University of Central Florida

College of Engineering & Computer Science

Department of Electrical Engineering & Computer Science

Senior Design II

Autonomous TankBot

Group 1

Anthony Speake	Electrical Engineering
Stephen Greene	Electrical Engineering
Emmanuel Sotomayor	Electrical Engineering
Conan Wilson	Electrical Engineering

Table of Contents

List of Figures

List of Tables

Appendices

List of Acronyms

ACK .... Acknowledge
ADS .... Advanced Design System
BJT .... Bipolar Junction Transistor
CAD .... Computer Aided Design
CECS .... College of Engineering and Computer Science
CNC .... Computer Numerical Control
CPU .... Central Processing Unit
CPW .... Co-Planar Waveguide
CSI .... Camera Serial Interface
CV .... Computer Vision
DC .... Direct Current
DDR .... Double Data Rate
DIP .... Dual Inline Package
EDA .... Electronic Design Automation
ESR .... Equivalent Series Resistance
FPS .... Frames Per Second
GDB .... GNU Project Debugger
GPIO .... General Purpose Input/Output
GPL .... General Public License
HD .... High Definition
IEEE .... Institute of Electrical and Electronics Engineers
IC .... Integrated Circuit
ICDI .... In-Circuit Debug Interface
I/O .... Input/Output
IP .... Internet Protocol
JTAG .... Joint Test Action Group
LAN .... Local Area Network
LAW .... Lethal Autonomous Weapons
LED .... Light Emitting Diode
LIDAR .... Light Detection and Ranging

MCU .... Microcontroller Unit
MIPI .... Mobile Industry Processor Interface
MMC .... Multi-Media Card
MMU .... Memory Management Unit
MOSFET .... Metal-Oxide-Semiconductor Field Effect Transistors
OS .... Operating System
PCB .... Printed Circuit Board
POSIX .... Portable Operating System Interface
PWM .... Pulse Width Modulation
QFN .... Quad Flatpack No-leads
RF .... Radio Frequency
RPM .... Revolutions Per Minute
SDIO .... Secure Digital Input Output.
SMD .... Surface Mount Device
SoC .... System on Chip
SPI .... Serial Peripheral Interface
TCP .... Transmission Control Protocol
ToF .... Time of Flight
UART .... Universal Asynchronous Receiver-Transmitter
UCF .... University of Central Florida
UDP .... User Datagram Protocol
USB .... Universal Serial Bus
WLAN .... Wireless Local Area Network

1. Executive Summary

Since the dawn of the microchip the technologies surrounding humans has become more and more integral to allowing mankind to experience the fullest of the human experience. Machines make our furniture, harvest our crops, and do some hundred-thousand other tasks that we have deemed too menial or degrading for a human to have to do. However, all over the world, military, law enforcement, and even civilian lives are put at risk every hour of every day. From bank vaults to battlefields, we trust only human beings to be the last line of defense when confrontation arises, and threats need to be assessed and dealt with in a timely manner.

New technology is being developed every day to take more and more of the responsibility out of the hands of human beings and into the more precise, unfeeling, unerring hands of machines. New age electronic devices have all the capabilities to identify, target, and process images and objects at many times the rate of the human brain, all while being un-phased by the less than helpful feelings that plague humans like fear for one-self, indecisiveness, or any other emotion that prevents us from acting when we are most needed to.

The goal of our project aims to combine the emerging technological field of computer vision with the efficient motion and decision-making power of today's machines to eliminate the need for human combatants or first responders in the day to day operations and missions required to keep society functional and our enemies at bay. Some of these missions include but are not limited to: patrol and identification of targets, guarding a location from hostile forces, and even ranging to seek and destroy missions in some of the most dangerous places in the world that we are constantly risking human lives to protect. These tasks are eminently solvable by modern machines, with the processing power and agility of decision making allowed by the parallelism in the technology of today.

To meet the goal of being able to remove humans from these very dangerous situations, we have developed an autonomous wheel-based robot, henceforth called TankBot, which will in the future be able to replace soldiers on the frontline, guards in high security areas, and even law enforcement officers when the need arises. To be able to do all of this though, the robot was designed for a couple of important objectives like being able to move quickly and independently of an operator over rough terrain. Also, the TankBot can identify not the shape of a person. Additionally, it can track a moving target and follow it with the option of compensating for that motion and then targeting and firing the on-board weapon at the target. TankBot must accomplish all these tasks, sometimes simultaneously, without any human intervention beyond the command of what mode it should be in.

