Autonomous Target Detection NERF Battlebot

Daniel Agudelo, Rachel Gremillion, Aaron Hoyt, Rafael Ramirez

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — The autonomous target detection NERF battlebot is a collaborative project among three engineering disciplines to design and build a robot that can automatically detect targets in an area. The electrical design of the robot is covered within this document. This robot uses computer vision and sensor modalities to search an area for targets and uses NERF blasters to fire upon the suitable targets. This robot will battle against other robot projects to determine the best design and fastest computing time.**

***Index Terms* — Battlebot, Bluetooth, Microcontroller, MOSFET, NERF Blaster, Robotics.**

I. Introduction

Robotics is an advancing technology that applies computing applications to the real world to perform tasks automatically. The scope for this robot is the ability to apply computer vision to search an area for targets that contain human faces, distinguishing what is a real target and then firing upon appropriate targets with a weapon system, all doing so autonomously. This robot uses the deliberative architecture which uses a simple paradigm to sense the world, plan the world and act upon the world. Since the world model used for this robot will be known it was appropriate to build the architecture to fit these needs.

In order to be able to track different targets, a camera module is used as the eyes of the robot. Once the computer algorithms are able to depict the different potential targets in the area, a range finder is the true connection to differentiate how far an object is to be fired upon. If the aiming system needs to be adjusted, there is a stepper motor as well as a dc motor equipped with encoders that are used to pinpoint the exact location to aim the firing system. If the robot needed to relocate itself, the use of manual navigation allows the robot to move freely within an area. Once the robot has the right coordinates and positioning it is ready to fire upon the target.

The robot attempts to create this tracking system within a desired budget of $1000 all inclusive. The goal of this system is to create an affordable, yet accurate system that will accurately aim, detect, and fire Nerf ammunition at enemy targets. This conference paper will go in depth of the process of how the robot is designed and implemented along with the hardware used to be able to carry out these tasks.

II. Battlebot Competition

The Lockheed Martin battlebot competition is set to take place on Friday April 14, 2017. The competition will be held at a Lockheed Martin indoor facility with a layout of a conference room. There will be three rounds in the competition. Each round will last 10 minutes and each team will compete twice. The course will be a 40 x 20 ft (L x W) area. Each round a team's robot will be restricted to a 15 x 10 ft (L x W) zone. There will be a course keepout area in between these two zones that will be populated with 3 obstacles partially blocking a robot's view to the opposite side. Each course zone will be lined with different colored duct-tape borders. Robots are allowed to be equipped with a maximum amount of two Nerf-blasters. Nerf-blasters cannot be modified to increase velocity of firing mechanism. The robot is allowed to carry a maximum of 100 nerf ammunition. A maximum of 50 nerf balls and 50 nerf darts can can be used. The robot is allowed to use up to two Nerf Blasters, but a maximum of one Nerf Ball Blaster can be used.

The point system shown in Table 1. (Point System) denotes how the competition's scoring will be done. Shots landed with Nerf darts are awarded more points than successful shots with a Nerf ball. Hitting an enemy medic is the most rewarding shot. Medics are designated team members that may enter the course to make any repairs or adjustments to their robot during a round. Course targets will consist of pictures of different faces printed on 2 x 2 ft. posters mounted on easel stands about 5 ft. off the ground. The course targets will be stationed in a predefined location, but points will only be awarded a maximum of two times per round. Entering a restricted area or colliding with course obstacles receive a 5 point penalty. If obstacles are knocked down disqualification can occur. If the robot exits the course points will also be docked and disqualification can occur if the robot does not return within 5 seconds.

The robot will be placed in a predefined starting position on the the team's designated zone. The starting position will be a safe area as an obstacle will block its view. The robot's camera feed or view will be displayed on a projector to show detected targets.

Point System	
Enemy Bot Nerf Ball	2
Enemy Bot Nerf Dart	3
Enemy Medic	8
Opposite Side Course Target	5 [Max x2]
Own Side Course Target	2 [Max x2]
Entering Restricted Area	-5
Colliding with Course Obstacles	-5
Colliding with Course Targets	-5

Table 1 Point System

III. Initial Design

The initial design of the robot was inspired by collaboration among three engineering disciplines, mechanical, electrical, and computer science, with a goal to establish an ideal autonomous targeting system within the given $1000 budget. Things to consider included finding reliable hardware, placement of the sensors for autonomous detection, placement of the Nerf Blasters, wiring, motor selection, and material selection. Fig 1 illustrates the initial design captured by the collaboration between the different engineering disciplines.

