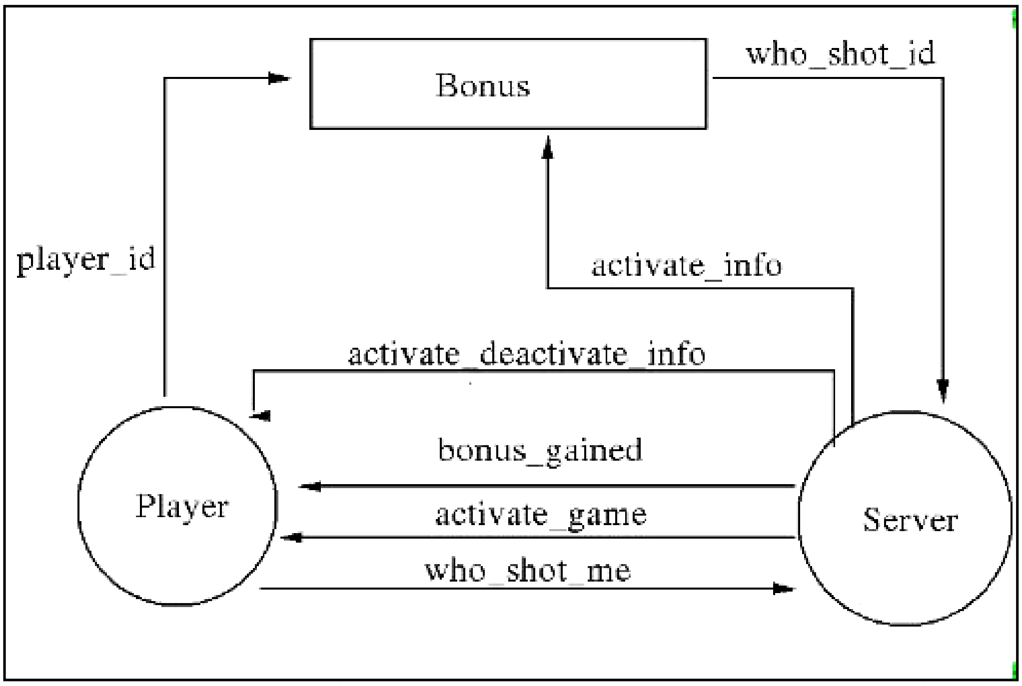

Figure 20: Level 0

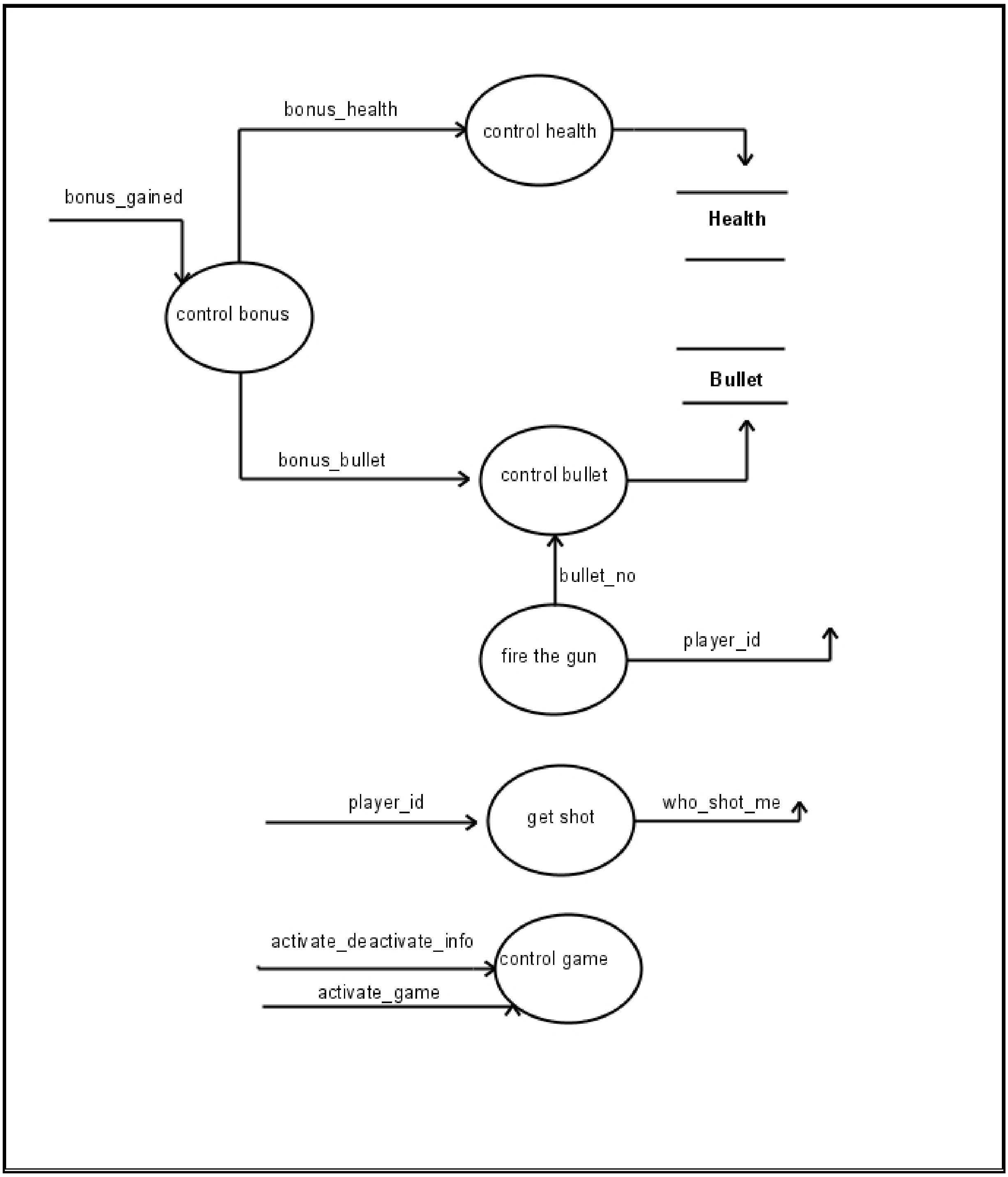

Figure 21: Level 1 Player

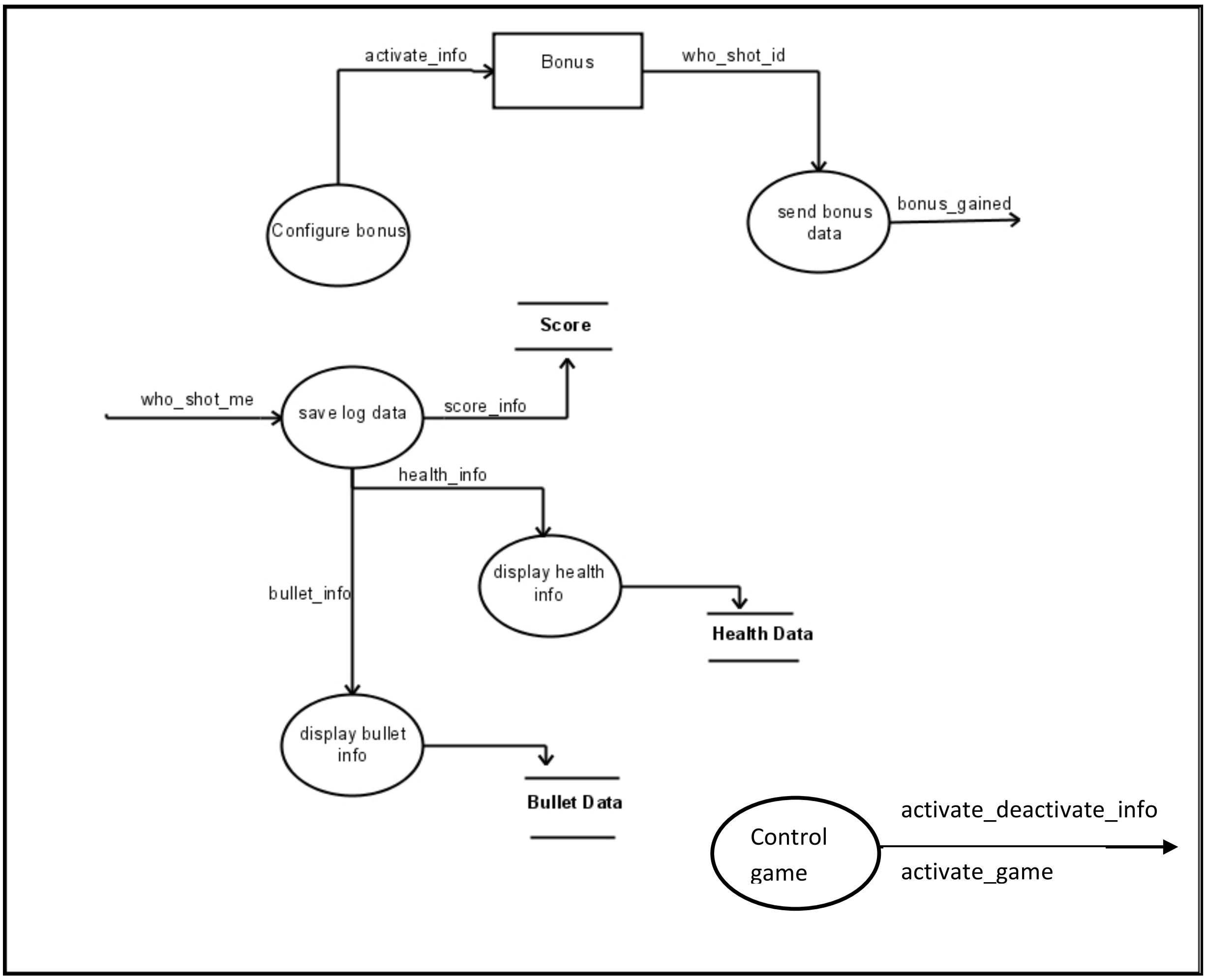

Figure 22: Level1 PC

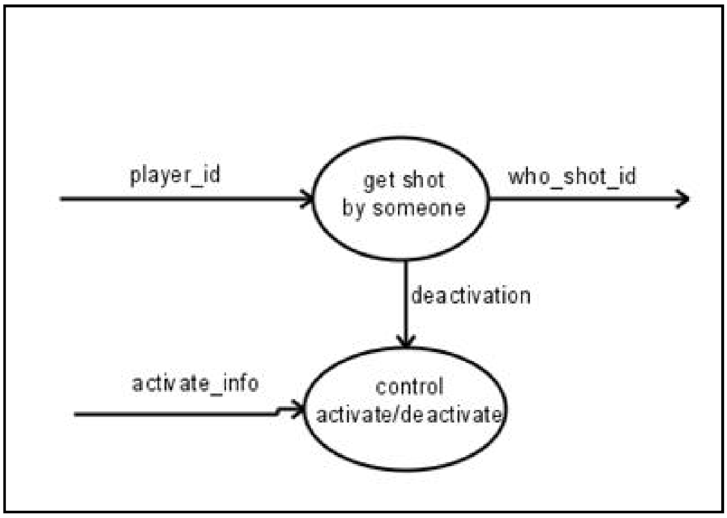

Figure 23: Level 1 Bonus

5.1.2 Data Dictionary

Name	Bonus pack
Where used?	Level 0 – External Entity
Description	The components which will be placed at somewhere in play area and activated by Server. It includes IR receiver and RF kit.

Name	activate_info
Where used?	Level 0 , Level 1-Server Output from ‘configure bonus’, Input to ‘Bonus pack ‘
Description	Server sends some information using the RF kit to make the bonus package activated at some time of the game. Also the type of the bonus is in the activate_info.

Name	player_id
Where used?	Level 0 , Level 1-Player Output from ‘fire gun’, Input to ‘get shot’
Description	Every time the player fires the gun, the gun is sending the id of the player.

Name	who_shot_id
Where used?	Level 0 , Level 1-Server Output from ‘Bonus pack’, Input to ‘send bonus data’
Description	When the bonus package is shot by some player, the IR receiver captures the player_id of the shooter, which is the who_shot_id. This id then goes to Server to understand who gained the bonus.

Name	bonus_gained
Where used?	Level 0 , Level 1-Player, Level 1-Server Output from ‘send bonus data’, Input to ‘control bonus’
Description	Server the bonus information (its type) to the player who gained it.

Name	who_shot_me