The following document provides a detailed description of the complete design process used in senior design 1 and senior design 2. We will start off with a more in-depth description of the project and motivation further than was is described in this section. Then the document will go over the research phase of senior design one including an overview of similar projects and what our group learned and could apply from each of the previous projects to our project., as well as detail the choices we made for each of the parts both hardware and software and details some of the most important technologies implemented within the project. Next, we will discuss some of the relevant standards put forth by IEEE and other standard developing organizations and discuss the impact these standards will have on our project as we continue forward. The hardware and software design will be covered in two separate sections such that each part gets its due time in the spotlight. After the design is discussed we will begin to talk about our prototyping process as well as specifics about the testing environment and processes that was implemented for the testing. Evaluation of TankBot will discussed and benchmarks set so that at the end we can determine if the project was a success. Towards the end of the document you will find the more administrative content, including our projected milestones, budget, and schedules.

2. Project Description

The objective of this project is to design and build a fully autonomous mobile robot that can correctly identify and neutralize targets.

The function of the project is then to increase security in required areas while limiting human intervention so that the relative safety of friendly personnel can be maintained. Project features stem from comparison to autonomous vehicles that already perform this task (drones, automated turrets, and such) as well as features we believe are missing from these products. The bot should remain lightweight, low cost, relatively low power, accurate, and implementable within the time frame provided.

2.1. Project Motivation and Goals

The motivation of this project is to remove the risk to friendly personnel associated with engaging a hostile enemy. In hostile environments, such as combat or high security zones, troops/guards must have some downtime to keep their bodies and minds at the highest possible condition. As such, rotating shifts of troops to pose as lookouts for hostiles that might engage the camp is not uncommon. The TankBot can act as a sentry, assisting in these operations to observe and detect enemy presence and increase security while at the same time decrease the need for human intervention and thus decrease the danger posed to friendly forces.

One of the many motivations for our project was the outcome of the Pulse nightclub shooting on 12 June 2016. Pérez-Peña, Robles, and Lichtblau of the New York Times writes:

> “Many questions persist about those three hours at the blood-drenched Pulse nightclub, and about how law enforcement handled the crisis on June 12. Orlando police officials have been peppered with queries from the public, survivors and the news media about whether they should have confronted the gunman sooner and whether any of the victims were shot by the police.” [1]

The level of skepticism and criticism over the decisions made that night highlight one of the key motivations for this project: TankBot cam make swift decisions about who the threat is and neutralize them without harming civilians. To that degree, it isn’t odd for people to question whether a team of autonomous, gun-wielding robots could have better handled the situation than live people who are subject to their own doubts, uncertainties, and indecision. Of course, in the specific light of the Pulse nightclub shooting, officers’ lives would not be at risk and there wouldn’t be any skepticism about who was shot and what they were shot by. The question could still be raised as to whether TankBot is necessary or whether it has only niche use rather than being a multipurpose machine implementable in different situation across the globe.

One of the other key motivations is the recent developments in autonomous, military-use technology. Consider DoDAMM's Super aEgis 2:

> "If there's one place you don't want to be caught wandering around right now, it's the demilitarized zone that separates North and South Korea. Especially since South Korean military hardware manufacturer DoDAMM used the recent Korea Robot World 2010 expo to display its new Super aEgis 2, an automated gun turret that can detect and lock onto human targets from kilometers away, day or night and in any weather conditions, and deliver some heavy firepower." [2]

The demilitarized zone is an area between North and South Korea that is a result of the violence of the Korean War. It is constantly manned by military forces of both sides of the conflict to maintain security in the border region between North and South Korea. The Super aEgis 2 does not completely remove military personnel from the area but does enhance security with its ability to detect, target, and then eliminate foes from a relatively safe distance. As such, it serves as one more thing to help reinforce the safety of the many military combatants there. So, this situation, again, highlights the usefulness of autonomous military-use technology in the real world. Now, TankBot is not limited to serving as a stationary autonomous turret. It can also move, and move quickly, which offers a great improvement to the Super aEgis 2. However, given the amount of funding and time that the Super aEgis 2 has been provided, TankBot is a small-scale version with much less effective range and coverage capability. As such, future developments of TankBot are encouraged to replace Super aEgis 2 with a mobile variant. Whereas this might not be necessary for the demilitarized zone (where a stationary, autonomous turret fully serves its purposes well), it does extend the range of application of autonomous turrets in the future.

Consider another situation: friendly military combatants having to enter a dangerous zone on a seek and destroy mission against enemy forces. The question that stands is whether a team of TankBot's, or even a single unit, could better handle the situation. Now, while this is incredibly debatable there are many key advantages to using TankBot's over military personnel:

- Human lives aren't at risk
- Human emotional inhibition does not directly affect TankBot
- Inaccuracies from humans are eliminated
- TankBot can make thorough, well-thought-out decisions quicker than a human

These advantages serve as key motivations for military uses of TankBot. As such, the team views the creation as a necessity for human security and protection moving forward into the future of conflict the world over.

