Remote Fireworks Launcher

ECE 445 Fall 2018 Design Document

Daniel Middendorf, Michael Hlinka, Trenton Sanford
Team 5
TA: Kexin Hui
10/4/18

1) Introduction

1.1) Objective

There are many annual occasions where fireworks are a staple in celebrating. These occasions, most notably the Fourth of July, often have large shows where fireworks are launched by professionals. Unfortunately, there are also plenty of amateurs during these events that launch fireworks privately. In 2016, it was estimated that 11,100 people were injured and 4 deaths were reported as firework related [1],[2]. These numbers have varied a bit over the past decade, with the lowest estimate of 7,000 in 2008 and the highest estimate of 11,900 in 2015 [1]. Most injuries from fireworks are from sparklers, while most of the firework related deaths are due to close proximity when lighting off mortar firework rounds.

> *"On July 4, 2016, a 42-year-old male from Florida suffered fatal injuries when the fireworks device he was lighting malfunctioned. According to the county deputies, the victim was trying to set off large mortar-type fireworks in a PVC pipe that was anchored to the ground." [1]*

There is not much to be done about the individuals' safety while using sparklers, but more can be done to help prevent the more serious accidents. Our solution for these types of injuries is to design and implement a wireless launcher that can ignite mortar rounds for amateur consumers. The user would be able to observe and control the launching of the fireworks a safe distance from the launch deck. Our design would be equipped with motion sensors to detect anyone within close proximity of the fireworks. We want our product to emphasize safety above all else. While backyard fireworks are always dangerous, we want to help minimize injuries at an affordable cost.

1.2) Background

Right now, there are some products that help set off fireworks from a safe distance. Cobra Firing Systems provides professional equipment for large shows. This equipment ranges from $500 at its cheapest, to $1350 [3],[4]. This is obviously not reasonable for the average consumer. Firefly Firing System offers a cheaper alternative for firework enthusiasts, at $200 [5]. While this is cheaper than the professional firing systems, there are no safety measures in place to indicate when someone may be in danger around the fireworks.

Our proposed solution would be cheaper and, more importantly, safer than what is currently available on the market while still offering a reliable (95%) success rate.

1.3) High-level Requirements

- The controller must be able to reliably communicate from at least 150 feet away from the receiver, which must be igniting fuses at a 95% success rate.
- Reliably detect people within an 8 foot radius from the firework launch deck.
- Entire launcher system must be as affordable as possible, ideally under $120.

2) Design

To make this fireworks launcher complete, we designed three different parts: the user's controller, the main receiver, and two identical sensor pods (referred to simply as pods). The controller is what the user will interact with to launch the fireworks. The receiver is where the data from the controller and pods are processed and interpreted. This is also where the fuses for the igniters will be attached. The pods are small units that contain sensors to cover more area. Each unit will have a power system that will consist of batteries that will provide the internal modules with the power they need. Each unit will also require an RF module so that all units are able to communicate with each other. The receiver and pod units will contain motion sensors that detect if people are within the launch area. Lastly, the receiver and controller will have a control unit responsible for: displaying information, interpreting data, and deciding when to ignite a fuse.

2.1) Block Diagram

Along with designing the modules for each of our different parts, we also need to understand how, and where, our different parts will be located and interact with each other. We came up with a design where the pods and receiver will be positioned around the firework launch pad while the user, who has the wirelessly connected controller, will be a safe distance away. Refer to **Figure 2.1.1** for a more quantitative description.

After knowing where each part is located, we can then move on with figuring out what different components we are going to need for each part, and get a general sense of power and data flow. Our block diagrams for the individual components can be seen in the figures below.

