Lucent Tag

Project Description

Riley Culligan

Fall 2013

1. Table of Contents

1. Functional Description

2.1. Introduction:

Laser tag was first designed by the US military, but became commercially popular in the 1980s. That being said, laser tag systems are not a new idea, but it has a chance now more than ever to innovate a once popular game. Lucent Tag gives the user both the instrument to play the game and the module to control the game. The control module, which will be designated as the "ref module", is not necessary to host a game, but it will track data and make the process easier and fair for the players. The instrument, which will be designated as the "gun module", is shaped and operates like a gun. Each gun module has several "receiver nodes", which are sensor nodes that each player places on their body which designate places they can be shot.
Project Hardware and Applications

2.2. Project Hardware:

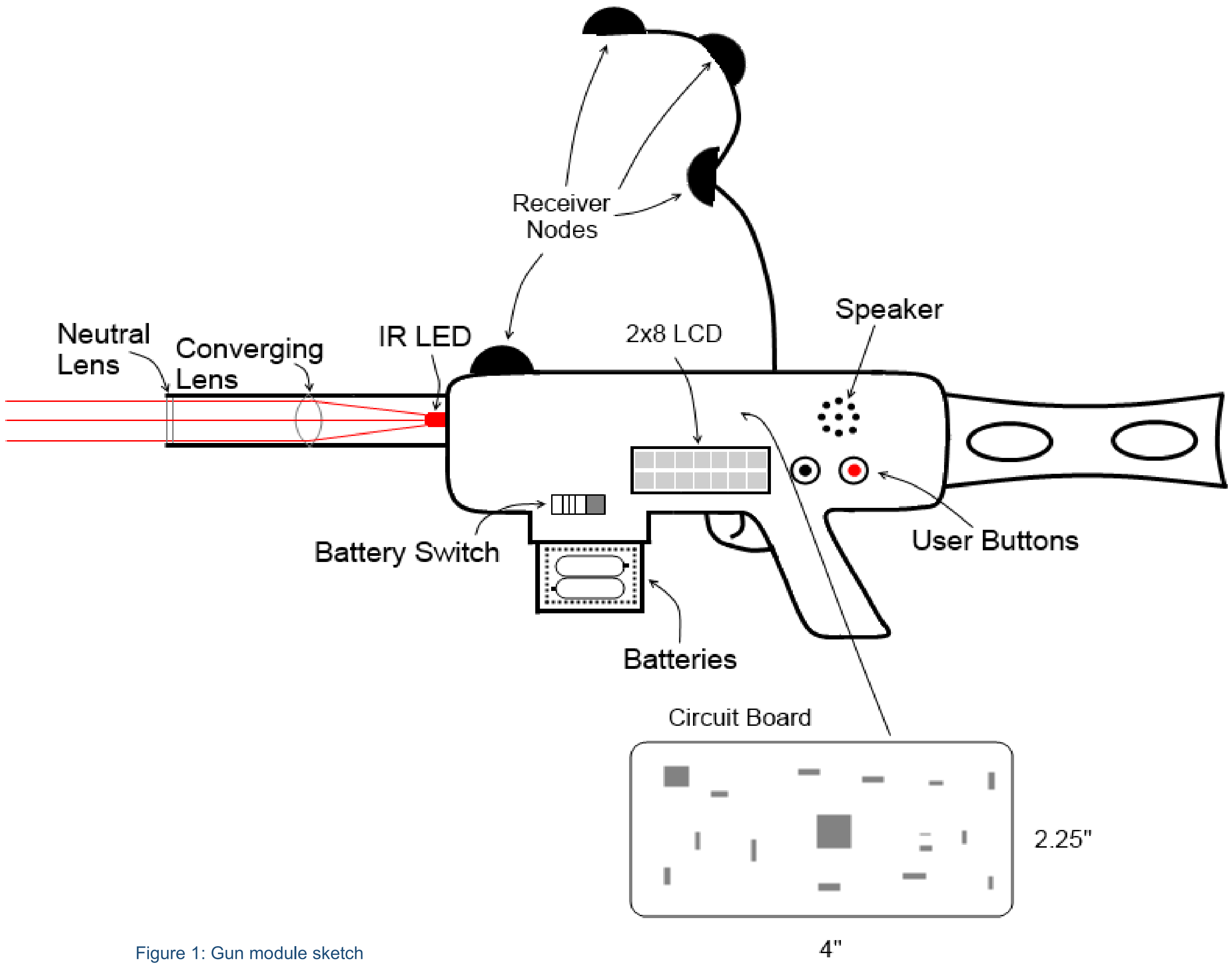


Figure 1: Gun module sketch

2.2.1. Gun Module

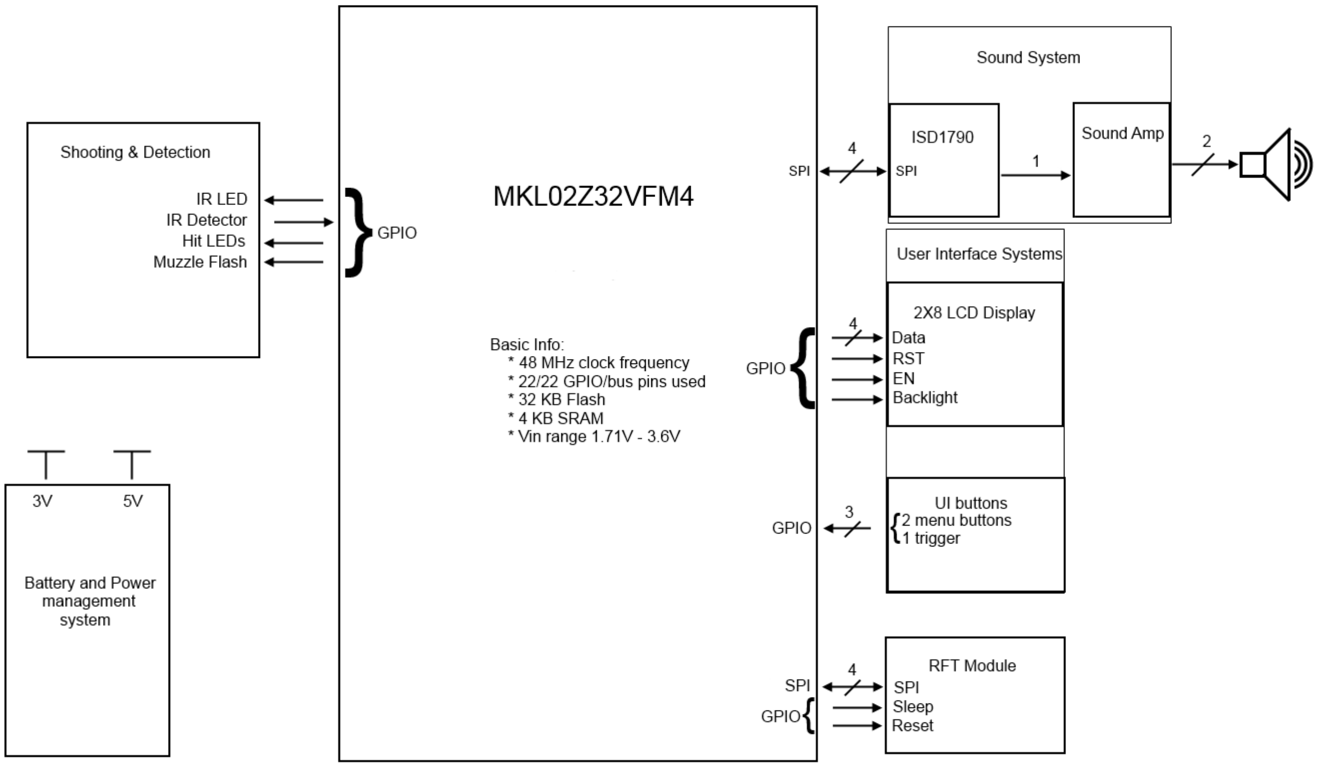


Figure 2: Gun module block diagram

Refer to figure 2 for all gun module blocks.

The MCU block contains all the necessary hardware and peripherals for the microcontroller, the MKL02Z32VFM4. The details of the MCU are described in detail in the hardware description section.