Where used?	Level 0 , Level 1-Player, Level 1-Server Output from ‘get shot’, Input to ‘save log data’
Description	This id will be sent by the player kit to the Server when the player is shot by someone to understand who shot the player.

Name	Health Data
Where used?	Level 1-Server Data Storage
Description	Health data of the players which will be stored in the server.

Name	Bullet Data
Where used?	Level 1-Server Data Storage
Description	Bullet data of the players which will be stored in the server.

Name	Score Data
Where used?	Level 1-Server , Data Storage
Description	Scores of the players which will be stored in the server.

Name	Health
Where used?	Level 1-Player , Data Storage
Description	Health degree of the player stored in player kit (in PIC).

Name	Bullet
Where used?	Level 1-Player , Data Storage
Description	Bullet number of the player stored in player kit (in PIC).

Name	bonus_health
Where used?	Level 1-Player Output from ‘control bonus’, Input to ‘control health’
Description	The info to give the bonus to the player when the player gained the health bonus.

Name	bonus_bullet
Where used?	Level 1-Player Output from ‘control bonus’, Input to ‘control bullet’
Description	The info to give the bonus to the player when the player gained the bullet bonus.

Name	bullet_no
Where used?	Level 1-Player Output from ‘fire gun’, Input to ‘control bullet’
Description	When the player fires the gun, the bullet number in PIC will be controlled by the player kit and stored bullet number will be recalculated.