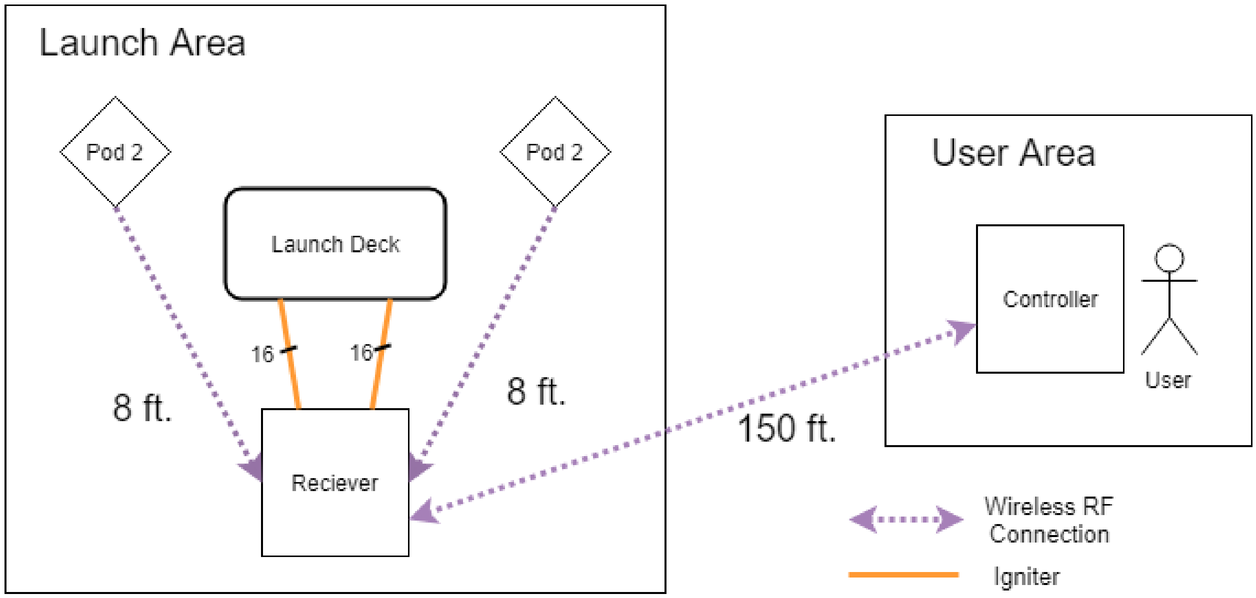

Figure 2.1.1 Space Diagram

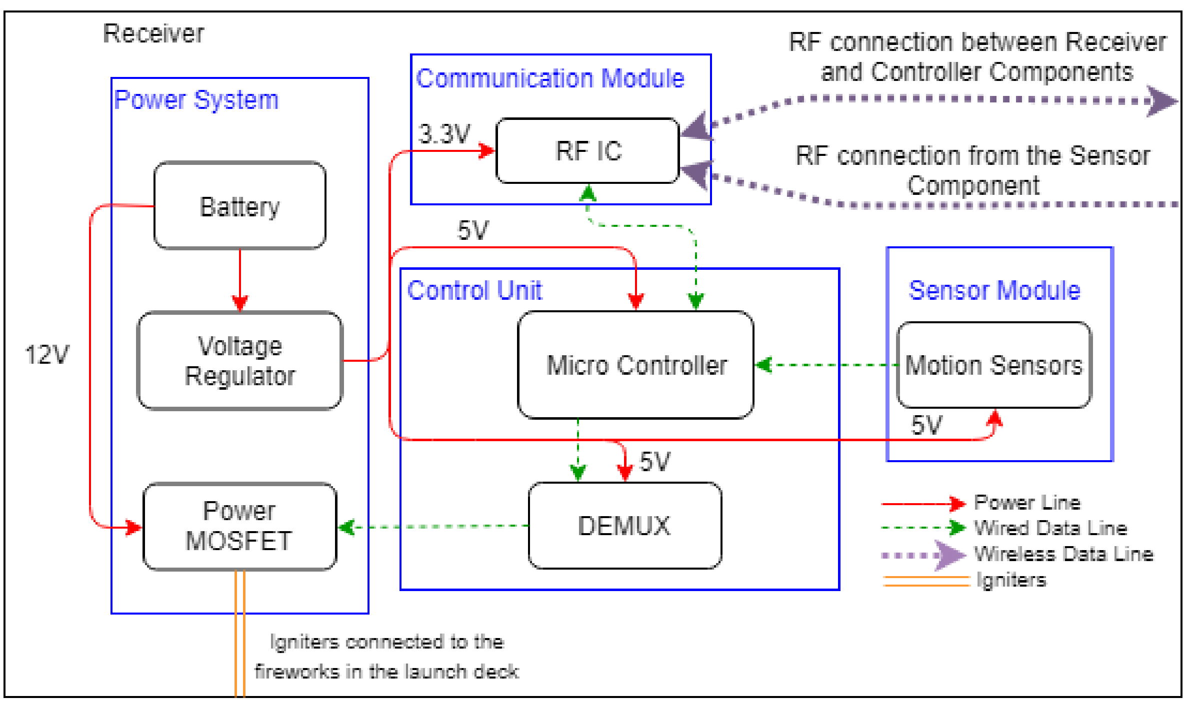

Figure 2.1.2 Receiver Block Diagram

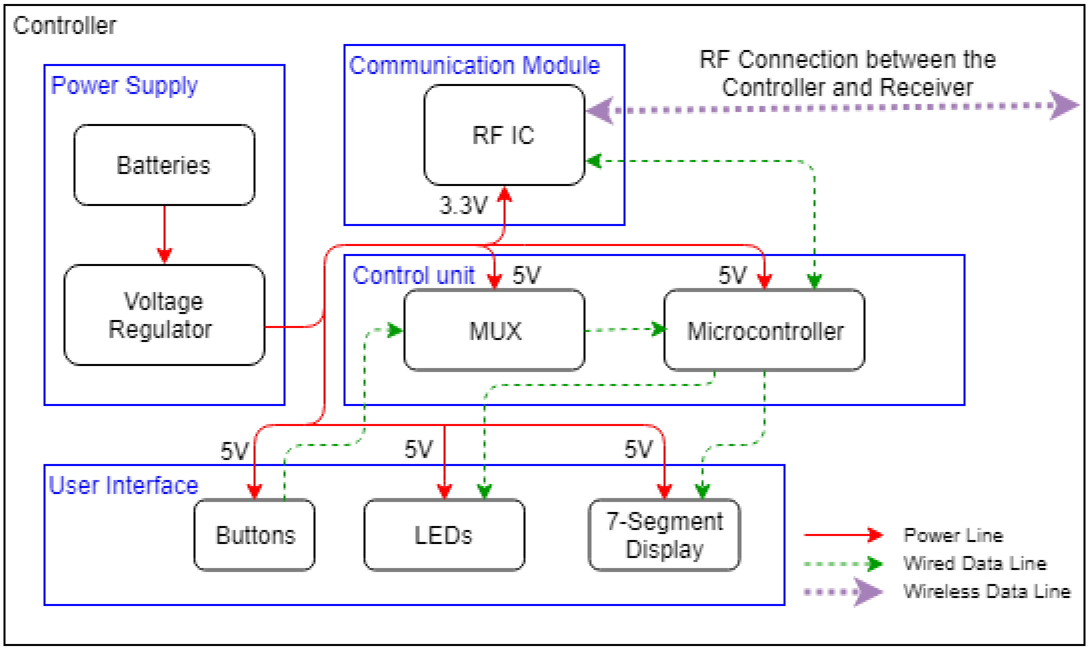

Figure 2.1.3 Controller Block Diagram

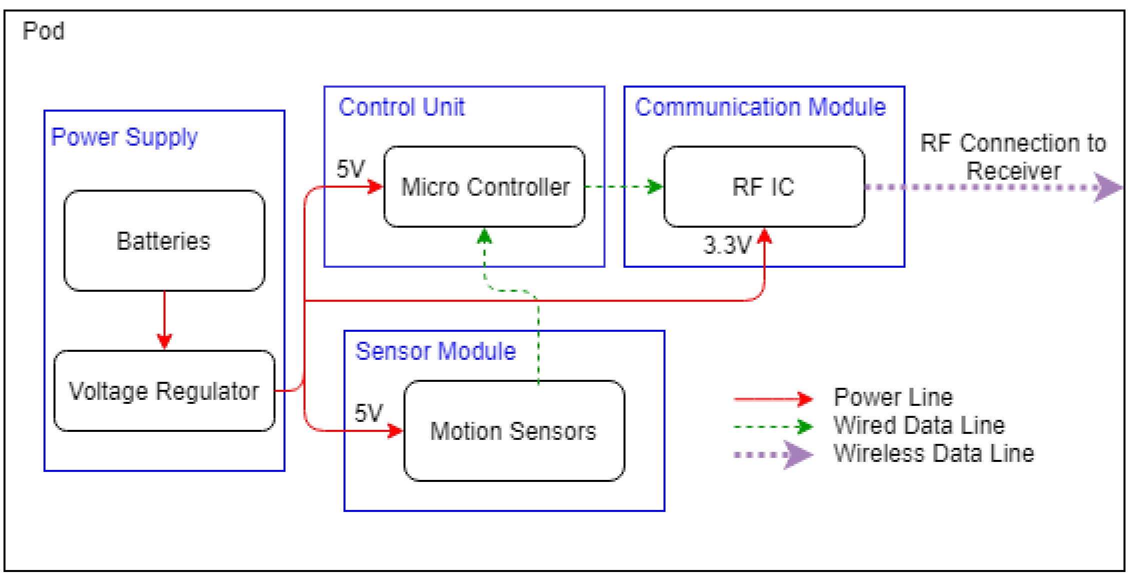

Figure 2.1.4 Pod Block Diagram

2.2) Physical Designs

Since we are making a product that is not reliant on other hardware (other than disposable fuses), physical designs were necessary, and developing three different systems, we needed to make three different physical containers to house our different components. We knew that the biggest electronic component in our project is going to be the receiver PCB. Since this project is more geared towards designing the electronic components instead of the physical, we decided to go with a minimalistic design. We are not completely set on exactly how our devices will end up looking, but we were able to develop the basic shapes and how most of the parts will be fitting together.