The Shooting & Detection block handles all the sending and receiving of IR messages. The IR transmitter circuit is constructed of a high power IR LED, a resistor, and a converging lens for focusing the light. The LED shines through the converging lens to focus the light into parallel beams to give the gun a longer range than without the lens. The IR receiver circuit is contained within several nodes external to the gun chassis to be placed in several placed around the user. Each node will have a couple IR detectors and a hit LED. The IR detectors are placed in parallel to detect the IR signal from several locations. If a signal was successfully received, then the hit LEDs give visual feedback to all other players that they have been hit.

The sound system block is comprised of a sound card ship, an audio amplifier, and a speaker. The sound card chip will be most likely an ISD1790 or similar. The sound card chip can record sound, store it, and play it back with a lot of functionality. The sound played through it will be amplified and played through the speaker to provide auditory feedback to the players.

The RFT block provides communication between the ref module and the gun modules over a personal area network. The IC used for this task will likely use the SPI bus with a couple additional GPIO ports. This IC will likely be the AT86RF231-ZURCT-ND.

The user interface block composes of 3 buttons and a 2x8 LCD screen. 2 of the buttons will be used simply as buttons, and will be placed close to the LCD screen, while the other button is attached to the trigger of the gun. The LCD screen will always be displaying relevant information, while the backlight will automatically turn on and off when the user needs it to conserve power.

2.2.2. Ref Module

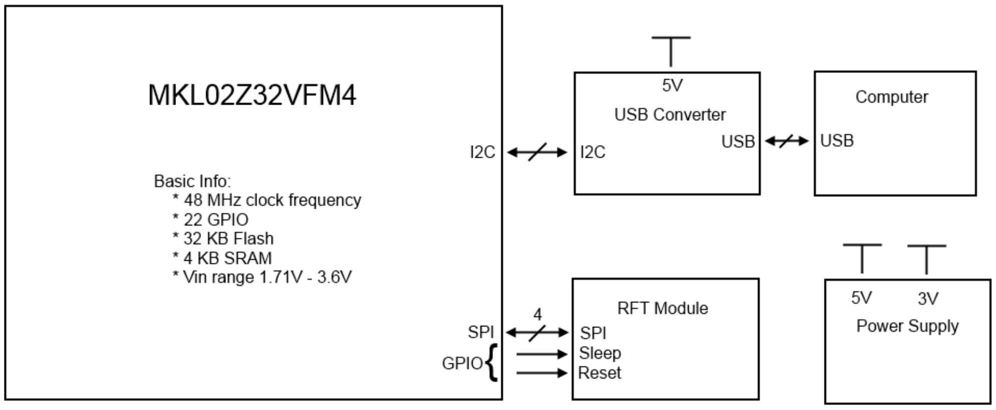


Figure 2: Gun module block diagram

Identically to the gun module, the ref module shares the same MCU, the MKL02Z32VFM4, and the RFT module. Refer to figure 3 for the ref module block diagram.

The USB block contains the necessary features for communication to the computer and for drawing power from the computer. The module communicates to MCU block via I2C bus.

The power supply block receives the 5V supply from the computer and converts it to a usable 3V for the other blocks. The components used in the power supply will likely be a buck switching DC-DC converter.

2.3. Prioritized List of Features:

High Priority

1. All the gun's hardware and basic software required for a standalone game.
2. Basic ref module connecting to the computer via USB.
3. Create a GUI on the computer for the ref module.

Low Priority

4. Create a database to store stats of the game.
5. Create a website to host the stats of the games.

2.4. Hardware Description

The ideal microcontroller focusses on 2 things, low power and IO. Since all communication is fairly slow, this project is time insensitive. Additionally, the instruction set can be simple since there aren't any complex calculations that need to take place. The microcontroller (MCU) used for this project is going to be the Freescale MKL02Z32VFM4, which is an ARM M0+ core. The KL02 runs at 48 MHz, has 32 KB Flash, has 4KB SRAM, and has 22 GPIO/PWM/Bus pins. The MCU has several resources and busses available, and the ones used on the gun module will be the SPI bus, UART bus, and all 22 GPIO pins. The ref module will use I2C, some GPIO, and will be connected to a USB converter. This MCU is a good choice for the project since it is very low power but has extensive IO and busses to connect to several peripherals.

2.5. Software Description

The software for both the gun module and the ref module will be written primarily in the C programming language, with occasional sections written in assembly. C provides a good balance between low level flexibility and high level ease of development.