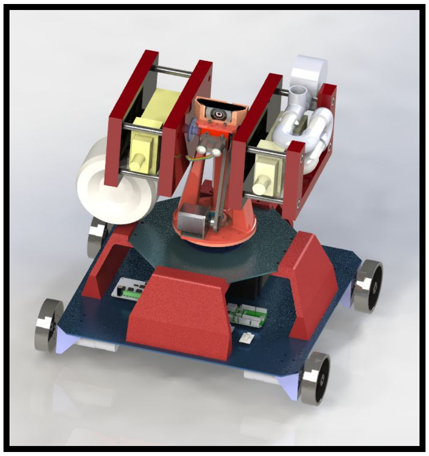

Fig 1. Initial Design

Ultimately, the components chosen by the ECE department included the Jetson TK1, the ATMega328P, DC motors with encoders, DC motors drivers, a Stepper motor, a Stepper motor driver, the Nerf Blasters, and two sensor modalities (Lidar Lite v3 and a webcam). These components encompass the primary functionality of the robot's electrical design.

IV. Mechanical Design

In the final design, the base of the robot and the top plate holding the turret system were created using multiple layers of laser cut wood in a mostly square orientation. The turret system itself was created using a 3D printer and bolsters two Nerf-blasters, one on each side. At the base of the turret mount is a DC motor (attached by an aluminum motor mount) connected to a chain (using hub sprockets) that is linked from the base of the turret mount to the top of the turret mount. The top of the chain is connected to a rotatable metal rod that connects the two Nerf blasters to create a tilting system. At the base of the robot, underneath the turret mount, a stepper motor is positioned and attached to a 3D printed ball bearing gear system to create a panning motion.

Navigation of the robot was designed to integrate differential drive using two rear-wheel DC motors (attached to the base by 3D printed motor mounts) supported by motorless wheels at the front of the base.

The overall size of the robot is roughly 21 x 18 x 22 inches and weighs about 25 pounds.

V. Autonomous Detection and Application

The computer science division created motion detection and facial detection algorithms to detect enemy targets for this project.

Utilizing these algorithms, the robot's optical mount moves at precise increments in order to successfully track these targets with the Nerf Blasters. Once the optical mount has appropriately centered an object utilizing the webcam, the Lidar Lite laser rangefinder reads the distance of a target and determines whether this target should be fired upon. Once the confirmation is given that the target is centered and the target is within the appropriate distance, a chosen Nerf Blaster fires at a given target based on their position at a given time. For closer targets, the Dart Blaster will be utilized, since accuracy at long ranges for the Dart Blaster is not always reliable. For longer range targets, the Ball Blaster will shoot at the enemy target. The language of choice used for the autonomous detection for this project was C++.

The computer science division also created a graphical user interface (GUI) for the manual navigation robot and provided a video stream from the Logitech HD Pro Webcam C920. The capability of seeing through the eyes of the robot and controlling its movements are interfaced through this computer application. The application is written in C++, with implementations from other services, such as MJPG Streamer. Fig 2. demonstrates the simple GUI that has been implemented in the design.

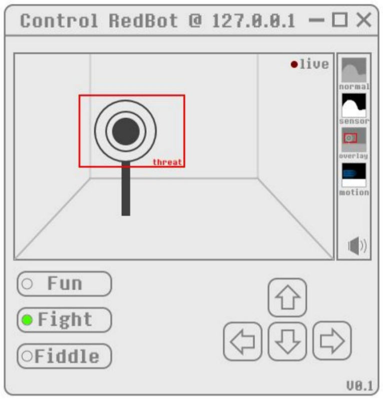

Fig 2. GUI for Video streaming and Robot Controls

VI. Jetson tki

The NERF battlebot uses an NVIDIA Jetson TK1 board dedicated for computer vision processing and sending control signals to the microcontroller through Bluetooth, mainly manual navigation input and detected target coordinates.