When coming up with these designs, we first started off with our smallest piece, the pod. Other than including spots where we think the sensors should be, we added trapezoides to two sides of our pods. These will allow the user to attach the pods to the receiver for easier storage and to help protect the sensors from getting damaged. The added trapezoides fit into holes cut out of the receiver.

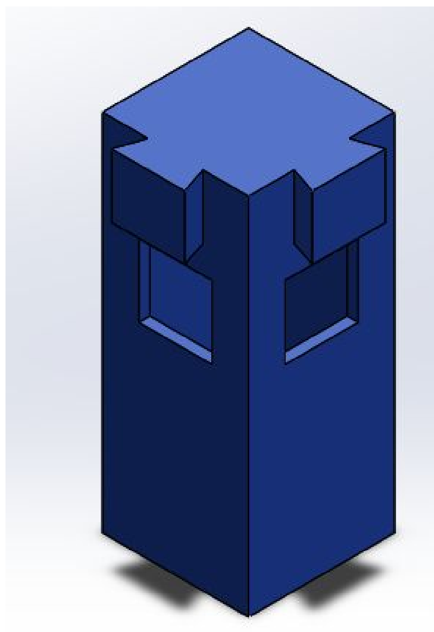

Figure 2.2.1 Physical Pod Design

After designing the pods, we moved our attention towards how we are going to store them. We wanted to have a way to store them nicely while not interfering with the other components of our project. We chose the design below based on the reasoning that it allowed us to attach both pods on a single side while still allowing room for another sensor to be mounted in the front. The front has slots carved out of it so the pods can be slid in and the trapezoides will allow for a firm hold. On either side of the receiver we plan to attach speaker connectors that the fuses will then be plugged into. Currently we are planning on using Uxcell a1312250ux0991 spring loaded connectors [9]. Each connector will be able to house four fuses.

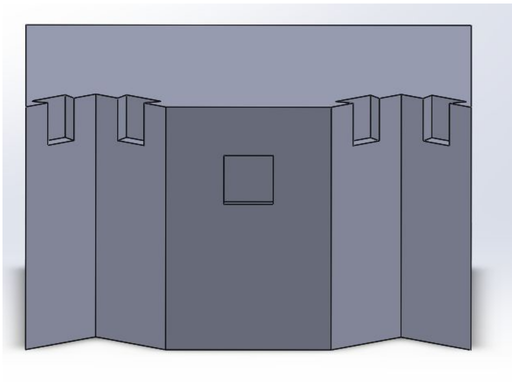

Figure 2.2.2 Front view of Receiver

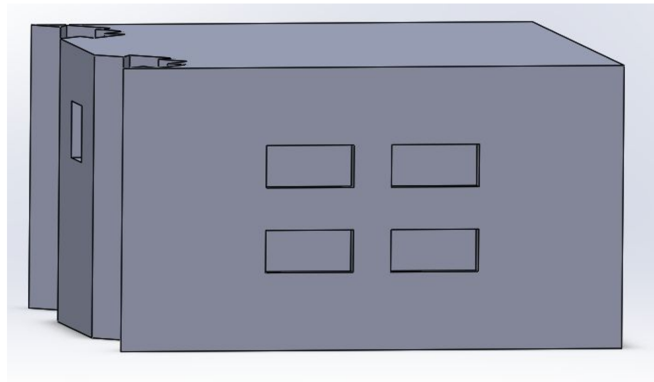

Figure 2.2.3 Side view of Receiver

Our last component is the controller that the user will be interacting with. For this we knew that we are going to need holes for the thirty-two push buttons corresponding to the thirty-two different fuses we could potentially wire up. We also need a space for the 7-segment display we plan on using to display error codes to the user along with an acknowledge button for the user to clear the message.

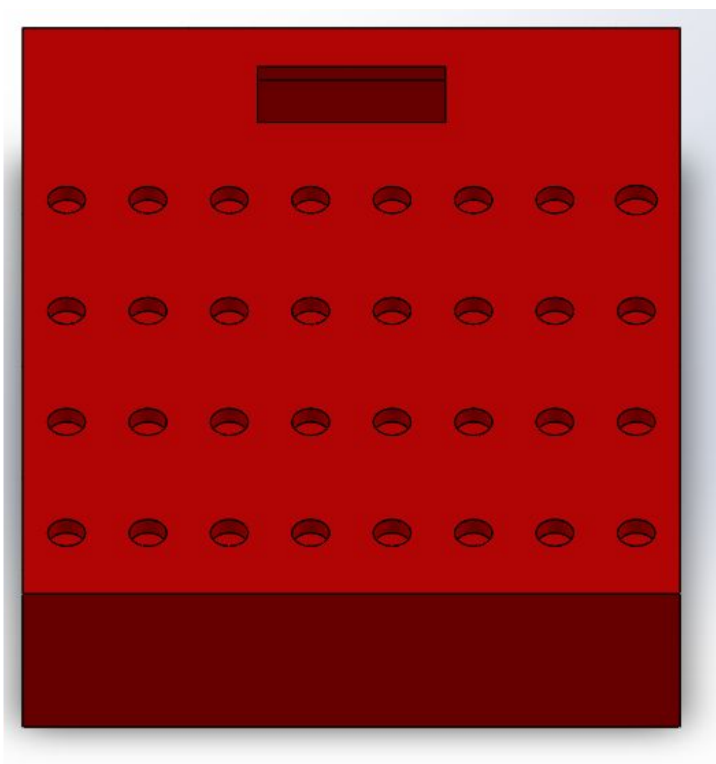

Figure 2.2.4 Physical Controller Design

2.3) Block Descriptions

2.3.1) Power Supply

- **Batteries**
  These batteries must be able to provide the current and voltage necessary for the other modules within our three main components for at least one entire session of launching fireworks. We plan to use store bought batteries as much as possible in order for this project to meet a customer friendly design. The receiver will use ABENIC DC 12V Li-ion battery, the controller will use 4 AA batteries, and the sensor pods will each use 3 AAA batteries.

- **Voltage Regulator**
  Step down the batteries voltage to either 5V for the IC circuits or 3.3V for the communications circuit so the parts are not damaged. We will be using the adjustable LD1117STR voltage regulators for the 5V step down and the 3.3V step down.

- **Power MOSFET**
  The purpose of the power mosfet is to act as a gate to prevent the battery from igniting the fuses before the user wishes to do so. They also protect the more sensitive components of our receiver's circuit, the DEMUX and microcontroller, from the current and voltage needed to ignite the fuses.

2.3.2) Control Unit (Controller)

- **MUX**
  For translating all the button information to the microcontroller we plan on using the ADG731. The IC is a 32:1 analog multiplexer. We will use the simple pull-up resistor circuit shown below for each of our input buttons. The microcontroller will cycle through each 32 input channel and monitor the data line. Whenever the microcontroller reads a low signal in the data line, it will interpret that as the user pressing the button corresponding to the current binary number the controller is outputting.