The two modules have many similarities in design. Since they communicate with each other, they must share all message codes and methods of transmitting and receiving messages. Besides that, their functions are completely different. The gun module is used to actually play the game, however the ref module just collects the data and helps set up and manage games. The two modules have several tasks they must complete under certain circumstances, which are listed below.

2.5.1. The gun module

- User interface task:
  - This task periodically checks any UI buttons and responds to them as described below in the user interface section. This task has to check 3 different buttons, update the LCD display, manage the backlight for the LCD display to minimize power, and manage the speaker.
- Projectile task:
  - This task handles firing the IR messages, or "projectiles". Each projectile is modulated with a message to be sent to another player. The encoding of the messages are described below in the communications protocol section. The minimum time between pulses is 600µs, so this task must run on integer divisors of this pulse time.
- Sensor task:
  - This task handles sensing incoming projectiles, and responding to them. After a projectile has been fully received it must be decoded to reveal its message. The sensor task will then respond to the messages and if the ref module is used, create a message to be sent to it. The actual sending of the message though is handled by the transceiver task.
- Transceiver task:

    - This task handles the receiving and transmitting of data via the radio transceiver, when it is used. It will receive messages on the initialization of the match before the game starts, then send data about the game while in-game. The communications protocol is described below in the communications protocol section.

2.5.2. Ref Module

- Transceiver task:
    - This task handles the receiving and transmitting of data via the radio transceiver. It will register players set up the game options pre-game and collect data and manage the game in-game. The communications protocol is described below in the communications protocol section.
- User task:
    - This task handles all the UI for this module. It will request the user for game options in the pregame and display streaming data about the game while in-game.

2.6. User Interface

UI devices used on the gun module are:

- 3x buttons: 2 of these buttons will be mounted on the chassis of the gun close to the LCD display, and the other button will be the trigger to the gun.
- 8x2 LCD display: This display will be mounted on the chassis of the gun. This display will display relevant information to the user at all times.
- Speaker: A speaker will be mounted inside the chassis of the gun with a few holes to allow sound through it. The speaker plays sounds in response to events, e.g. shooting or getting shot will each play a unique sound.

The user navigates the menus by using these buttons. They can either setup their own game or search for a game hosted by a ref module. There is a countdown from the pre-game to the time the game starts to help synchronize it for all players. While the game is going the player uses the buttons to perform various tasks (eg the red button reloads).

The flow diagram of the UI is displayed below.