The TK1 comes with 192 GPU CUDA cores and 2GB of RAM[1], of which are an importance for this robot because of the heavy image processing that it does. This is a vast improvement over hobbyist level microprocessing boards such as the Raspberry Pi which in most cases are found to have a visible delay between image processing and a physical response.

The Jetson TK1 on this battlebot is compiled with an NVIDIA package called Jetpack 2.3.1 which contains an optimized linux operating system named L4T - Linux for Tegra and several other software packages essential to computer vision such as OpenCV, CUDA, and VisionWorks.

OpenCV and VisionWorks contain library packages that were imported and reconstructed upon for computer vision algorithms. CUDA is the platform that NVIDIA sources to its users to enable the usage of the onboard GPUs and work in conjunction with OpenCV and VisionWorks.

This robot runs L4T version R21.5 and was recompiled using kernel source code provided by NVIDIA to customize the kernel and allow for Bluetooth serial communication and a wireless connection using an AC-7260HMW mini-PCIe wifi card and coupled antennas.

Scripting was created to configure the Jetson on each boot to maximize usage of the onboard CPU's and GPU's in order to reduce workload and unlock the full performance potential that NVIDIA supplies.

The were a few concerns with the use of the Jetson TK1 on the Nerf battlebot. The TK1 is expensive, costing almost 1/5th of the allotted as-is budget, and, as previously stated, uses a linux environment, which is keen to the more advanced computer users. The operating system is very sensitive as most of the setup is done via terminal command line, leaving plenty of room for malfunction if configured incorrectly. Additionally, this board is capable of consuming a large amount of power from the batteries as it is rated for 12V and 4.8A.

VII. Communication

The purpose of communication is for the microcontroller to be able receive motor commands sent by the Jetson TK1, and for the Jetson to receive Lidar readings sent by the microcontroller. The Serial protocol is used to implement the two way communication needed between the devices.

The hardware chosen to perform the job is the HC-06 Bluetooth module which is attached to the microcontroller, and an Intel Dual-Band Wireless AC-7260 hooked to the Jetson TK1, which provides both Bluetooth and WiFi connection.

To establish a connection between the two systems it was necessary to utilize the Bluetooth protocol RFCOMM on the Linux powered Jetson TK1. RFCOMM provided an API used to pair and connect the HC-06. A configuration file /etc/bluetooth/rfcomm.conf was made which allows us the option to create a serial device under rfcomm and to automatically bind or connect to the device. Once a connection between the Jetson TK1 and the HC-06 is made a serial port we can use is made.

AT commands were used to configure the port settings on the The HC-06. The Baud rate was set, the pair code was set, and the device name was changed to prevent any connection issues with other

Bluetooth devices. The Jetson pairs the device via the Bluetooth GUI, and once the module has been paired the Jetson will not forget the HC-06. Connection between is finally established when the device file is opened by our software system.

Using the serial port on the Jetson is done by accessing the HC-06's serial device file. The device filename in Linux is /dev/rfcomm0, and the software operations on the device file are done using the POSIX terminal interface. This interface is needed because the software implementation is to be done in the C++ language and is standard to Linux systems. The advantage to using this interface is it allows many serial communication configurations to be made such as setting the baud rate, character size, parity checking, flow control, and timeouts.

The most important setting being the baud rate set 115200, which needs to be agreed upon by both ends. The character size was set to 8 bits to receive bytes on both ends. Also an important detail of the software implementation is the use of a timer before reading or writing to the device file, as opening it will cause the microcontroller to reboot. To receive data the Jetson will read data from the Serial port and store the data in a buffer. Buffer size must be specified therefore we chose a size that is twice the size of our formatted message length to avoid overflow. Similarly the transmitting of data is done using a write function

The Arduino Serial library is used to establish communication on the microcontrollers end. Once the correct baud rate set, the microcontroller can both receive and transmit messages. The Arduino uses the Serial Available function to get the number of bytes it is being set. It then transmits to the serial port with the print function. Special care must be taken when reading this data as newline and carriage return bytes will be sent.

Messages are sent in a formatted method both ways. Therefore the Jetson has a format to follow and the microcontroller knows what to expect. Each message coming from the Jetson specifies the instantaneous control over the motors, thus the microcontroller needs to continuously poll the Serial line for the robot to be responsive. The microcontroller also sends data in a formatted method to the Jetson which only contains the lidar readings.