Another motivation the team recognizes is the recent spending in autonomous weaponry:

> "Defense officials say the weapons are needed for the United States to maintain its military edge over China, Russia and other rivals, who are also pouring money into similar research (as are allies, such as Britain and Israel). The Pentagon's latest budget outlined $18 billion to be spent over three years on technologies that included those needed for autonomous weapons." [3]

In 2015, Pentagon spending reached nearly $600 billion. As such, $18 billion represents a very small percentage of the total, but is not in any way insignificant. The Pentagon's investments in these technologies highlights the beginning of a military industry shift towards these technologies and indicates that these technologies are not only plausible in the future but also deemed necessary in the present. After all, a heavy amount of investment would not be made into a field that people didn't believe was going to be a necessary field of innovation in the not to distance future. Although it may take many years for a complete re-outfitting of the military to occur, many people following military technology development do see a new age coming, one where the modern soldier is replaced by an even more modern autonomous robot. What was once an imaginative creation of science-fiction media, is now an oncoming reality that can promote and enhance defense of a nation.

The goals of this group are to delve into research concerning autonomous, motive, turret robots that utilize computer vision to identify and differentiate allies from enemy forces. The group has improved on other past designs that have been researched and attempt to make a high-quality functioning robot that is cheaper than most other designs that have been employed in the past and that works for a small-scale situation due to our limited time and budget. The robot itself is small in stature and light-weight but accurate and quick in its decision-making processes. An airsoft gun was employed in place of a conventional firearm to follow demonstration restrictions which would certainly not allow an actual gun to be fired on campus during demonstration. As such, the product created is a small-scale replica of what the intended product could be in the future. Research was performed on computer vision, airsoft gun selection, motor selection, power efficiency options, mechanical framework options, electrical component options, wheel selection, camera selection, battery selection, and similar designs to increase the quality of the bot. All in all, the results should be a small, unmanned, autonomous vehicle that can find enemy targets and fire at them, while withholding fire against friendly targets. The vehicle can communicate with a nearby screen to show the decision-making process of the bot, as well as what the bot sees. It can communicate with a controller to initiate its 'standby', 'follow target', 'patrol', and 'go to location' actions. To that degree, humans still exert some level of minor control over the robot in the macro sense. These actions are discussed in the relevant upcoming sections of the report.

2.2. Objectives

The following objectives are given as the bare minimum needed to achieve a successful, autonomous bot:

- Shall identify and be able to differentiate a human target from the surrounding environment as well as determine the distance to the target
- Shall move forward and reverse as well as make turns
- Shall aim and shoot an airsoft gun to respond to hostile infiltration of a secured area or as a process to eliminate a target.
- Shall follow a target but not shoot at a specified distance
- Shall accept commands via wireless link from a remote device

These objectives were carried out through research and design involving the following concepts, whose study and implementation provides further educational and experiential group objectives:

- Computer vision
- Microcontrollers
- Motor control
- Battery regulation and control
- Electronic airsoft guns
- LiDAR
- Wi-Fi
- Voltage supervision

2.3. Requirements Specifications

In engineering, a requirement is a singular documented physical or functional need that a design, product, or process must satisfy to be considered successful. The requirement specifications can also be negotiated between the stakeholders, or client and the engineers working on the project. Thus, the requirement specifications are a set of documented requirements that the design must satisfy. Requirement specifications are also key to the verification process since all tests should trace back to and verify specific requirements. The requirements created ensure that a high-quality, effective product is produced according to both parties.

The following section describes some the most important decisions behind the various requirement specifications of the project as well as the expected, final product turn-out. It can easily be said that the final goal of this project is to meet and/or exceed the minimum requirement specifications listed below.

2.3.1. House of Quality

The house of quality is a tool to assess the interdependence and tradeoffs of different levels of the design requirements. This is shown in Table 1.

A house of quality breaks down these tradeoffs in a relatively easy to understand format. Positively impacting requirements are denoted by a + while negatively impacting requirements are denoted by a -. Positively correlated requirements have a ↑ and negative a ↓. The stronger the correlation the more arrows are present.