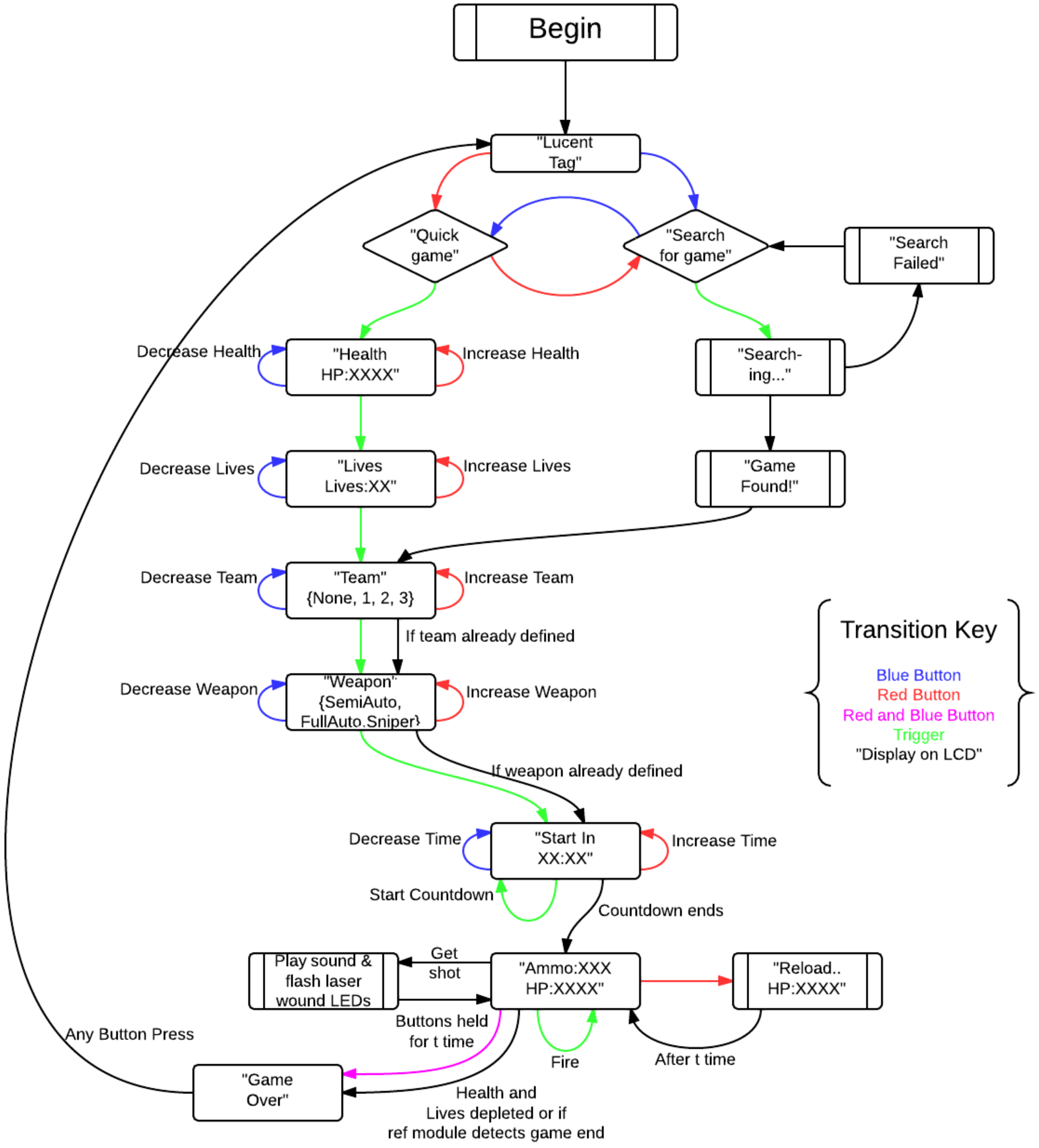
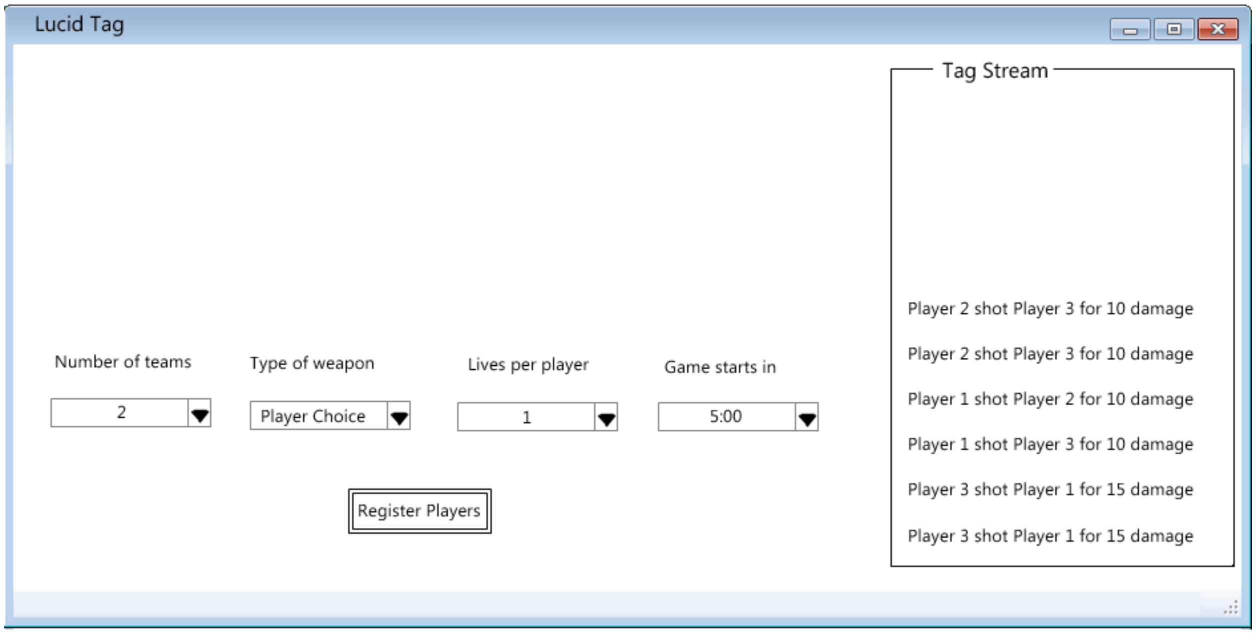