Name	deactivation
Where used?	Level 1-Bonus Output from ‘get shot by someone’, Input to ‘control activate/deactivate’
Description	When bonus is shot by some player, it sends a deactivation message to control activate/deactivate in order to be deactivated.

5.2 Behavioral Modeling

5.2.1 State Diagrams

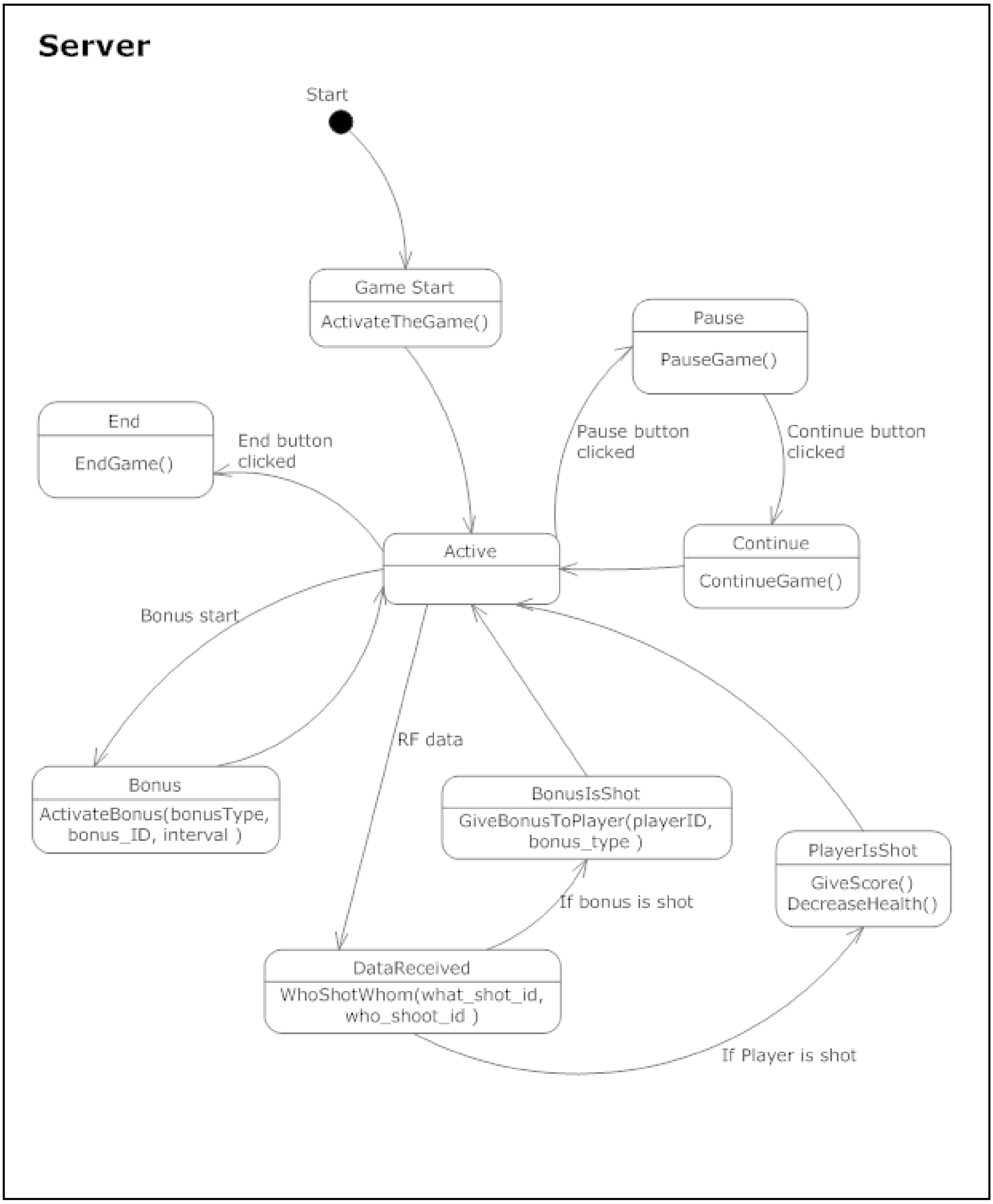

Figure 24: Server State Diagram

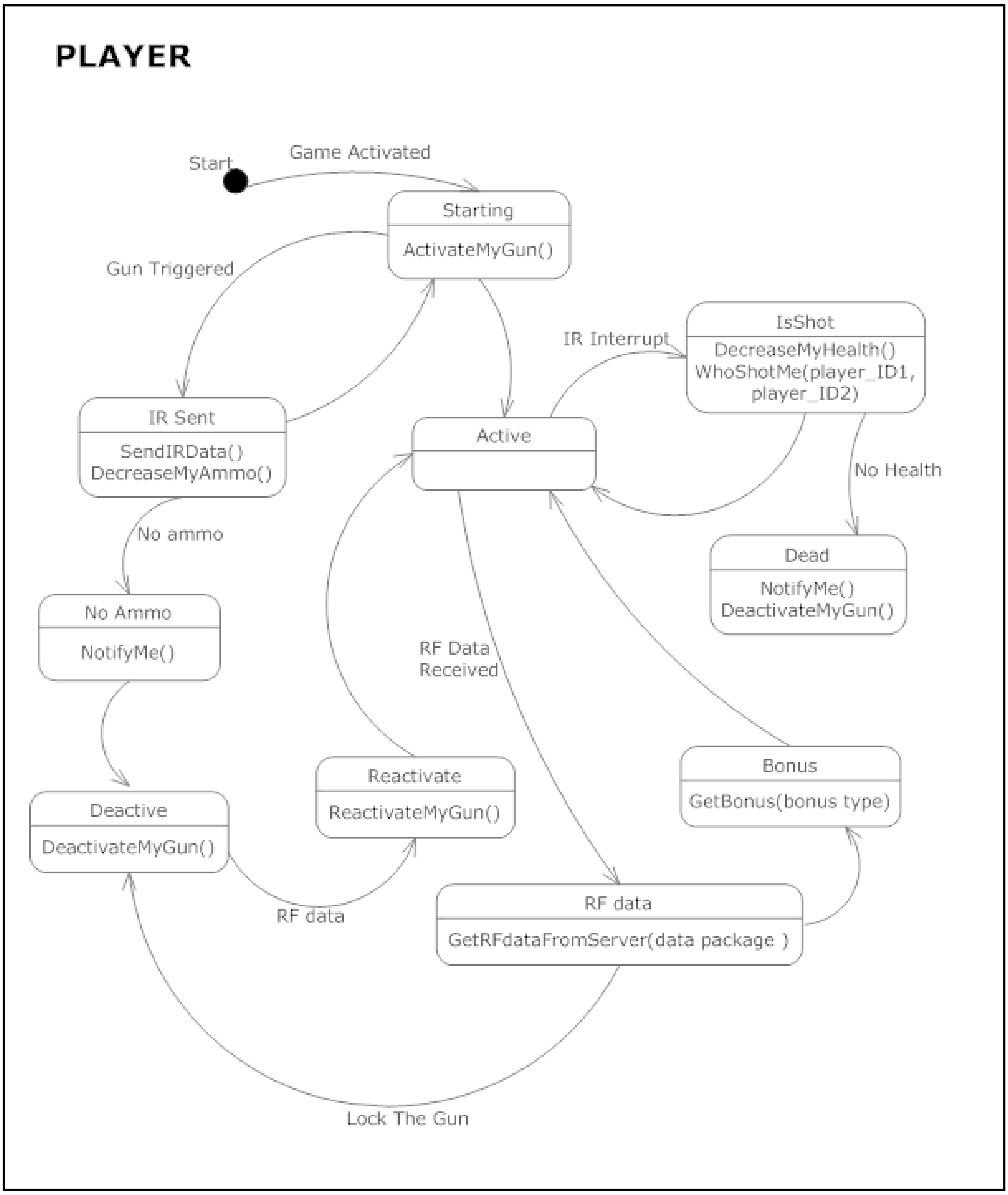

Figure 25: Player State Diagram

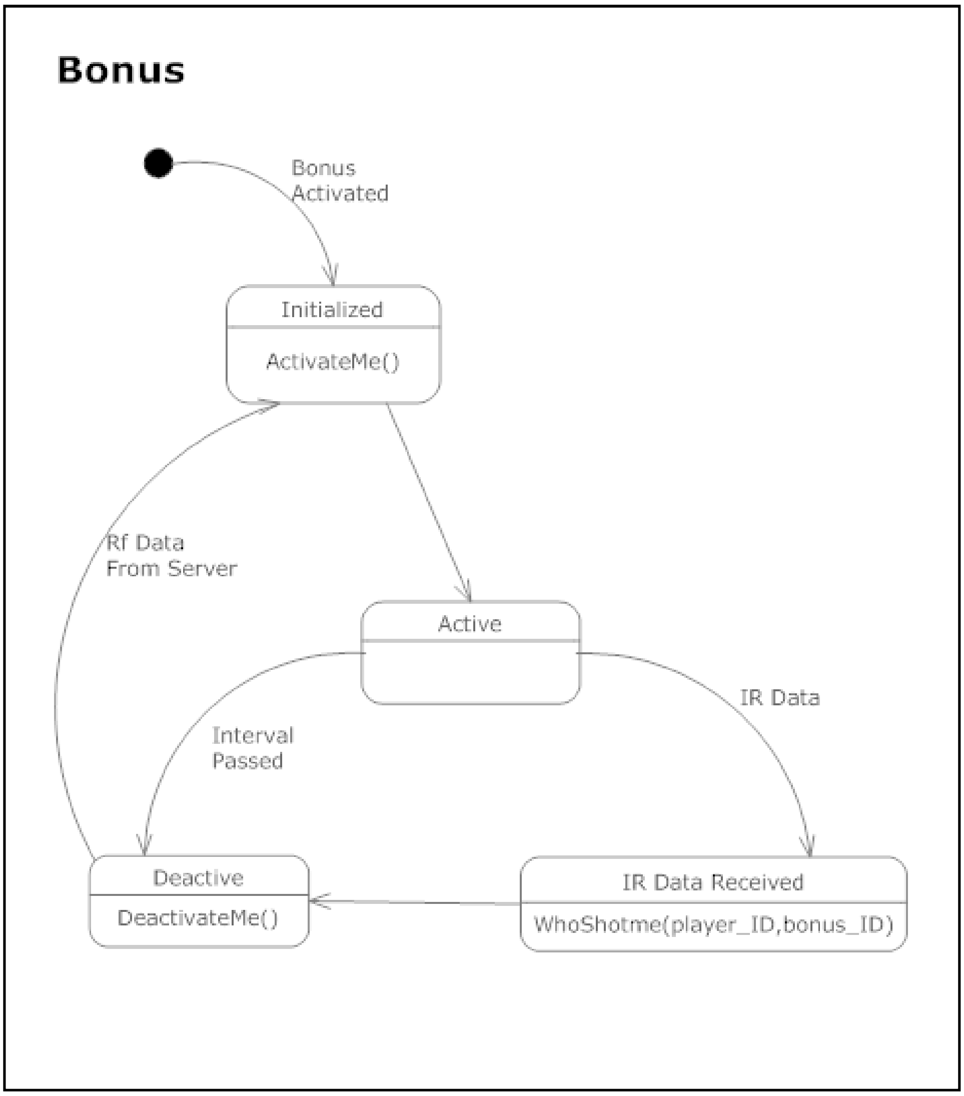

Figure 26: Bonus State Diagram

5.3 Data Modeling

In this section, the data protocol used in infrared communication will be explained.

5.3.1 IR Data Protocol

In our project, we need an IR data protocol in order to send and receive data packages in an accurate and secure way. When a shot is fired, IR data which is unique for each gun is sent to the target sensor. The data transmitted is composed of 5 bits which describes player ids. These bits are represented as signals which are series of ups (5V) and downs (0V). Before these 5 bits, we need a preamble signal to understand the arrival of a data package at the receiver side.

At first, it is important to check the data to process it correctly. We must check signals coming from the IR transmitter at each millisecond and decide whether it is a bit value of 1 or 0. We designed two deterministic finite automata (DFA), one for preamble signal and one for data package, to figure out which part of the signal being processed at that moment and what we should expect in the next step.

In figure 27, the DFA for preamble signal is shown. As can be seen from the figure, we use 3ms up, 6ms down and 3ms up as a preamble signal format. S1 is the starting state which is waiting for the preamble and S2, the ending state, is reached when the reception of preamble is finished. After this preamble signal is captured, receiving of the 5 bits for the player id will be started.