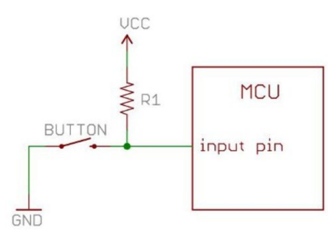

Figure 2.3.2.1 pull-up resistor circuit

- **Microcontroller (Controller)**
  The controller's primary microcontroller will manage input information from the buttons, feedback from the receiver, and communicate that information back to the user. When a button is pressed the buttons number will be queued until the acknowledge button is also pressed. Once the acknowledge button has been pressed, the microcontroller will send the launch information to the Communication Module. The controller's microcontroller will also read feedback from the receiver and display error messages accordingly. The functionality of this microcontroller is described below.

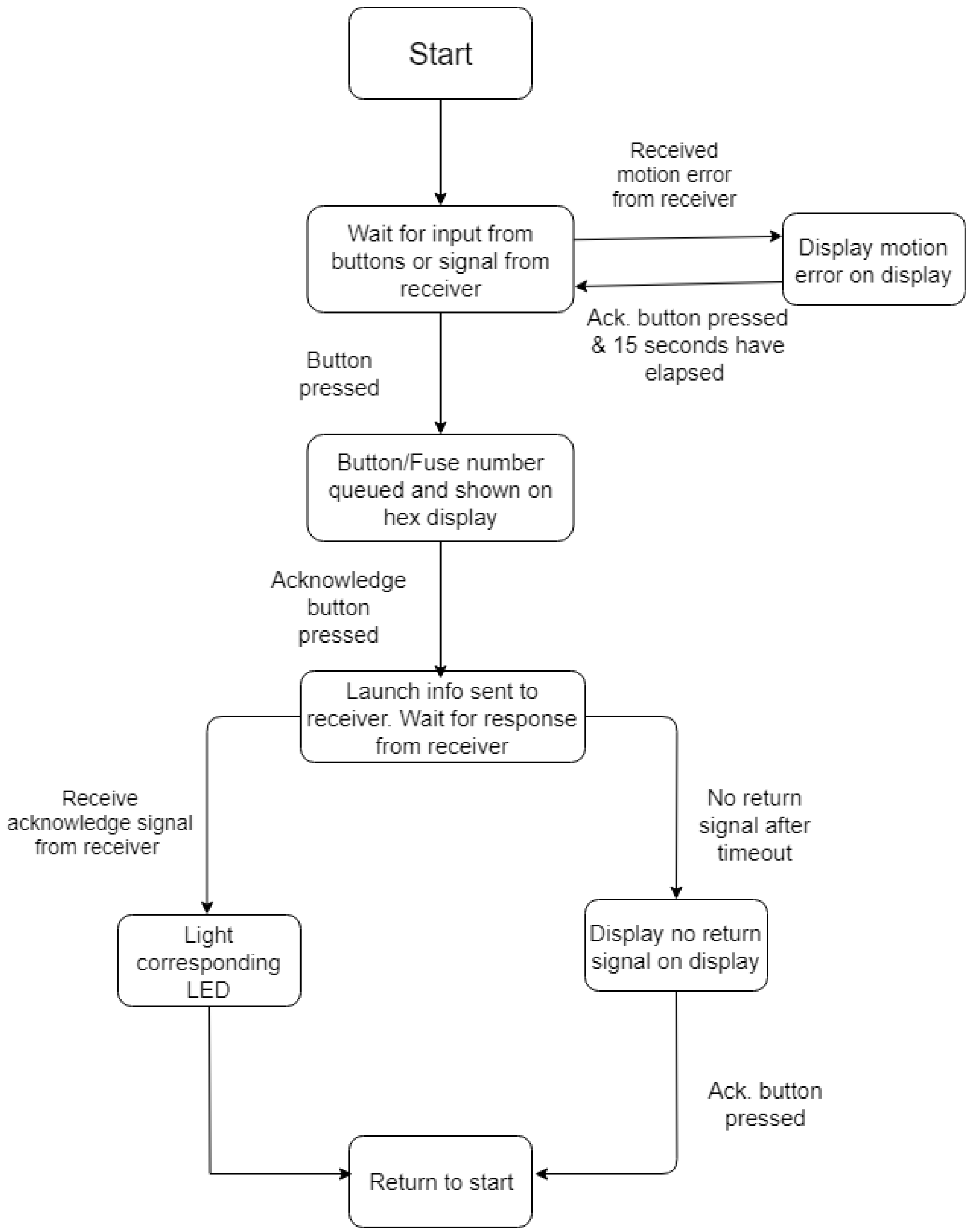

Figure 2.3.2.2 Controller Software Diagram

2.3.3) Control Unit (Receiver)

- **Microcontroller (Receiver)**
  The receiver's microcontroller will check for signals through the Communication Module. If it receives data from the Sensor Module, it will stop accepting launch commands from the controller for a timeout period (~15s) and send an error message to the controller. Otherwise if it receives launch data from the controller, it will send a signal to the decoder to light the specified fuse. The functionality of this microcontroller is described below.

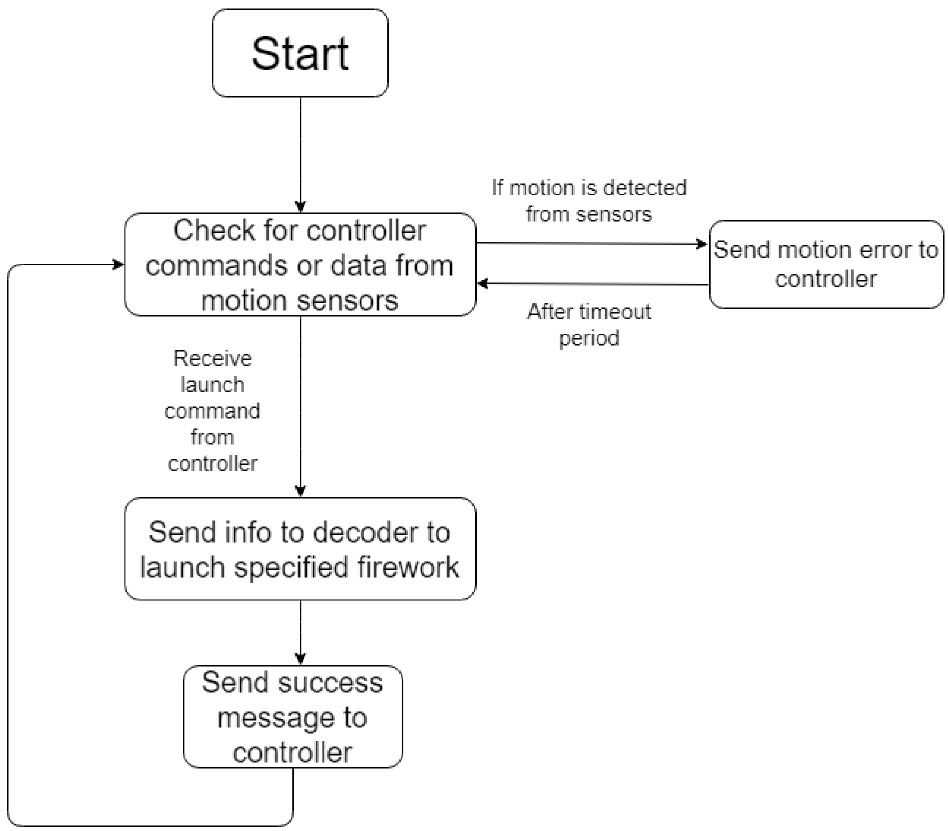

Figure 2.3.3.1 Receiver Software Flowchart

- **DEMUX**
  The demux will take the binary number of the fuse wanting to be launched and allow a high activation signal to be sent to the corresponding numbered MOSFET.