Figure 4: Gun module UI flowchart

The ref module uses a graphical user interface written for a computer. The GUI offers options for the game mode to be sent to players before the game starts, and while the game is underway it will display a stream data from the game. The user has the ability to change the total number of teams that are available to play in, which weapon each player gets to use, how many lives each player gets, and the time when the next game starts. The choice of weapon and team can be made by the player if chosen on the GUI. While the game is running, the ref module will be


Lucid Tag

Number of teams: 2

Type of weapon: Player Choice

Lives per player: 1

Game starts in: 5:00

Register Players

Tag Stream

Player 2 shot Player 3 for 10 damage

Player 2 shot Player 3 for 10 damage

Player 1 shot Player 2 for 10 damage

Player 1 shot Player 3 for 10 damage

Player 3 shot Player 1 for 15 damage

Player 3 shot Player 1 for 15 damage

Figure 5: Ref module GUI

periodically requesting new data from the gun modules. Any important data will be displayed in the tag stream section of the GUI.

2.7.   Communications Protocols

2.7.1. IR LED Protocol

Every “projectile” shot between players has a message encoded in IR light. The encoding protocol was developed by MilesTag, an open source laser tag system. The full protocol can be read here <http://lasertagparts.com/mtformat-2.htm>.

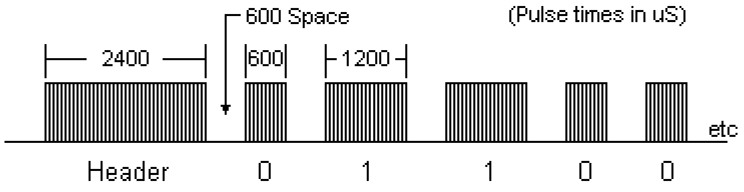
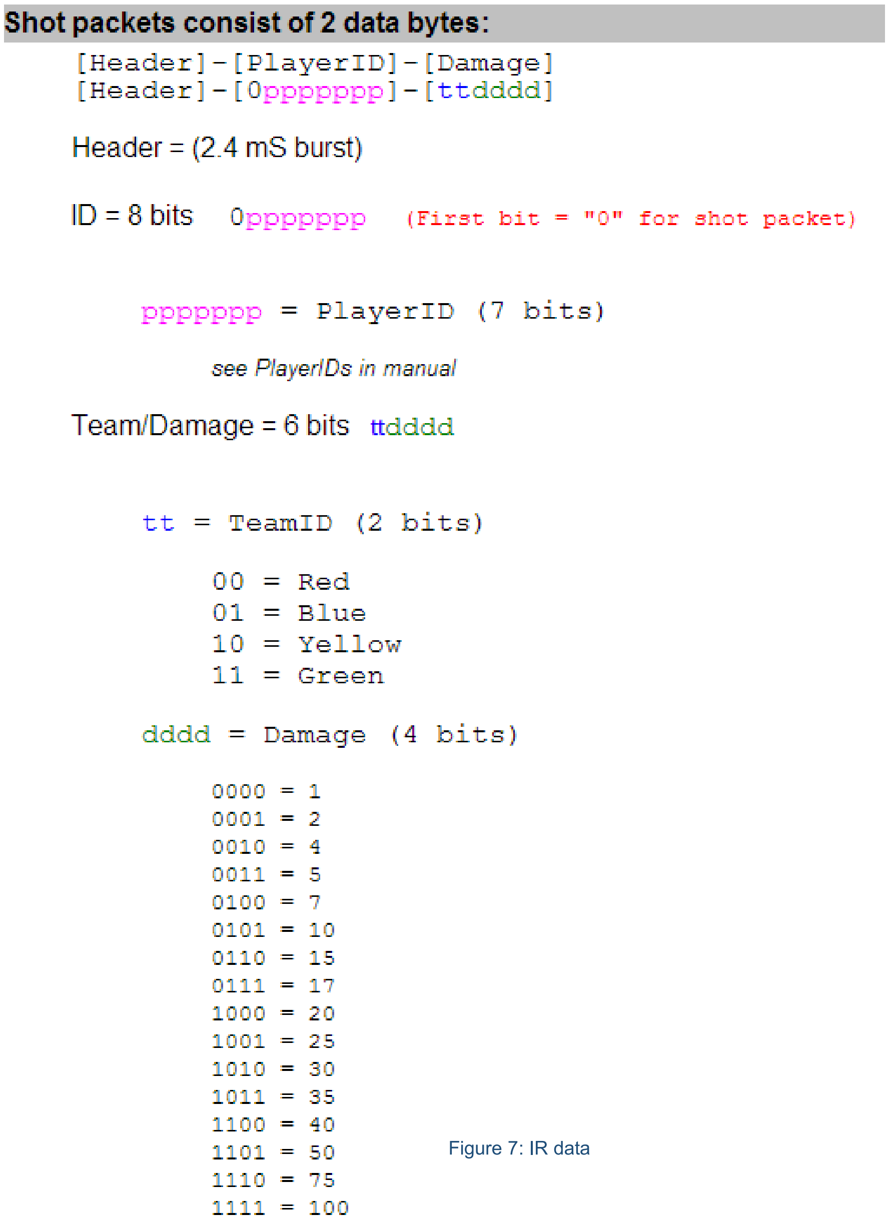