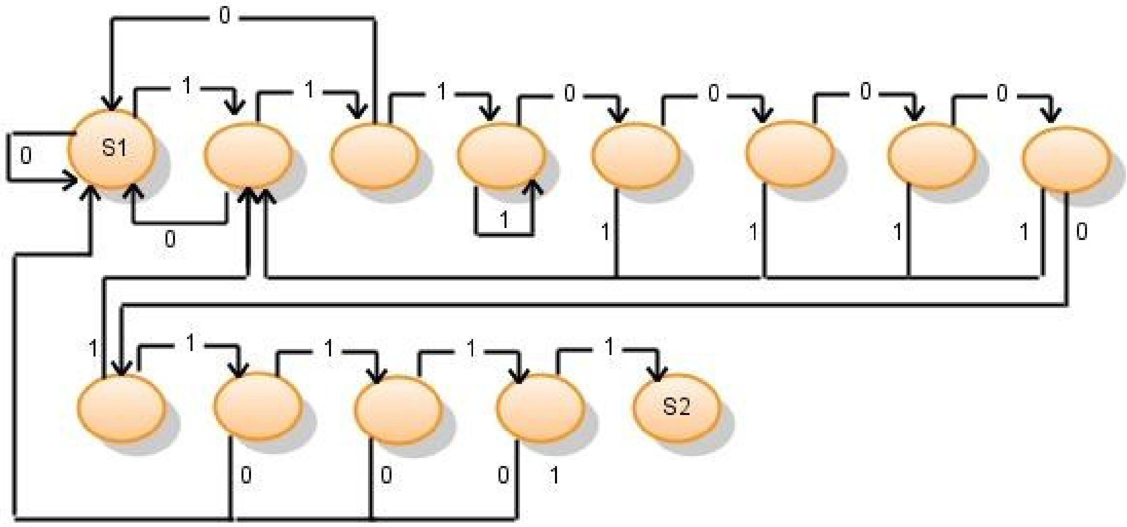

Figure 27: DFA for preamble signal

In figure 28, the DFA for data bits is shown. In our protocol design, a bit value of "1" and "0" are represented by a pulse (38 KHz carrier ON) with a period of **2ms** and 1ms respectively. To separate a bit value from others, we decided to put 2ms down signal before each bit values. The state transitions are as seen in the figure. Being at the state 0 and receiving a down signal means that the bit value is "0". If an up signal is received than the state goes to state 1 and expects a down signal for a bit value "1". Capturing 5 valid data bits means that it is a successful reception of the player id and the microcontroller goes to S1 state and starts to wait for a preamble signal again.

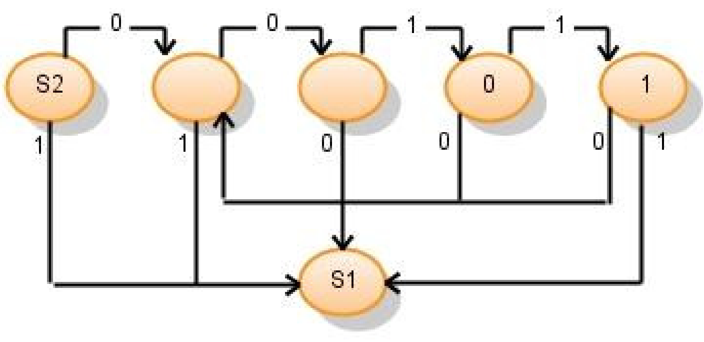

Figure 28: DFA for data bits

6 GUI DESIGN

We designed a GUI for initializing the game, choosing the mode of the game and deciding other options and specifications. This GUI will be used by the person who is responsible for managing the game at the server side.

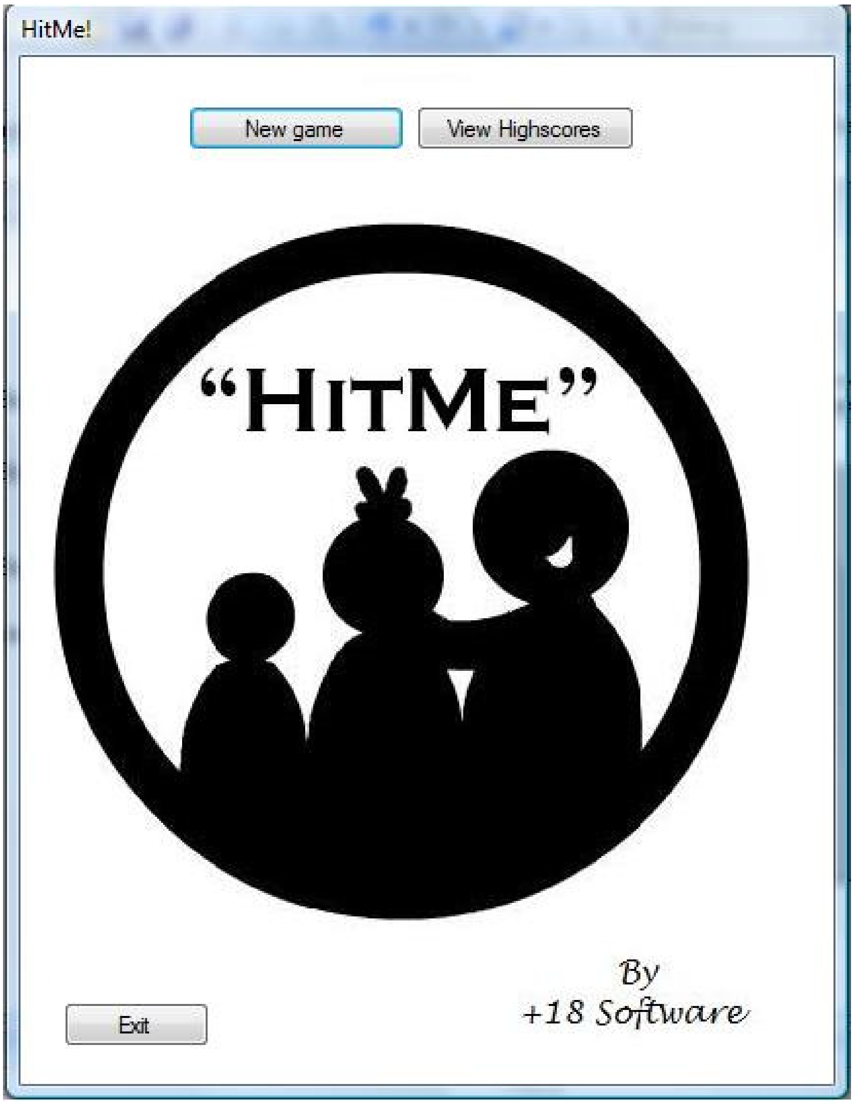

Figure 29: Hitme Main Window

The main screen of our game can be seen above in figure 22. The user can see the high scores by clicking "View Highscores" button. The scores of the people will be sorted in our XML file. We have chosen to use XML instead of database, because only the high scores and player names are stored. This XML file will be updated after every game and high score table will be holding up to 100 players. When the user presses the “Exit” button the program terminates. Clicking “New game” button will direct the user to the “Game Mode Selection” window, figure 23.