2.3.4) Control Unit (Pods)

- **Microcontroller (Pods)**
  The microcontroller in the sensor pods will be constantly checking the input from the Sensor Module until it detects a spike in motion data. If so, the microcontroller will send a motion error through the Communication Module to the receiver.

2.3.5) User Interface

- **Buttons**
  The buttons are what the user will mainly be interacting with. There will be 32 launch buttons each representing a fuse that can be ignited. Pressing a button will signal the microcontroller to prepare to send the signal to launch that firework. An additional acknowledge button must be pressed to before the launch signal is sent to the receiver. The acknowledge button will also clear errors on the controller.

- **LEDs**
  There will be a LED over each of the 32 launch buttons to let the user know which buttons have been pressed and received a launch confirmation signal from the receiver.

- **7-Segment Display**
  Currently we believe that we will only need a 4-digit 7-segment display to convey the different types of error codes the user will ever encounter. Keeping it simple allows us to keep the cost down and we will only need a limited amount of unique errors. The display will show the fuse number that is queued in the controller after a button is pressed. It will display a brief success message after the controller receives an acknowledge message from the receiver, as well as display unique error messages for motion detected near the launch pad and a time out after sending a launch signal.

2.3.6) Communication Module

- **RF Module**
  As of now, we are planning on using the nRF24L01 as our RF Transceiver. This transceiver requires the use of a microcontroller and will be directly tied to our control unit. We plan on utilizing the SPI pins on both the ATmega328p and ATmega2560 in order to create a network for these devices. The nRF24L01 is cost effective and capable of 2-way communications. This will allow the receiver to get data from the sensor pods and controller while also communicating error messages to the controller for the user to see.

2.3.7) Sensor Module

- **Motion Sensors**
  The sensors will detect motion within 8 ft of the launch pad and send an error signal to the receiver if a spike is detected. The receiver will then throw out any launch commands from the controller and send it the corresponding error message.

2.4) Module Circuit Designs

Some of the modules in our design consist of a single chip, like the MUX and DEMUX. Others require more circuitry design on our part. For the controller, we already showed how the buttons would be wired up and connected to IC chips in our case the 32:1 MUX (**Figure 2.3.2.1**). For the other components of the user interface we are going to be using a very common method that utilizes a human's Persistence Of Vision, or POV for short. This method treats LEDs (and the 7 segment display) as a grid with one of the terminals connected as a row and the other connected as a column. When only allowing one row to have current flowing through it, the microcontroller can power any column to make that LED light up. After those LEDs have been lit, the controller then selects a new row to have current flowing thought it and the process is repeated. When doing this fast enough it appears that all the LEDs can be lit at the same time. Example circuits of our display and LEDs can be seen below.

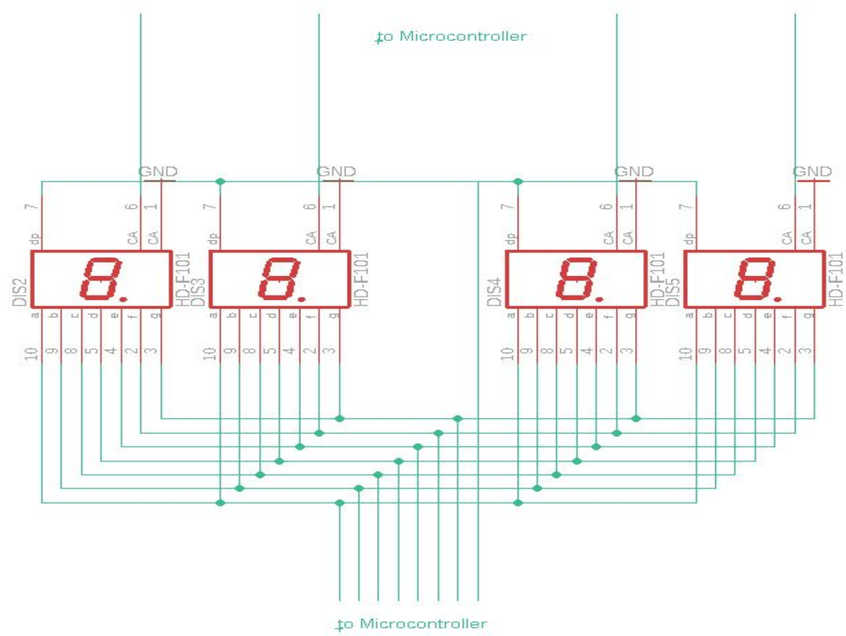

Figure 2.4.1 7-segment Display Circuit

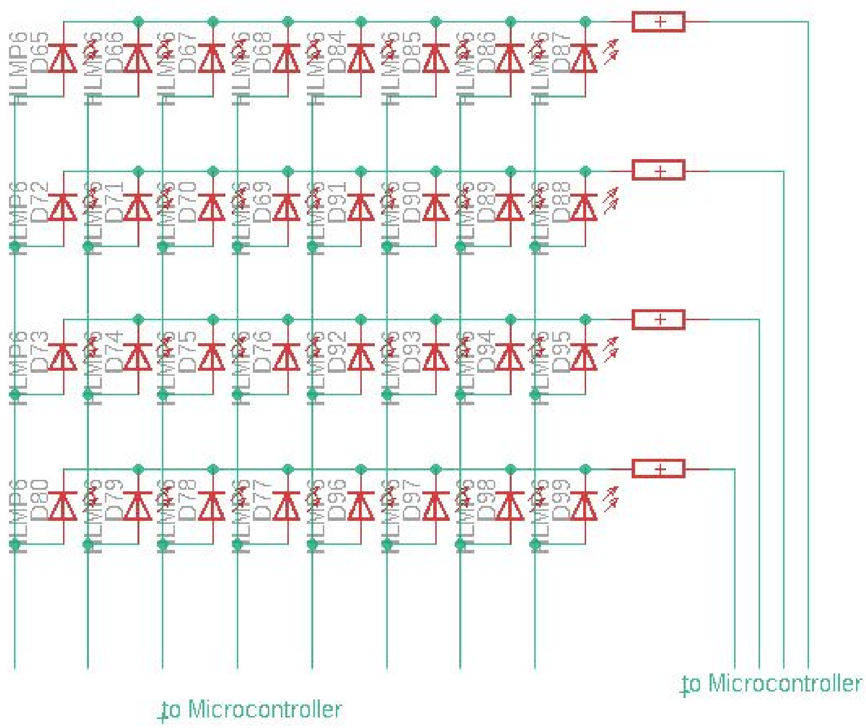

Figure 2.4.2 LED Circuit

Another important circuit we needed to create was the power circuit for our fuses. The battery is directly connected to the drain of our mosfets, the gates are connected to the output of our DEMUX and the source leads to our fuses. A simple diagram of one of our fuse circuits can be seen below.