Figure 6: IR message

Each message is constructed serially beginning with a header, followed by the data. The header is a pulse of light lasting 2400µs, the HIGH (1) pulse lasts 1200µs, and the LOW (0) pulse lasts 600µs. Each pulse is separated by 600µs of no signal. The standard projectile message is 16 bits long (excluding the header), meaning the whole message will always be between 50.4ms and 58.8ms. Additional longer "system" messages are available.

For the standard message, after the header is sent 2 bytes of data are sent. The message always begins with a 0 bit to help further with initialization. After that a 7 bit player ID is sent. Beyond simply explaining where the projectile originated, there are some reserved player IDs for special units. Afterword a 2 bit team ID is sent to denote how user will react to the shot. Lastly is the damage taken by the user on a successful hit. Different weapons will deal different amounts of damage.



Shot packets consist of 2 data bytes:

```
[Header]-[PlayerID]-[Damage]
[Header]-[0ppppppp]-[ttdddd]
```

Header = (2.4 mS burst)

ID = 8 bits `0ppppppp` `(First bit = "0" for shot packet)`

```
ppppppp = PlayerID (7 bits)
```

see PlayerIDs in manual

Team/Damage = 6 bits `ttdddd`

```
tt = TeamID (2 bits)

    00 = Red
    01 = Blue
    10 = Yellow
    11 = Green

dddd = Damage (4 bits)

    0000 = 1
    0001 = 2
    0010 = 4
    0011 = 5
    0100 = 7
    0101 = 10
    0110 = 15
    0111 = 17
    1000 = 20
    1001 = 25
    1010 = 30
    1011 = 35
    1100 = 40
    1101 = 50
    1110 = 75
    1111 = 100
```

Figure 7: IR data

2.7.2. Radio Frequency Transceiver

When the ref module is used, it creates a personal area network (PAN) for the use of the game. The ref module becomes a full-functioning device (FFD) by managing and maintaining the network. Each gun module part of the game is a reduced-function device (RFD) meaning they can only converse with the ref module. The protocol that this radio frequency transceiver (RFT) follows the IEEE 802.15.4 standard.

2.8. Sustainability Design Issues

The gun module is powered by 2 rechargeable lithium batteries. With the right power settings, each battery charge can last several hours of constant use, enough for a full day's use. Fortunately the lifespan of a single set of batteries will be very long. In full production, the ideal material for the chassis is plastic. Although each module is built to last a long time, disposal of modules should be expected. Each unit is expected to be composed of about 4lbs of plastic and 1lb of electronics and hardware, where nothing is hazardous to dispose of.

The ref module will be powered through USB, so there will be no worrying about battery use and disposal. Each module will be require about 1lb of plastic for its chassis and 1lb for its electronics and hardware.