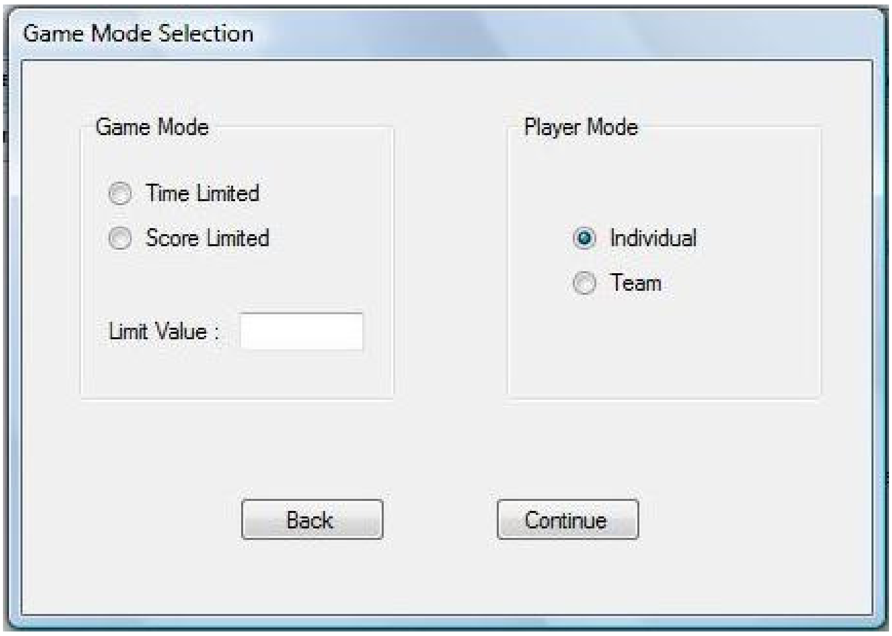

Figure 30: Game Mode Selection Window

In this window user decides the mode of the game which can be either “Time Limited” or “Score Limited”. In each case the user will enter the Limit Value. The game will be ended when this limit value is reached during the game. Besides, user should select either “Team Mode” or “Individual Mode”. User should click “Continue” button after selecting the Mode.

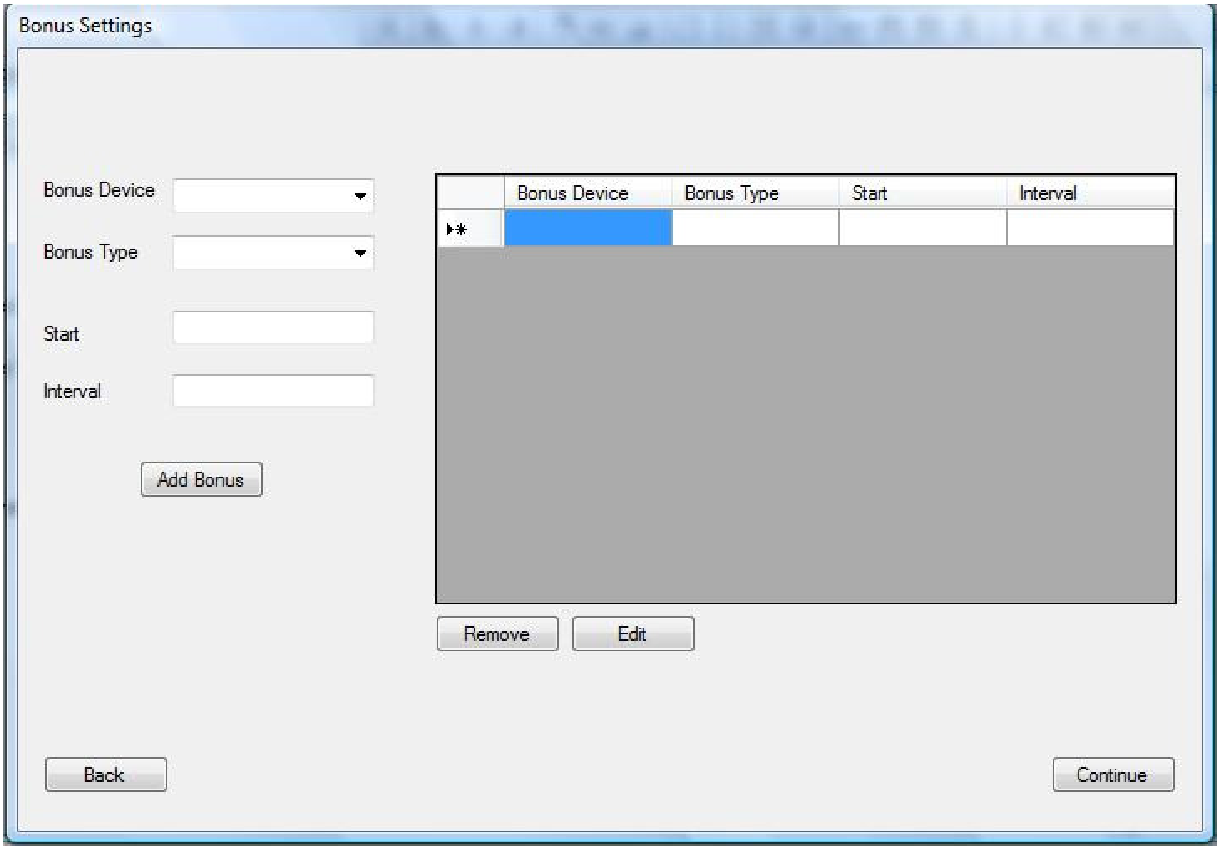

Figure 31: Bonus Settings Window

Before starting the game, the user must adjust the settings of the bonuses. In the game play area there will be some bonus devices located at different places. All of these bonus devices have a unique id and user can select one of these by "Bonus Device" combo box. User must select the type of the bonus of the selected device by "Bonus Type" combo box. These bonuses have two attributes which are start and interval. The start attribute decides the activation time of the bonus and this bonus will appear active within interval time which is typed by the user in "Interval" text label. It is possible to remove and edit the data entry selecting the row and then clicking one of these buttons. Clicking "Continue" button directs the user to the "Player Settings" window.