The first column details the minimum design requirements while the first row suggests provisions to meet these requirements. The minimum requirements are:

1. To fit within a designated budget (cost)
2. To make the device as low power as possible within the time limits (power)
3. To achieve some level of smooth motion within the bot (smooth motion)
4. To aim the gun effectively (accurate aim)
5. To acquire identity of targets effectively (accurate target identification)

As for the provisions to meet these requirements (parentheses below link the provisions to the minimum requirements listed above):

- Power efficient (2)
- Effective Range (4)
- Dimensions (3)
- Hit Rate (4)
- Overall Cost (1)
- Weight (3)

As mentioned before, the table below shows the relative trade-offs and advantages of the proposed provisions as they relate to the minimum requirements of the project.

		Power Efficient	Effective Range	Dimensions	Hit Rate	Overall Cost	Weight
		+	+	-	+	-	-
Cost	-	↓↓	↓	↓	↓↓	↑↑	↓
Power	-	↑↑		↑	↓	↓↓	↑
Smooth Motion	+	↓	↑	↑↑		↓	↑↑
Accurate Aim	+	↓	↓	↓	↑↑	↓	
Accurate Target Identification	+	↓	↓	↓		↓	
Targets for Engineering Requirements		< 400 W	30 ft.	< 2 $ft.^3$	25% target hit rate	< $4000	< 50 lbs.

Table 1 *House of Quality*

The following list comprises the requirements this project must be able to achieve to be considered a success:

2.3.2. Hardware Requirements

- Accurate distance range measurement minimum: 20ft.
- Minimum target hit rate: 25%.
- Minimum effective range: 30ft.
- Minimum run time: 30 minutes
- Real-time position tracking of gun turret.
- Omnidirectional movement.

2.3.3. Software Requirements

- Remote controllable.
- Stream video to remote device.
- Both autonomous and manual control modes.

- Display captured image from camera to display.
- Target identification/differentiation.

The physical systems of the robot in addition to the software used to run them were designed and developed to satisfy these requirements.

2.4. House of Quality Analysis

Power efficiency was one of the hardest requirements to satisfy as it is quite an open-ended requirement. However, the goal we have set for the project was to keep it drawing below 400 W. This power level was chosen via rough calculations of the power required by each individual section of the project aggregated. Work was done to lower the power of the bot to the lowest possible if possible, to increase the run time of the bot. Lower power means more expensive components though because cheaper components often don't minimize power in their designs. Smoother motion may also be imposed by the neglect of higher-power, yet better performing pieces. An accurate aim may also be sacrificed by the cost because less technologies may be implemented to improve the aim. Accurate target identification is restricted by the lower power because multiple on-board cameras and CV systems used for comparison and relation are neglected to reduce the overall power.

The effective range being larger means more power was needed to drive the high-quality components used to increase the effective range. Also, a larger effective range can be a challenge for accurate aim of the bot because distance from the target increase the chance of random variables like wind messing with the accuracy, so increasing the range while maintaining sufficiently accurate aim was a challenge that we must overcome to be successful. Furthermore, as the effective range of the bot is increased, the CV has a less clear image of the target, making target identification significantly harder. As extending the effective range of the bot improves the overall quality of the bot, it was the utmost importance to balance the largest possible range within the other design requirements.

The dimensions of the robot's core being tighter imposes a risk to the cost since smaller and smaller implementations of an otherwise common device can often be prohibitively expensive. Furthermore, if the dimensions are made too small, the device might have a problem aiming well. This also imposes on the design space of the target identification system since a smaller camera was used, instead of a large, possibly better-quality camera. The dimensions of the robot remain important for ease of transportation and for the robot's own ability to move. A large, bulky bot would not move as well as a smaller, lightweight bot.

The hit rate of the bot is one of the most important quantities related to the project as a hit highlights the bot's success, while a miss highlights its failure. A higher hit rate would impede the cost, however, as a costlier gun and more subsystems to improve aim would most certainly raise the cost of the final design. As for the power of the bot, adding more facilities to aid in aiming would only drain more power from the battery, so some compromises would have to be made. If the gun can fire two rounds per second and the target is identified at thirty feet to be hostile, with nearly a second and a half of response to initial shot (counting human reaction time at the time involved in turning to flee) running away from the bot at the average unloaded human running speed of 10 feet per second, approximately 1 in 4 shots must hit the target. This minimum rate of fire must be maintained and preferably improved to respond well to targets with premeditated knowledge of the bot's existence and location.

The overall cost is another aspect of the project that was very difficult to maintain. One of the largest drawbacks is keeping minimal power consumption as objects that consume less power tend to be far more expensive. Furthermore, the cost limitations impose upon the smooth motion of the bot since extra faculties involved in smoother motion would only increase the cost of the project. A more accurate aim would mean a better gun and more faculties involved in improving the aim, which also, in turn, increases the cost of the bot. Since OpenCV is being used for target identification, instead of a perhaps more sophisticated, yet costlier program, the cost imposes upon the target identification. Further minimizing the cost of the robot was not sought since any extra decrease in the cost can mean further improvement to the quality of the project. As such, the overall cost stood as a limitation for how much can be spent of the project, rather than a quantity that must be further reduced.