2.8.1. Power

The gun module will be powered by 2 rechargeable 3.7V lithium batteries. This main power supply will then be split into 5V and 3V voltage rails by buck switching regulators, so we can assume that the power supplies will be around 85%. The maximum power consumption of several large components are listed below in the preliminary parts section.

The power consumption of a lot of these components are misleading in how high they are since most will be on for only moments. I expect one of the biggest power consumers to be the audio amplifier, but that can always be remedied by lowering the amplification of the speaker, or ensuring each sound clip is brief. Also the power consumption listed above for the RFT is for transmitting data, but receiving data requires under half that amount. Utilizing components' sleep modes and buying a powerful enough battery will be crucial to maintaining an acceptable battery life.

3. Development Plan

Here is the planned weekly development plan for the project.

Quarter	Week #	Project tasks	Deliverables
	1	Order all parts by this week	
	2		
	3	Get MCU up and running	
	4	Test out IR	
	5		
	6	Test out RFT	
	7	Test out the sound	
	8		
	9		
Winter quarter	10	Design Circuit board	
Spring Break			
	1		
	2	Fabricate Circuits	
	3		
	4		Hardware Review
	5		Software Systems
	6		
	7		Code Reviews
	8	Program	
	9		
Spring quarter	10	Prepare for presentation day	

Table 1: Schedule

3.1. Development Tools

This project will only require fairly simple development tools. For hardware dev tools, a breadboard and some switches will probably be sufficient. For software development, I will need the Codewarrior IDE to develop and test the code. The MCU comes with a USB debugger on board, so all debugging will be software based. When testing the several systems, an oscilloscope will come in handy too.

3.2. Prototype Description

The gun module will have a fairly complex prototype, however the ref module will have a very simple one. Each MCU board has Arduino header specifications, making it very easy for expansion. I plan to design a PCB that will connect directly to the MCU board that will house other components. The ref module however does not require many extra components, but any that it does require will be in a bread board. Also the ref module takes its power from the computer, so power supplies will not be an issue.

For the project demonstration I plan to have 2 full functional gun modules and a single functional ref module plugged into the computer. The gun modules will be set up to play against each other with a stream on the computer of any data it receives. Additionally, I will create a tri-fold poster with information for attendees to read about the project.

4. Electrical Specifications

- Project specifications
  - FCC requirements: Since an RFT is used, then I must abide by FCC guidelines for the radio frequency spectrum. The frequency band I will be using will be around 433MHz, which is within an amateur band, so I may use it freely.
- Special environmental requirements
  - Shock
  - Moisture

Gun Module Specifications			
Component	Parameter	Value	Unit
	Life	2400	mAh
	Voltage	3.7	V
	Power MAX	2.8	W
	Power Nom	590.1	mW
Battery	Lifetime	20+	hrs
	X	4	in
	Y	2.25	in
PCB Size	Z	2	in

Table 2: Gund Module

5. Preliminary Parts List

#	Component	Voltage	Max Current	Max Power	Nom. Power	Cost	Lead	Source
3	KL02Z Microprocessor	3V	6.4mA	19.2mW	13mW	0.82	3 weeks	Freescale
2	IR LED (TSAL6100)	1.6V	100mA	1.6W	172.8mW	0.15	8 weeks	Mouser
2	Resistor for IR LED	3.4V	100mA	0.34mW	.036mW	0.34	8 weeks	Mouser
8	IR Rx (TSAL 4856) (8x of these)	5V	5mA	25mW	25mW	0.64	9 weeks	Mouser
3	AT86RF231	3V	14mA	43mW	2.3mW	5.28	1 week	Digikey
2	Sound card (ISD17XX)	3V	39mA	117mW	23.4mW	2.72	1 week	Aliexpress
2	Audio Amp	3V	333mA	1W	200mW	.90	1 week	Jameco
10	Muzzle/hit LEDs (5 of these)	1.8V	20mA	36mW	1.8mW	.07	2 weeks	Digikey
10	Resistor for muzzle/hit LEDs (5 of these)	1.2V	20mA	24mW	1.2mW	0.001	2 weeks	Digikey
5	Buck-Boost Converter	5V	0.06mA	0.3mW	0.3mW	4.28	0 weeks	Digikey
3	Crystal CX-4025	--	--	--	--	0.31		

Table 3: Preliminary parts list