VIII. Microcontroller

The robot features a standalone Atmel ATmega328p microcontroller, which is also found on the very popular Arduino Uno. The ATmega 328p hosts a total of 20 input and output pins, of which 14 are digital and 6 are analog. The six analog pins are also capable of being used as digital pins because of the analog to digital conversion that is built in. The 328p is the designated controller for the signals of the laser rangefinder, DC motors, stepper motor, and Nerf-blaster firing system.

The 328p was purchased as a standalone chip and it was delivered in an inoperational state because the version that was ordered had a missing bootloader. In order for it to become programmable, it had to be bootloaded with a functioning Arduino Uno via a connection on a breadboarded circuit. The 328p was connected to an Arduino Uno which in turn was connected to a computer running the Arduino IDE. The Arduino IDE includes a bootloading option in the menus, which was used to upload the bootloader to the 328p through the Arduino Uno.

The software for the 328p is being developed and deployed using the Arduino development environment. The Arduino IDE provides developers with a platform that combines the C and C++ programming languages. Signaling of the components connected to the chip are being made through a serial communication line via bluetooth. Code is uploaded to the ATmega chip using a USB 2.0 A to B cable connected from a computer to an Arduino Uno board. The microcontroller is then pried out of the Arduino board and placed onboard the PCB of the robot.

The ATmega328p was chosen because of its affordability and popularity in the consumer market of technology. A large and active community of hobbyists and plenty of documentation exists online for the 328p, reducing the difficulty in integrating hardware components and troubleshooting. Additionally, the 328p was a median in the decision of how many pinouts were needed for the robot.

The alternative options were boards that featured much to many pinouts, such as the ATmega2560, or were more expensive and had unappealing IDE's, like the TI brand of microcontrollers. The 328p was an efficient fit for the Nerf battlebot by using roughly 90% of the pinouts and leaving a small footprint on the final PCB design. Table 2. labels the most important components implemented in our design and their respective pinouts.

Pinouts	Component	PWM	I2C
RX/TX	Bluetooth	N	N
5, 6	Lidar Lite	Y	N
11, 12	DC Motor 1	Y	N
15, 16	DC Motor 2	Y	N
27, 28	DC w/ Encoder	N	Y
23, 24, 25, 26	Stepper Motor	N	N
18	Nerf Trigger	N	N
4	Servo Motor	N	N

Table 2: ATMega328P Pinouts with Components

IX. Camera Module

For use in computer vision algorithms, the Logitech HD Pro Webcam C920, the eyes of the robot, sits atop the robot inside of 3-D printed container of it's shape, moving in conjunction with the Nerf-blasters and rangefinder.

The webcam is capable of 1080p at 30 frames per second and has a viewing angle of 70 x 43 degrees[2] It comes complete with an H.264 video encoding software for bit compression, which is useful for video streaming over poor networks or peer to peer file transfer on a direct internet-free link. This feature is utilized as the robot delivers feedback to the user in the form of a video stream over an ad-hoc like network.

The C920 uses a six foot USB for connectivity, making it highly compatible - in a plug and play manner - with systems such as this robot. On this robot, the camera is connected to an NVIDIA Jetson TK1 for image processing via USB. The lengthy USB cable comes in handy as it provides slack when the turret system pans and tilts, allows for cable management, and makes a snug connection with the TK1 which sits at the base of the robot.

Originally, the camera of choice was a Raspberry Pi Camera Module. It was of comparable quality, had a much smaller form factor, and was more budget friendly. However, the Pi Camera used a short 15 centimeter ribbon cable, making its ability to be modular nearly nonexistent. Also, since the Raspberry Pi was found to be underperforming for the scope of autonomy implemented in this project, the camera module was quickly changed to the C920.

X. Lidar Lite

The Lidar Lite v3 was chosen as one of the sensor modalities to aid in autonomous tracking of enemy targets. The Lidar Lite allows for user interface through either I2C or PWM connections. The Lidar Lite is interfaced in the robot's design using a standard PWM wiring, a resistor coupled with the connection to the microcontroller, ATMega328P. Utilizing the Arduino platform to send a signal to the Lidar, a range value up to 40 meters[3] is returned. This range value assists the robot in order to properly aim the optical mount at a given target through adjustment of the motors. The Lidar is positioned at the top of the optical mount to acquire accurate readings of the enemy's position.