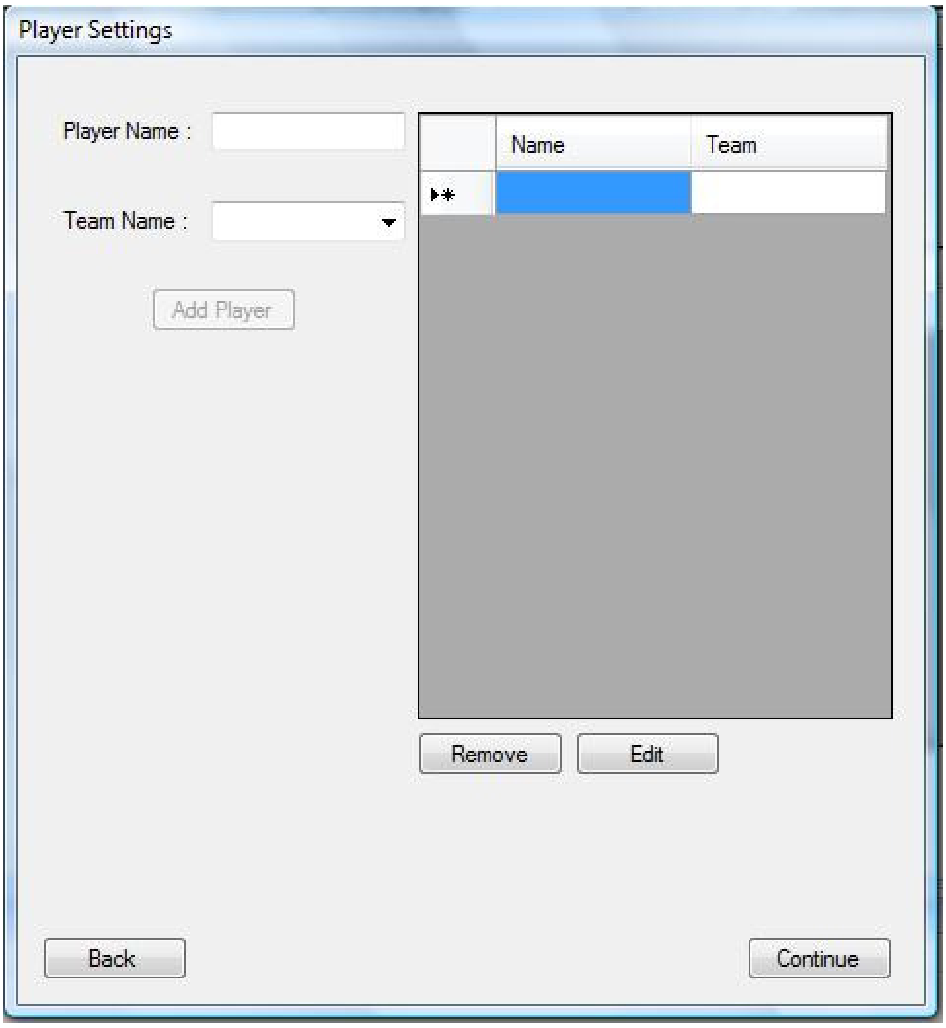

Figure 32: Player Settings Window

In this window user will create players by entering their names and selecting their teams. If Game mode is decided as “Individual mode” in the “Game Mode Selection” window, “Team Name” combo box will be locked and cannot be used by user. On the other hand, if “Team Mode” is selected, user should select a team name from predefined team names. After these steps, user should click “Add Player” button to add created player to the table. User can remove or edit data entries by selecting the related entry and clicking “Remove” or “Edit” buttons. Clicking “Continue” button directs the user to the “Now the Hitme Time!” window.

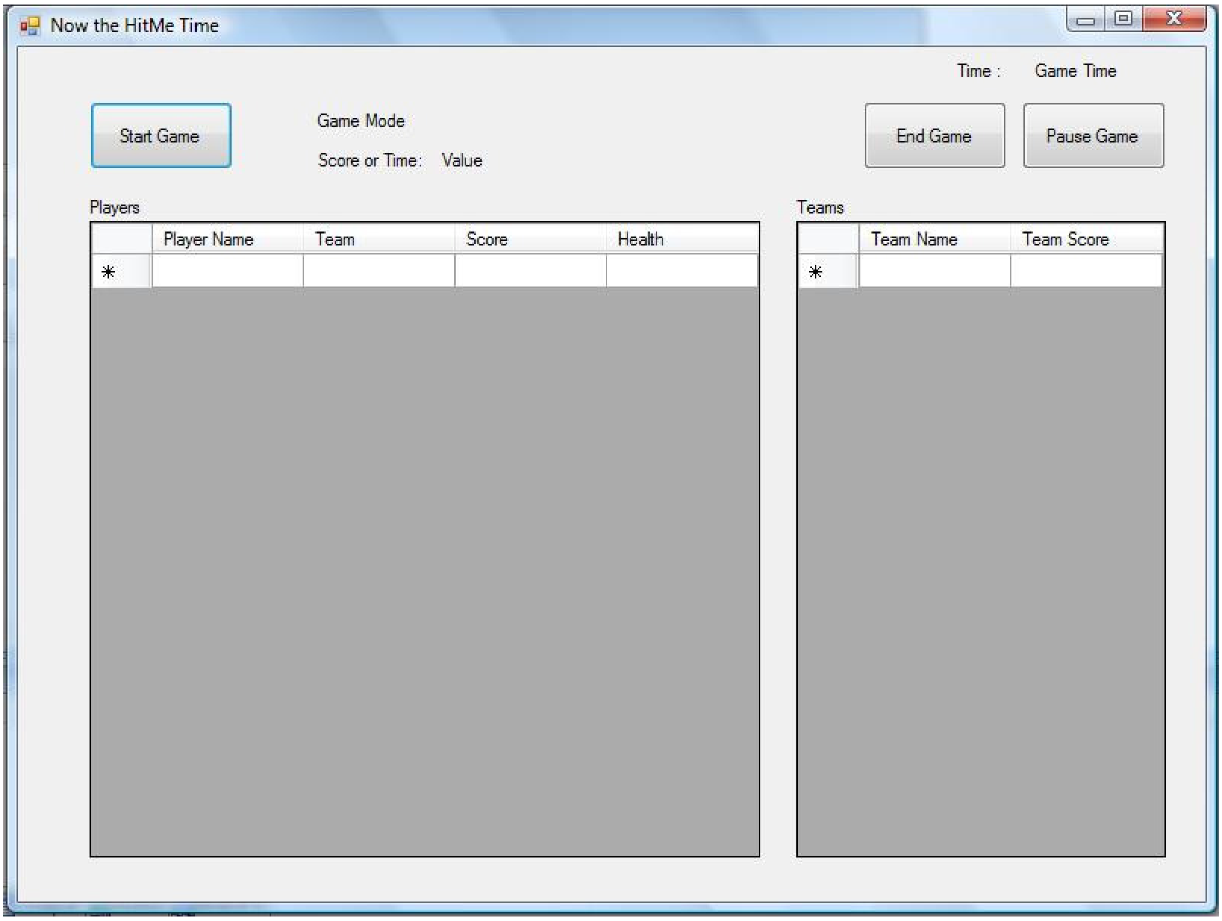

Figure 33: Hitme Game Play Screen

This window is displayed during the game. User can see player names, their team names, score values and health values in “Players” data table. If the game is in “Team Mode”, user can see team names and team scores in “Teams” data table. User can also see the limited value right side of the “Start Game” button. Time of the game can be seen at top right corner. When players are ready to play, user clicks the “Start Game” button to start the game. In any time user is able to end or pause the game by clicking “End Game” button or “Pause Game”. When the user clicks the “End Game” button, all of the game data will be saved to an XML file.

7 TESTING METHODOLOGY

Basically, our system has 4 main parts (figure: Module view of the system). Because of our design methodologies, testing will take place after each part has been developed. Every code segment will be tested just in time (we could not do the tests by test frameworks, they will be manually and semi-automated). In addition to that, some systematical tests for communication parts will be performed.

Those are:

- RF communication testing
- IR communication testing

7.1 RF Communication Testing

The RF module that we will use is an easy-to-use and practical device. But in addition to that, using it with a newly designed microcontroller system has some risks. Together with untested software, it could be a big problem. To avoid this, we are planning to perform two dimensional testing on this part. One dimension is the data length and the other dimension is the distance between receiver and transmitter.

Some test codes (functions) will be implemented for these operations. Arranging the arguments of these test functions will give us the ability of extreme testing.