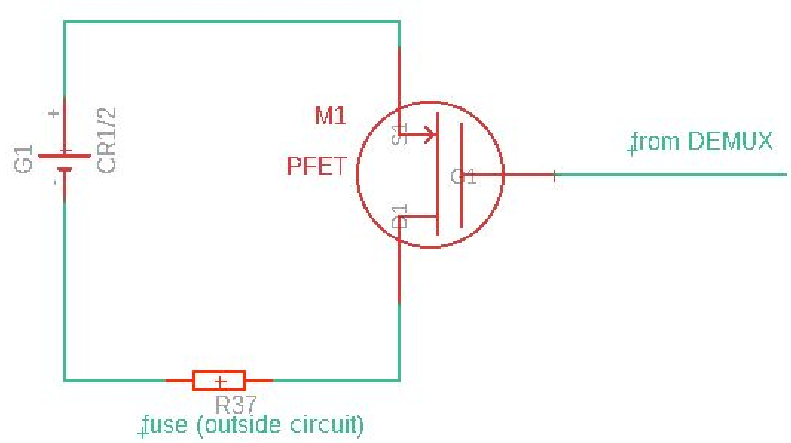

Figure 2.4.3 Fuse Igniter Circuit

2.5) RV Table

Block	Requirements	Verification	Points
Voltage Regulator	1. Reduce any voltage >5V to 5V +/-5%. 2. Reduce any voltage >3.3 to 3.3V +/-5%. 3. Operate continuously for over 1 hour under 125 degrees Celsius.	1. Verification for Requirement 1 & 2 a. With a power supply, apply 6V to the input pin of the regulator and connect GND to GND b. Probe the output pin and GND pin with a digital multimeter and read output. c. Confirm that output on multimeter is within 5(3.3)V +/-5%. 3. Verification for requirement 3 a. Apply 6V to each regulator. b. Run circuit for 1 hour. c. Measure temperature of each regulator and confirm it is less than 125 degrees celsius.	(6)
Power MOSFETS	1. The MOSFETs are able to send the 12V & 1A to the igniters.	1. Verification for Requirement 1 a. Tie the drain of the MOSFET's to a 12V source. b. Attach a resistor from the MOSFET's source pin to GND c. Toggle the voltage at the gate of the MOSFET between Vcc (5V) and GND and probe the voltage drop across the resistor. d. Confirm a voltage drop of 12V +/- 1.2V with 1A +/- 0.1A of current.	(4)
MUX	1. Take in all 32 different button signals and output the signal of the numbered button corresponding to the binary representation sent from the microcontroller.	1. Verification for requirement 1 a. Tie all inputs to Vcc (5V) b. Connect the selection data lines and output to the microcontroller. c. Switch one input line to GND. d. Observe if the corresponding binary number is represented with the LEDs.	(3)

Microcontroller (Controller)	1. The microcontroller recognizes input from the MUX. 2. Microcontroller can be programmed to follow the logic in **Figure 2.3.2.2** 3. Microcontroller can send data to and from the communication module	1. Verification for requirement 1 a. Tie output and selection lines of the MUX module to GPIO pins on the microcontroller. b. Program microcontroller to cycle through all 32 possible input lines c. Tie one input line from the MUX to GND d. Program microcontroller to output the binary number of the data line that was tied to GND to a bank of LEDs. 2. Verification for Requirement 2 a. Write code following the logic in **Figure 2.3.2.2** b. Test software behavior given hard coded inputs 3. Verification for for Requirement 3 a. Setup 2 communication modules with one using this microcontroller. b. Attach a button to one of the microcontroller's GPIO pins for both microcontrollers and have it send a signal through the communication module when pressed. c. Have the receiving communication module's microcontroller light an LED if it receives a message.	(6)
Microcontroller (Receiver)	1. Microcontroller can be programmed to follow the logic in **Figure 2.3.3.1**	1. Verification of Requirement 1 a. Write code following the logic in **Figure 2.3.3.1** b. Test software behavior given hard coded inputs	(6)

	2. Microcontroller is able to read inputs detect spikes in motion sensor data	2. Verification of Requirement 2 a. Power motion sensor and connect its output pin to a GPIO pin for our microcontroller. b. Program microcontroller to detect a significant change between readings and power a small LED circuit when one is detected.	
	3. Microcontroller can send data to and from the communication module	3. Verification for Requirement 3 a. Setup 2 communication modules with with one using this microcontroller. b. Attach a button to one of the each microcontroller's GPIO pins and have it send a signal through the communication module when pressed. c. Have the receiving communication module's microcontroller light an LED if it receives a message from the sending RF IC.	
	4. The microcontroller is able to send the binary value of the fuse number to the DEMUX and send a high signal	4. Verification for Requirement 4 a. Program the microcontroller to cycle through all possible binary values from 0 to 31. b. Output each index of the binary value to a GPIO pin (5 in total). c. Connect each GPIO pin to a LED and resistor in series connected to GND. d. Using the LEDs verify that all possible 32 combinations can be outputted. e. Have one more GPIO pin connected to another LED and resistor and confirm that it can output a high and low signal.	

DEMUX	1. Take a 5 digit binary number and translate it to a unique signal line.	1. Verification of Requirement 1 a. Connect all output lines of the decoder to an LED. b. Connect input pins to Vcc (5V) or GND to create a specific binary input. c. Test with all possible input combinations and see that they each light a unique LED.	(3)
Microcontroller (Pods)	1. Microcontroller is able to read inputs detect spikes in motion sensor data. 2. Microcontroller can properly send data to the Communication Module	1. Verification for Requirement 1 a. Power motion sensor and connect its output pin to a GPIO pin for our microcontroller. b. Program microcontroller to detect a significant change between readings and power a small LED circuit when one is detected. 2. Verification for Requirement 2 a. Setup 2 communication modules with with one using this microcontroller. b. Attach a button to one of the microcontroller's GPIO pins and have it send a signal through the communication module when pressed. c. Have the receiving communication module's microcontroller light an LED if it receives a message from the sending RF IC.	(4)
Buttons	1. Must be easily-pressable	1. Verification for Requirement 1 a. Press button and ensure that it can be done without strain (that would be bad).	(1)