The laser rangefinder is used to prevent the robot from firing outside of course boundaries or within the friendly zones during battlebot competitions.

The benefit of using the laser range finder over other range detection devices such as sonar was its fast data polling rate and minimal interference potential. The only potential hazard with using the laser range finder is encountering a reflective countermeasure such as a mirror, which would throw of distance readings cause misfires or no firing of the blasters at all. The range finder, if pointed at an angled mirror, would reflect off the mirror until it hits a non-reflective surface and return a distance that is much farther than the mirror actually is.

XI. DC Motors

Multiple DC motors are featured within the robot's design. One motor is utilized for the tilting the robot's optical mount towards an acquired target. Two of these motors are used for manual navigation. The motors chosen operate at 12V and 1.3A of continuous current and each are equipped with encoders.

A motor with these specifications was required on the optical mount, since it has enough current to support the weight of the Nerf Blasters that is applied to the turret system. This DC motor must use an encoder in order to track the position of the turret system. A motor driver that receives the encoder readings is used to track the current position of the DC motor. In order to interface the motor driver with the ATMega328P, an NXT cable was stripped to reveal the 5V, GND, CLK, and DATA wires. Fig 3 shows an image of the stripped NXT cable as well as the labeled wires of the cable. The I2C pins located on the ATMega328P are connected to the CLK and DATA wires that were exposed from the NXT cable. Then, addresses being sent from the motor driver were used to move the DC motor to the encoder target position. These addresses were also used to control the speed and direction of the motor.

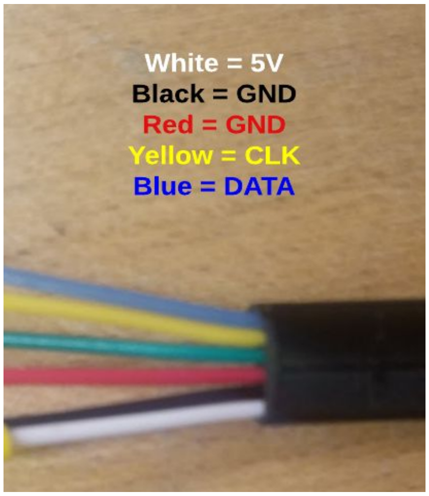

Fig 3. I2C Connection with NXT Cable

The two motors that are used for manual navigation are interfaced through a Cytron DC motor driver that allows a maximum continuous current of 10A and a maximum voltage of 25V. The motor driver allows two channels, so both motors can be connected. Utilizing an analog and a digital pin from the ATMega328P, the motors are able to move at different speeds and in both directions. These motors are then programmed using the microcontroller for differential drive.

XII. Stepper Motor

The stepper motor allows the complex turret system to pan in as little as .9 degree increments for accurate positioning. The stepper motor operates at 3V and 1.07 A. This motor is capable of rotating the gear connected to the base of the optical mount, since the Nerf Blaster's weight only greatly affects the tilting of the turret system. Although another stepper motor was initially considered in the tilting of the turret system, the second stepper motor proved to be unable to support the entire weight of the system.

These stepper motors are connected to drivers that utilize the A3967SLB chip. The two logic inputs, MS1 and MS2, for the A3967SLB chip allow microstepping up to an eighth of a step. Both the MS1 and MS2 pins as well as the step and direction pins are connected to the ATMega328P. The speed, step increment, and direction can be controlled seamlessly through the microcontroller. Although more powerful stepper motor drivers were considered, since these drivers only support up to 750mA/phase, the budget prevented the team from acquiring a motor driver that could microstep and support a higher current.

XIII. Nerf Blasters

The final task in the robot system is firing the NERF blaster at detected targets. Our system uses two NERF blasters to ensure enough ammo and allows the robot to differentiate between the two blasters for different range and desired accuracy.