Also, we will see that if our results agree with vendor’s specifications or not. This is a good way of measuring the efficiency.

7.2 IR Communication Testing

IR communication is the most detailed and difficult part in our system. Some complementary material (lens, completed optics module.) will be used with the hardware. Testing of IR communication will take much time than developing it.

In addition to software testing, there will be physical tests. The efficiency of optics module will be tested by laboratory experiments. We will be able to make this test by the help of Department of Electrical and Electronics Engineering. The optics laboratory of that department will be in our use (by the help of Assistant Professor Behzat Şahin).

Other tests (software and hardware) will be two dimensional like RF tests. One dimension is data length and the other dimension is distance between transmitter and receiver. Test functions will be written and automated tests will be performed.

8 DEVELOPMENT SCHEDULE

8.1 Completed Parts

By the early days of this semester, we have started to make researches about this project. Due to time restrictions in this project, we have an early start to development. The completed parts of development process are described under 3 main titles:

- Hardware & Software Design
- Infrared Communication and Data Protocol
- Complementary Material Design and Prototype

8.1.1 Hardware & Software Design

As a development board, we assembled USB bit whacker to place PIC18F2553 on it, logic level converter and XBee module onto copper plate. Also, we have completed the pin connections between them. Assembled circuit can be observed from figure 34.

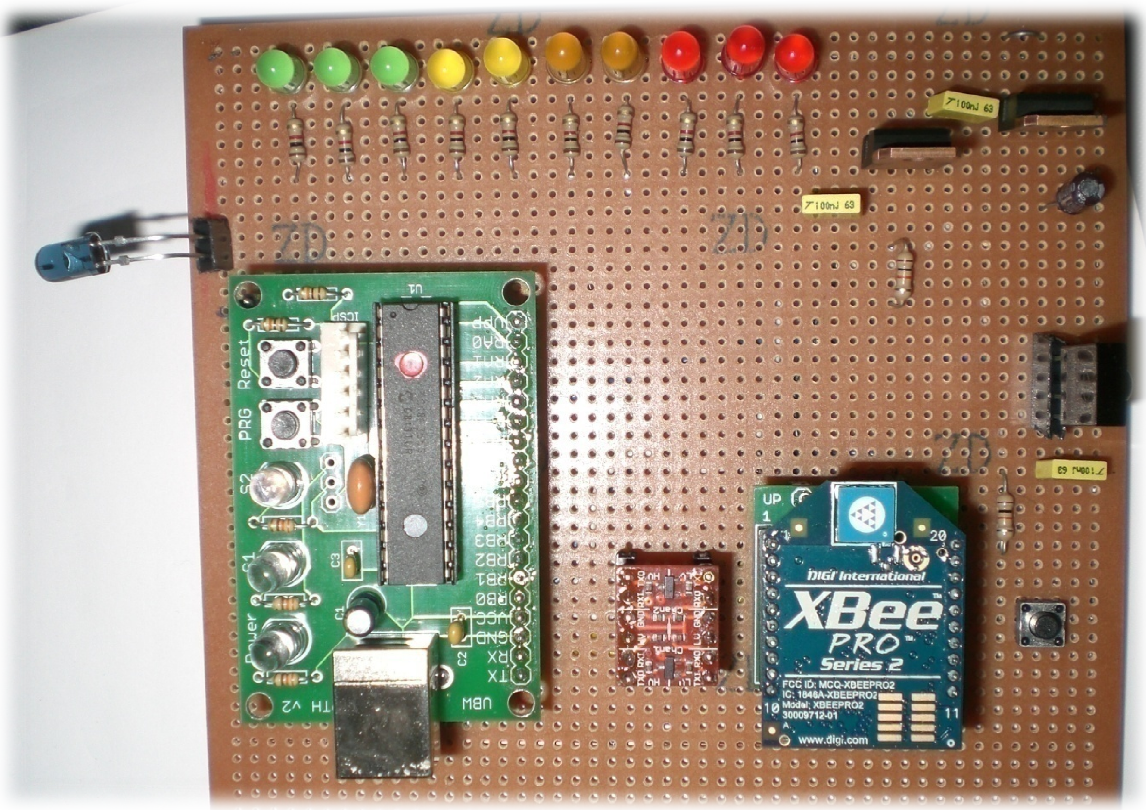

Figure34: Assembled circuit

8.1.2 Infrared Communication and Data Protocol

The usage of IR LED and sensor has been tested by the help of oscillator. Simple test code is written for the test of data transfer via IR. This simple code has proven the success of the hardware. Besides, data protocol for infrared communication is decided and ready to implement.

8.1.3 Complementary Material Design and Prototype

The most ambiguous part in our project, optics module has been developed. Also we have made a basic prototype.

This prototype consists of three parts: a carton made tube, a convex lens and a carton made back cover. The tube has a length equal to the focal length of the convex lens. And the convex lens fits perfectly to the front of the tube.

We have placed an IR LED to the back of the tube (to the center of the cross section of the tube), and given constant voltage to IR LED. With the collective effect of the convex lens, the spread light of the IR LED has turned into a straight beam.

Completion of this prototype has been a milestone in our project. The straight beam would let the IR light to reach to receiver's sensor properly.

8.2 Future Work

At this point the first thing to do is implementing an IR protocol and RF protocol which is the milestone of this project. We plan to finish it till demonstration day, which is end of 08/09 fall semester. When we implement protocols, players will able to shoot each other and send this data packages to server module via RF device. After this step, we will develop our IR guns with the optics module to send data linearly. When we completed all of these, we will develop graphical user interface at the server module to process all of the data send by the bonuses and players.

If all the modules finish before the time we decide, we will design special guns and armors for players to make the game more attractive.

8.3 Gantt chart

Figure 35: Gantt Chart

8.3.1 Task and Person List

1. IR communication (software&hardware): Çağrı – Gamze
2. RF communication(software&hardware) : Zeynep – Onur
3. User interaction hardware : Çağrı -Gamze
4. GUI design : Onur
5. Circuit implementation : Çağrı
6. Complementary materials : Onur – Zeynep
7. Main microcontroller software : Gamze - Zeynep

9 REFERENCES

1. http://www.sparkfun.com/commerce/product_info.php?products_id=8265
2. TSOP1138 ,Datasheet for IR Receiver Modules for Remote Control Systems, Vishay semiconductor Co.
3. TSALxxxx ,Datasheet for GaAs/GaAlAs IR Emitting , Vishay semiconductor Co.
4. http://www.lasertagparts.com/mtoptics.htm
5. http://www.lasertagparts.com/mtsensors.htm
6. http://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-2.5-Datasheet.pdf
7. http://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-2.5-Manual.pdf
8. http://www.sparkfun.com/commerce/product_info.php?products_id=8745
9. http://www.sparkfun.com/commerce/product_info.php?products_id=791
10. http://www.sparkfun.com/commerce/product_info.php?products_id=7950
11. http://www.sparkfun.com/commerce/product_info.php?products_id=8687