LEDs	1. The microcontroller must remember what buttons have received success signals and have the corresponding LED light up. 2. The microcontroller must cycle through each row of the LEDs fast enough, such that they always appears to be lit.	1. Verification for Requirement 1 a. Program microcontroller to decode binary representation of the button number [0-4] b. Wire output pins to LEDs in series with resistors tied to GND. c. Send binary signals through microcontroller and check that it lights the corresponding LED. 2. Verification for Requirement 2 a. Connect microcontroller to LED circuit in **Figure 2.4.2** b. Modify duty cycle such that flickering/blinking is minimized or unnoticeable.	(2)
7-Segment Display	1. The 7-segment display must be able to display the value given to it by the microcontroller	1. Verification for Requirement 1 a. Program a microcontroller to send a test message to the hex screen using 15 GPIOs. Use 4 of those GPIO pins to cycle through the common anode pins and the others to specify which segments to light. b. Verify that the intended message is displayed on the hex display.	(4)
RF Module	1. Communicate reliably at a range of at least 150 ft with another communications module.	3. Verification for Requirement 1 a. Set up 2 communication modules 150 ft away from each other. b. Send sample data from one module to the other. Run 20 times. c. Verify sample data isn't changed 95% of the time.	(6)

	2. Use control unit microcontroller to perform error checking on received data packets.	2. Verification for Requirement 2 a. Setup communication with another RF IC. b. Use CRC checksum library c. Send random n-bit signals to the other RF IC and check that they are flagged as errors.	
Motion Sensors	1. IR motion sensors must be able to detect movement from 8 ft. +/- 1 ft. away.	4. Verification for Requirement 1 a. Power sensor with 5V Vcc b. Use an oscilloscope to check the analog signal and see if there are fluctuations when we create motion 8 ft. in front of the sensor.	(5)

2.6) Tolerance Analysis

In order for our project to meet our high level requirements, we need to be able to use electricity to ignite a fuse connected to a firework. This means that, at a minimum, we need a power source that can provide 12V and 1A sporadically while also being able to provide power to the other components of the circuit. Originally we had thought about using store bought disposable batteries, but after doing more research we found out that the spikes in current demand would be too much for any reasonable (less than 8) amount of store bought batteries. These batteries would die out too quickly for them to be an option. We settled on using the ABENIC DC 12V Li-ion battery because it makes dealing with the battery much more manageable for the user and it offers the current requirements needed for the fuses. According to the description, this battery has a capacity of 1800 mAh's meaning this battery will be able to power our receiver and provide the ability to ignite our fuses without the need of another power source. This product is advertised to provide between 12.6V and 11.4V [13].

This power will directly feed into the drain terminal of our power mosfet (NTJS3151P). The drain to source resistance is 45 mΩ [12]. Using Ohm's Law, and the lowest rated voltage:

$$V = 1 * .045$$

Gives us a mosfet voltage drop of .045V when the mosfet is turned on. After that the power will be going through our speaker connectors to our fuses. We could not find data sheets for either the speaker ports or the fuses. With this in mind, we will assume a good connection between the speaker ports and the ends of the fuse with a maximum voltage drop of .005V. This brings our total voltage drop from the batteries to the fuses to be .05V. Because this is a commercial product, we believe that the 12V and 1A already have the products tolerances accounted for and these values are the rounded up versions of those calculations. With these assumptions, we believe that the nominal voltage drop needed to ignite the fuses is around 11V (based off of an assumed tolerance of 10%).

If our battery provided below the assumed voltage of 12V, there is a chance that our battery will not be able to provide enough power for the fuses. If this is the case, we would need to get a more powerful battery with higher tolerances and voltage ratings. This analysis of the core aspect of our project gave us perspective on how we need to think about the power needs of each of our different modules and the steps we can take to minimize the chance of failure.

3) Cost and Schedule

3.1) Cost Analysis

This project is designed with the user in mind and because of this we needed to make this project as low cost as possible. Most of the circuit components we are using have a single unit cost and a lower bulk purchase cost. We included those in our cost analysis because as a consumer product we would buy and assemble these parts in bulk. With this in mind, our bulk cost is $35.63 cheaper than our prototype cost and meets our requirement for affordability.

We will be 3-D printing the body of our parts for the initial prototype. Our estimate for a one-off 3-D printing of our 4 main body parts is $30. This raises the total cost of our prototype to $166.24. Ideally for the consumer product we would be mass producing ABS injection molds. We estimate these molds would be $5 per unit. That would bring the total bulk cost per unit to $105.61 which is under our $120 limit.

3.1.1) Parts List

Name (Manu.)	Description	Quantity	Price/Unit	Cost (Prototype)	Cost (Bulk)
ATmega2560 (Microchip)	Microcontroller for the Controller and Receiver	2	$11.85	$23.70	$17.22
ATmega328p (Microchip)	Microcontroller for the Sensor Pods	2	$2.15	$4.30	$3.58
IML-0688 (Murata Technologies)	IR Motion Sensors for the Sensor pod modules and receiver module	3	$3.30	$9.90	$4.08
NRF24L01 (Nordic Semiconductor)	RF Transceiver IC for wireless communications between all 4 parts (Controller, Receiver, and 2 Sensor pods)	4	$1.20	$4.80	$4.80
ADG731BSUZ (Analog Device)	MUX we are using to encode our user button inputs for our Microcontroller. We will also use it to decode our data at the Receiver and light the correct fuse.	2	$10.67	$21.34	$12.46
NTJS 3151P (ON Semiconductor)	Power MOSFET to step up the voltage in the receiver module in order to successfully light the fuses.	32	$0.46	$14.72	$4.16

TDCG1050M-ND (Vishay Semiconductor)	7-segment display used to provide the user visual feedback for error codes and launch confirmations.	1	$2.91	$2.91	$1.04
LD1117STR (STMicroelectronics)	These are adjustable voltage regulators for us to use for both our 3.3V and 5V components	5	$0.43	$2.15	$0.85
HLMP-3507 (Broadcom)	LEDs we will use to indicate which fireworks have been launched.	32	$0.18	$5.76	$5.76
A4YBS Uxcell (UXCELL)	We will use 33 of these buttons. 32 will be directly tied to launching fireworks, and 1 will be an acknowledge button.	33	$0.39	$12.87	$12.87
A13122500ux0991 (UXCELL)	Speaker board connectors will be our interface for the fuse igniters.	8	$1.35	$10.80	$10.80
Rechargeable 12V Li-ion Battery (ABENIC)	Battery for the Receiver.	1	$20.99	$20.99	$20.99
PCB (PCBway)	Circuit boards for each of our components	4	$0.50	$2.00	$2.00
CIRCUIT PARTS TOTAL				$136.24	$100.61
FINAL PARTS TOTAL				$166.24	$105.61

3.1.2) Labor

Our labor cost comes from the average salary of students who graduated with a BS in Computer Engineering in the ‘14-’15 academic year. These students earned an average of $84,250/year [11]. This salary divided between 2,080 hr/year, which is the amount of work hours in a year (40 hours/week * 52 weeks/year), yields and hourly rate of $40.50 which we rounded down to $40/hour.