Keeping the weight low implies using smaller components which can often cost more, so this does impose a restriction on the weight. However, despite this restriction, maintaining a minimum weight won't be too hard since most of the weight on the bot was from the battery, frame, gun, and wheels. As such, this requirement is quite easily met and can easily be minimized, so focus won't be set to minimize as much as possible, when extra faculties which increase the weight can be implemented which also increase the quality of the bot.

2.5. Overall System Design

The following figures illustrate a very top-level view of the overall hardware design and software design that was implemented for the TankBot to achieve its functionality and minimum objectives. For further details involving these figures, please reference Section 5 for Figure 1 and Section 6 for Figure 2.

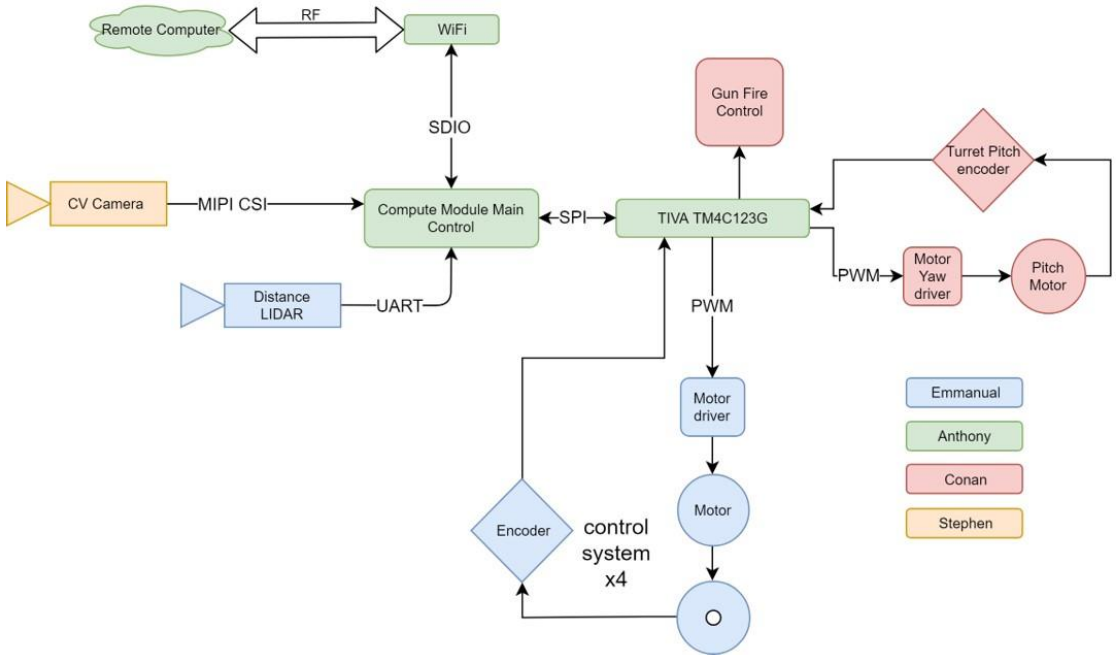


Figure 1 Overall Hardware Control

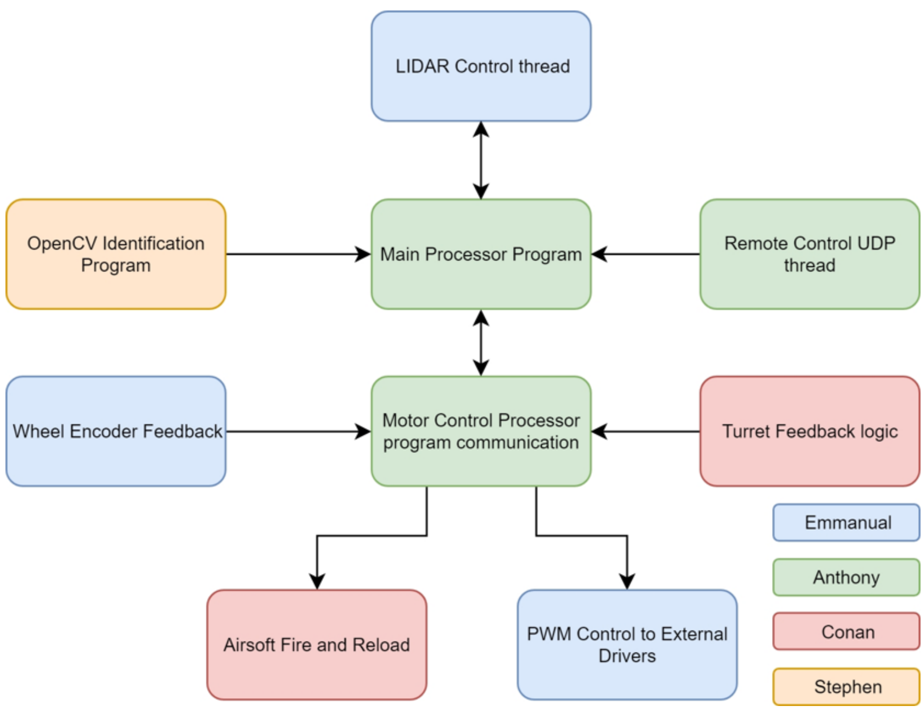


Figure 2 Overall Software Control

3. Research Related to Project Definition

This section will detail most of the research that was done that is relevant to the final product that is desired. Trade-offs and comparisons tables will be made between the products we considered to highlight why we made the choices we made in the final product. The requirement specifications and standards have been kept in mind in making the decisions in regards to which products we used in the final design. Sometimes inferior products are chosen because their price point is attractive enough or though their exceeding performance expectations is unnecessary to the design of the bot itself. Superior products are obviously the most important to the industry at large, so effort has been made to select the most superior product that achieved the requirement specifications without overdoing the task at hand. Since a higher performance of the final product is desired sometimes decisions contrary to the requirement specifications are made as the team decided that the final balance of the products minimum requirements is worth the contradictory decision.

3.1. Existing Similar Projects and Products

Below have been listed several products and projects already developed and completed which were closely related to the project we tasked ourselves with accomplishing. The goal in presenting these projects and products was to examine the inspiration behind the project as well as the realistic expectations for our project, in addition, each of the projects has been examined for points that we feel that we could have improved on or remarkable things that we could have emulated. This section will comment on the baseline feasibility of this project. Without these projects and products, it is questionable whether this project would even be possible to achieve. So, a nod has been given to similar projects, both those that are more advanced and those that have been less ambitious, to relate the intentions behind undergoing the tasks that follow. Immediately below, a project will be encountered which is far more advanced than TankBot due to the budget and far more relaxed time constraint that this project was allowed. Also, multiple projects of a similar time constraint and budget will be encountered via past students' senior design projects to show what history has provided to us in terms of inspiration and remark on what improvements can be made to the pre-existing projects.