A. Rapidstrike CS-18

The first NERF blaster uses a dart ammo system and can fire darts up to 55ft. This NERF blaster uses a total of three small 3V motors to automatically fire darts from a pull of a trigger. The first two motors are lined together to create a flywheel for the darts to be pushed through. The third motor is a pushing mechanism that allows each dart to be pushed to the flywheel. In order for this NERF blaster to be fired automatically, the Blaster was opened up and stripped away all the safety switches that came with the NERF manufacturer.

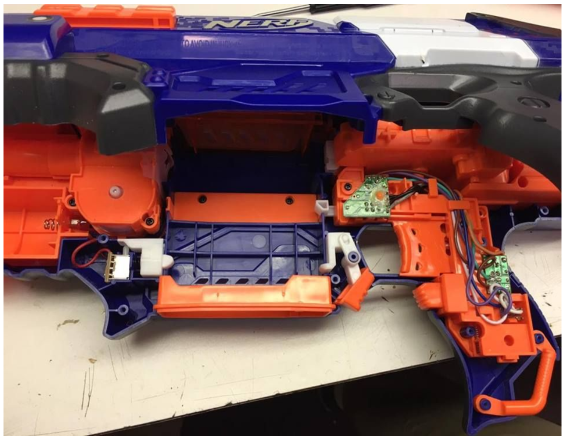

Fig 4. Inside of the Rapidstrike.

In order to activate this gun remotely from a microcontroller, a logic level MOSFET circuit is used. With the right delay and a 5V source, the MOSFET can power on and off the pushing motor forward to feed the darts into the flywheel.

B. Rival Zeus MXV-1200 Battle Gun

For the second NERF blaster, a ball ammo blaster is used. This NERF blaster was chosen due to its long range of 75ft. For the function of this NERF blaster, it uses only one flywheel. The mechanical engineering team provided a modification to the gun for a longer magazine and uses a 12V fan to push the balls through a tubing device into the flywheel and a servo motor is used to be the controlling mechanism from the microcontroller to allow only a few balls to be shot at a time.

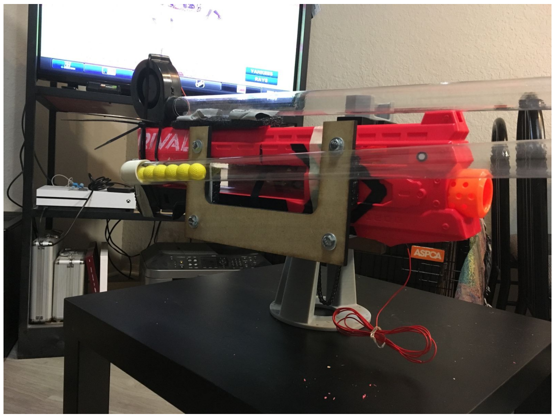

Fig 5. Modification of Rival Zeus model.

XIV. Power Regulation

The robot varies from different voltage ratings from 3.3V to up to 12V. The use of power regulation is needed. For each system requiring a different voltage rating, LMXX series power regulators were used with heat sinks that allow the different systems to be powered. These devices were considered for their voltage step down at a high efficiency and stability for the CPUs and MCUs. Each regulator contains a simple pinout of In, Ground, Out. The In and Out terminals are shared with coupling capacitors to stabilize any AC noise that the regulators may make. The 3.3V regulators will be able to power the flywheels of the nerf blasters. The 5V regulator is used for the microcontrollers, the bluetooth module, the lidar and the small servo motor. There is a 12V regulator to keep the power for the Jetson TK1 at a constant voltage.

XV. Battery Selection

For battery selection, it was important to find the right capacity that could power all the components at the right ratings. Power for all components were calculated and tallied up shown in Table 3 listed below.

Component	Qty	Voltage (V)	Current (mA)	Power	Mostly On/Off
MCU	1	5	46.5	.46	ON
Stepper	1	3	3400	10.20	OFF
NERF1	1	6	1500	9.00	ON
NERF2	1	9	1500	13.5	ON
Jetson	1	12	2500	30.0	ON
Lidar	1	5	130	0.65	ON
DC.M	3	12	4800	57.6	OFF
				Total Power	121.41

Table 3. Full power calculation.

Most of the current draw is from the motor systems, therefore to provide sufficient amount of current, a sealed lead acid battery is used. Sealed lead acid batteries contains the best chemistry type for high current and does not require complex charging. The downside to the sealed lead acid battery is its heavy weight and bulky size which was accommodated for the design.