Name	**Rate**	**Hours**	**Total * 2.5**
Michael	$40/hr	225hr	$22,500
Trent	$40/hr	225hr	$22,500
Daniel	$40/hr	225hr	$22,500
TOTAL LABOR			$67,500

3.1.3) Grand Prototype Total

Total Parts	**Total Labor**	**Grand Total Prototype Cost**
$166.24	$67,500	$67,666.24

3.2) Schedule

WEEK OF	Michael:	Trent:	Daniel:	Checkpoints
10/8	Solidify Power solution for lighting fuses	Start wireless communication tests	Circuit schematics for sensor pods	Have all parts to test ordered
10/15	Begin breadboarding the controller circuit	Breadboard 2-way communications module. Cont. wireless communications test and development.	Breadboard the receiver circuit	Communications module should be complete.
10/22	Put controller modules together	Eagle for the RF module. Begin programming controller microcontroller	Eagle for the sensor pods. Begin programming receiver microcontroller	PCBs must be completed for first round of ordering
10/29	Begin soldering components on controller PCBs	Cont. programming controller microcontroller	Cont. programming receiver microcontroller	
11/5	Assist with soldering and make final edits to PCBs	Solder Sensor Pod PCBs	Solder receiver PCB	Final round of PCB orders
11/12	Assemble final controller module. Test/Debug	Assemble final sensor pod modules. Test/Debug	Assemble final receiver module. Test/Debug	
11/19	Buffer week for reworks / start final report	Buffer week for reworks / start final report	Buffer week for reworks / start final report	Being Thankful for all the good in the world
11/26	Start final report	Start final report	Start final report	Mock Final presentation week.

4) Ethics and Safety

4.1) Ethics

Due to the nature of our project there are some ethical, and legal concerns to take into account. The primary use of this project is to safely ignite pyrotechnic products, specifically mortar fireworks. When using these products, federal and local laws/policies should be strictly followed. The use of fireworks is regulated federally and are illegal in the state of Illinois, including the city of Urbana, without a proper permit. However, smaller pyrotechnic devices such as sparklers, smoke devices, or trick noise makers can be legally used in Illinois without any kind of permit. More details about these laws can be found in Illinois' Pyrotechnic Use Act [7] and the city of Urbana's firework policies [8]. For the purposes of testing and demonstration, we currently only intend to use our project to ignite fuses that would normally be attached to fireworks. These fuses will not be attached to any combustible or explosive device.

Even when used legally, the pyrotechnic devices can still be dangerous. The primary purpose of our project is to promote safety and attempt to minimize such risk (aligning with the first clause of the IEEE Code of Ethics [6]). However, it is possible for the device to be used maliciously to harm someone or their property. This could be in violation of the ninth clause of the IEEE Code of Ethics [6]. There isn't any way for us to govern how our project is used, but we believe that benefit of this projects safety features outweigh the potential for its misuse.

4.2) Safety

Although our project has added safety features to help mitigate risk to users, our features aren't foolproof. External forces and improper usage can result in the harm to the user, property, and others. The added safety features in our project should in no way substitute normal safety precautions when dealing with fireworks. Safety tips like the ones provided by the National Council on Fireworks Safety should still be followed when using our project [10]. Our project should also not be used in adverse weather conditions such as rain storms, strong winds, etc. to avoid unpredictable behavior by the firework.

Considering our wireless design for interfacing between the controller and receiver, it is possible for our receiver to pick up unintended launch commands. This would obviously result in a safety issue, so we're planning on implementing a CRC checksum to detect errors in sent packets. This should greatly lower the chance of a misfire.

References

[1] U.S. Consumer Product Safety Commission, ‘2016 Fireworks Annual Report’, 2017. [Online]. Available: https://www.cpsc.gov/s3fs-public/Fireworks_Report_2016.pdf?t.YHKjE9bFiabmirA.4NJJST.5SUWIQJ [Accessed: 19- Sept- 2018]

[2] Andrew G. Simpson, ‘Facts About Fireworks: 11,000 Injuries, 4 Deaths in 2016’, *Insurance Journal*, 2017. [Online]. Available: https://www.insurancejournal.com/news/national/2017/06/30/456213.htm [Accessed: 19- Sept- 2018]

[3] Cobra Firing Systems, ‘Handheld Remotes’, [Online]. Available: http://www.cobrafiringsystems.com/remote.html [Accessed: 19- Sept- 2018]

[4] Cobra Firing Systems, ‘Modules’, [Online]. Available: http://www.cobrafiringsystems.com/modules.html [Accessed: 19- Sept- 2018]

[5] Firefly, ‘Product Details’, 2018. [Online]. Available: https://shootfirefly.com/pages/product-details [Accessed: 19- Sept- 2018]

[6] IEEE.org, ‘IEEE Code of Ethics’, 2018. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html [Accessed: 15- Sept- 2018]

[7] Illinois General Assembly, ‘(425 ILCS 35/) Pyrotechnic Use Act’, 2009. [Online]. Available: http://www.ilga.gov/legislation/ilcs/ilcs3.asp?ActID=1635&ChapterID=38 [Accessed: 15- Sept- 2018]

[8] City of Urbana, ‘Fireworks’, 2009. [Online]. Available: https://www.urbanaillinois.us/residents/fire-safety/fireworks [Accessed: 15- Sept- 2018]

[9] Amazon, ‘Uxcell a13122500ux0991 8 Pin Spring Loaded Speaker Terminals Board Connector 63mm x 30mm (Pack of 5)’, 2018. [Online]. Available: http://www.uxcell.com/pcs-pin-spring-loaded-speaker-terminals-board-connector-63mm-30mm-p-525690.html [Accessed: 19-Sept-2018]

[10] National Council on Fireworks Safety, ‘Recommended Safety Tips’, [Online]. Available: http://www.fireworkssafety.org/safety-tips [Accessed: 2-Oct-2018]

[11] Department of Electrical and Computer Engineering, ‘Salary Averages’, 2018, [Online]. Availalbe: https://ece.illinois.edu/admissions/why-ece/salary-averages.asp [Accessed: 3-Oct 2018]

[12] ON Semiconductor, ‘NTJS3151P, NVJS3151P Trench Power MOSFET’, 2016, [Online]. Available: http://www.onsemi.com/pub/Collateral/NTJS3151P-D.PDF [Accessed: 3-Oct-2018]

[13] Abenic, ‘ABENIC Super Polymer Rechargeable 20000mAh Lithium-ion Battery DC 12V ,DC122000’, 2017, [Online]. Available: http://w.abenic.diytrade.com/sdp/215983/4/pd-815612/9218861-0/ABENIC_Super_Polymer_Rechargeable_20000mAh_Lithium.html [Accessed: 3-Oct-2018]

[14] Sparkfun, ‘Pull-up Resistors’, [Online]. Available: https://learn.sparkfun.com/tutorials/pull-up-resistors/all [Accessed: 3-Oct-2018]