3.1.1. Samsung SGR-A1

The SGR-A1, shown in Figure 3, is an autonomous sentry turret made by the South Korean company Samsung for use by the South Korean government, Samsun is most often known for their consumer electronics products, so this is a little out of the wheel house for this company. The turret features a 5.56mm light automatic machine gun and a 40mm multiple grenade launcher, this last feature will not be included in our final design of the project.

The turret was initially funded by the South Korean government in 2003 with first prototypes being made in 2006, it was initially designed to replace human guards along the Korean demilitarized zone. [4]

This project made use of a combination of sensors to get the most accurate picture of what is going on around it. We made use of this idea by also having multiple systems in place that could be used in conjunction to detect, track, and identify targets. One major difference between this and our project is that the Samsung SGR-A1 is a stationary mounted turret system, whereas our project necessitated a mobile platform to be successful. This means that while insights could be made from the turret and detection point of view we were not be able to take any inspiration about navigation or mobility from this project.

Figure 3 SGR-A1 Autonomous Sentry Turret

3.1.2. Autonomous Target Recognition System Fall 2016-Spring 2017

The Autonomous Target Recognition System, shown in Figure 4, was the senior design project of Clayton Cuteri, Corey Nelson, Kyle Nelson, and Alexander Perez during the Fall 2016 - Spring 2017 semesters. The project consisted of a nerf gun mounted on a mobile turret designed for a competition conducted and sponsored by Lockheed Martin. The system would recognize targets using CV technology with blob detection and fire the nerf gun at the target.

The inspiration for our project this year came in part from seeing past groups attempt to solve the same problems we are attempting to solve. This robot especially made us realize how interesting of a challenge this project could be.

While many parts of this project were well designed and built, there were several systems we felt that we could improve upon though our own design and development process. One of those systems is the Object Detection system, where this group decided that a standard webcam was the best choice for what are essentially the “eyes” of the machine. Later in the document we will discuss why we do not think a webcam is the best solution, as well as present a couple of other options and weigh the benefit of each before coming to a decision.