In order to calculate the right battery capacity the following equation was used.

$$\text{Battery Life} = [\text{Battery Capacity(mAH)} / \text{Load Current(mA)}] * 0.70 \quad (1)$$

Therefore at 12V, 5aH this battery will be able to last for 15 minutes per full charge only costing the project $15 per battery.

XVI. PCB Design

For the PCB design of the robot, it was important to make any breadboard connection secure as possible. Therefore for any embedded circuit that could be soldered, it was included in the final design.

The PCB contains a microcontroller and all the connections needed to control the various systems, a power regulation circuit and a stepper motor driver mount to avoid any wiring. EagleCAD was used to draw the schematics and board design before being sent off to a PCB manufacturer to be printed. In the figure below is the final PCB design used to power and communicate with the various components of the robot.

Each component to be placed in the schematic had to be fetch from certain libraries to ensure that the part was available for purchase and will be properly soldered onto the board. The schematic is where the initial design is formed adding any electronic elements or embedded chips to the design. All the wiring must be labeled and connected correctly for a completed circuit.

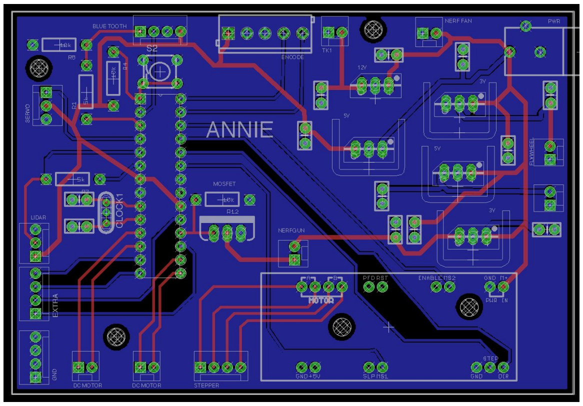

Fig 6. PCB Design

XVII. Vendors

Part acquisition in majority was done through online vendors RobotShop, Sparkfun, Amazon, and AndyMark. There was no particular preference for any vendor.

The PCB design was quoted through several vendors, but was ultimately manufactured by 4PCB and was assembled and soldered by the our battlebot team.

XVIII. Conclusion

The design of the robot system highlights and demonstrates the technical skills that each team has acquired throughout the course of the engineering program. Every aspect of the current system's design has been analyzed thoroughly with the hopes of developing a successful project. Working cross-discipline with other majors has allowed the Electrical and Computer Engineering team to gain experience that is closer to the real world. Dealing with various team members from different disciplines can be fairly common in the engineering work environment. This opportunity of learning and engaging with other individuals has been truly enlightening for the Electrical and Computer Engineering team, as many new ideas have continued to develop throughout the project's planning phase.

The key purpose of the Nerf-Enabled Battlebot is to be able to autonomously detect, aim, and fire at an enemy target. This opportunity to build the proposed system has allowed the entire team to be exposed to new concepts regarding computer vision and autonomous robot systems. Given the opportunity to learn and develop regarding electrical components and how they affect the autonomy of the system, the Computer and Electrical team have gained a better understanding to robotic systems.

Acknowledgements

Our team would like to give a tremendous thanks to our sponsors, Lockheed Martin and our representatives, Kenny Chen and Jonathan Tucker.

We would also like to give a special thanks to Don Harper, who helped us at a critical time in our design.

Finally, we would like to give thanks to the Mechanical Engineering Team and the Computer Science team for their dedicated hard work. Without their collaboration, this robot would not have been made possible.

References

[1] NVIDIA Cooporation (November 2014) "Jetson TK1 Development Kit Specification," Retrieved from http://developer.download.nvidia.com/embedded/jetson/TK1/docs/3_HWDesignDev/JTK1_DevKit_Specification.pdf

[2] Logitech (2011) "Logitech HD Webcam C920," Retrieved from https://www.arp.nl/webmedias/datasheet/575183859ce969320b1a2beb.pdf

[3] Garmin Ltd. (September 2016) "Lidar Lite v3 Operation Manual and Technical Specifications," Retrieved from http://www.robotshop.com/media/files/pdf2/pli-06-instruction.pdf