Figure 4 Autonomous Target Recognition System [5]

3.1.3. Self-Targeting Autonomous Turret System (STATS) 2014

STATS was the senior design project of Elso Caponi, Michael Lakus, Ali Marar and Jonathan Thomas at the UCF in 2014. As the name suggests, the project consisted of creating an autonomous mobile turret system very similar to the one that we have envisioned. The project implemented Arduino technology, computer vision software, wireless communication hardware, an audio alert system, and laser technologies. An image of the project can be found in Figure 5.

Areas of similarity included the use of computer vision software used to track and detect targets, as well as the idea of having the turret mounted on a mobile platform. Also, this project made use of a webcam connected to a Wi-Fi transmitter to be able to show on other systems what the robot was actually "seeing". These similarities allowed us to make use of the pre-existing technology such that we were able to improve upon the design in our own project.

Figure 5 Self-Targeting Autonomous Turret System

3.1.4. Autonomous Chasing Robot 2015

The Autonomous Chasing Robot was the senior design project of Bryan Diaz, Victor Hernandez Salomon, Khanh Le, and Luis Sosa at the University of Central Florida in 2015. This project was concerned only with tracking and following an object physically rather than also identifying and firing on said object. An image of the project can be found in Figure 6.

This project, conversely to the Samsung SGR-A1 project, was all about the navigation and mobility rather than targeting. From this project we could compare our tracking software, and mobility hardware specifically while drawing inspiration from the compact design and economy of motion. A point of note however is that this system seems to be using a relatively stock and, in our view, inefficient 4 wheel steering system, meaning the robot only had 2 degrees of freedom of motion. We would take this information going forward when deciding on the locomotive systems for our project.

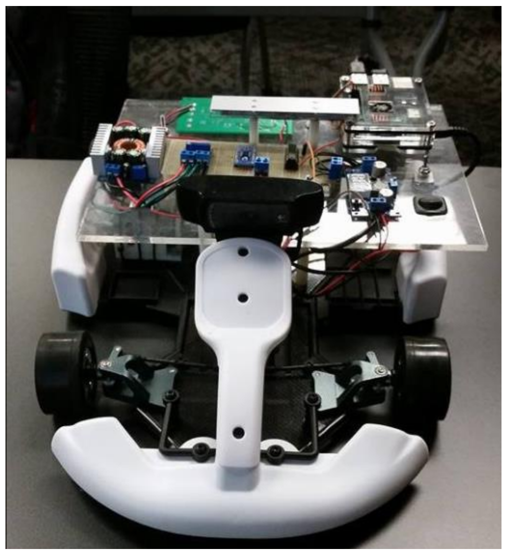


Figure 6 Autonomous Chasing Robot

3.1.5. S.H.A.S Bot

The S.H.A.S. bot was a project in the Fall of 2012 that was designed by Daniel Lanzone, Mike Roosa, Ryan Tochtermann, and Charlie Grubbs. Its primary goal was as either a surveying machine or as a lookout to detect intruders or other people in its vicinity. Unlike the last project this project was aiming to identify intruders in a manner very similar to what we are looking to do with our project, they however were content with just acknowledging that there was a presence rather than actually identifying what that presence was by use of computer vision, so we will be unable to gain any insight on our project in this area. This group chose to use an ultrasonic speaker and microphone to listen for things nearby by sending out sound waves undetectable by human ears and listening for the sound to bounce off of something and come back, then taking the time between sending and receiving the pulses and comparing it to the speed of sound to determine how far things are away from the robot. This was an approach which we had not considered until this point in our research, for this reason, we thought this project was a noteworthy inclusion in our report. In the end we decided that the computer vision route would be the best for our application rather than a system for listening for nearby threats, though the two processes could overlap. An image of the project can be found in Figure 7.

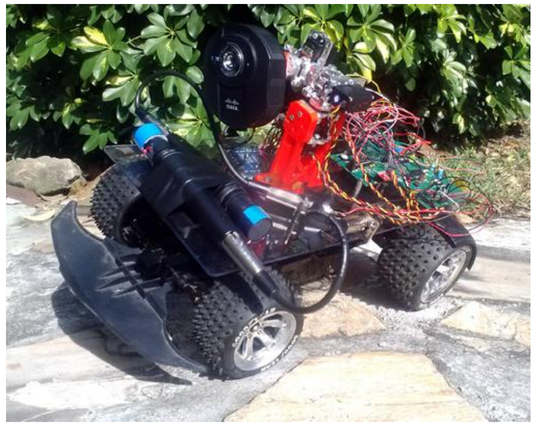


Figure 7 S.H.A.S. Bot

3.1.6. DoDAMM's Super aEgis 2

DoDAMM's Super aEgis 2 is an autonomous turret that uses thermal imaging to identify hostile targets at an approximate range of 3 kilometers. It was revealed by the South Korean firm DoDAAM in December of 2010, but still requires human intervention although DoDAAM has confirmed that it can act purely autonomously. It was test at the demilitarized zone in Korea and provides a verbal warning before shooting. For the turret to shoot, a human must confirm that action. It has been implemented in various areas of the world since its reveal, proving its general success and providing further motivation for this project's success. Note that this project is unique to the others in that it is uses thermal imaging rather than a conventional camera to detect the presence of a target. This was considered by the team initially before being dismissed since purchasing a thermal camera solely for this operation, when we don't think that it is really any better than what the others have shown, was a waste of money. An image of DoDAMM's creation can be found in Figure 8.

Figure 8 DoDAMM's Super Aegis 2

3.2. Relevant Technologies

This section describes the various technologies that where investigated as well as the technology that will be used in the project. Comparisons will often be made to other existing products to explain why the final products that are chosen. User experience, price, weight, power consumption, and many other factors will often underpin decisions made here, as well as careful analysis of datasheets and technical manuals available for the various technologies. While many of these datasheets and manuals will not be provided here, the reader is encouraged to peruse them to better understand the final choices made.

3.2.1. Range Sensors

Range sensors are used by the robot to measure the horizontal distance from the target. There are multiple types of sensors that use different methods with various degrees of accuracy and effective measurement range. Of the plethora of sensors available, 3 were investigated; ultrasonic sensors, LiDAR sensors, and microwave sensors.

3.2.1.1. Ultrasonic Sensor

Ultrasonic sensors work on the principle of time of flight or ToF. ToF works in two parts, first the emitter emits am ultrasonic pulse above 18Khz. Then the device then waits, counting the time it takes until the sound wave is detected by the receiver. It then uses this time along with the speed of sound at 344 m/s to calculate the distance from the object. The general formula for this calculation is given below.

$$D = \frac{t}{2} * 344$$

The quoted maximum range for this type of sensor was only 5m or 16 ft. Since this was about half of the target maximum range of 30 ft. Other devices will need to be investigated. A diagram that depicts ultrasonic sensor function can be observed in Figure 9.

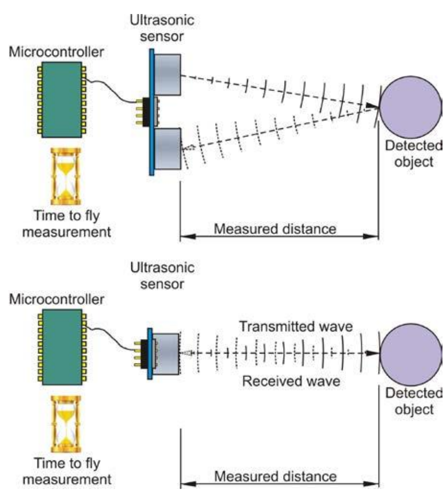


Figure 9 Ultrasonic Sensor ToF [6]

3.2.1.2. LiDAR Sensor

The next type of range sensor was had considered was the LiDAR range finder. This device also works on the principle of ToF but uses infrared light instead of ultrasonic pulses to measure distance. The benefit of using light waves over ultrasonic is the longer range as well as the polling speed. Light travels faster than sound so more pulses can be sent and received in the same amount of time as opposed to an ultrasonic sensor. Figure 10 below illustrates the basic concept of using Light as a ToF distance sensor.

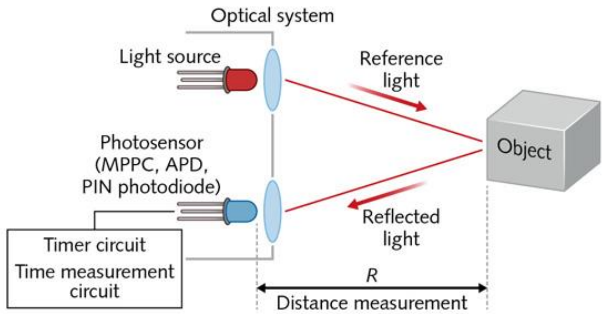


Figure 10 LiDAR ToF Range Finding [7]

Just like the ultrasonic sensor, a LiDAR can be used to find the distance from an object. However, LiDAR technology also offers a benefit over ultrasonic which is 3D depth perception. The image shown in Figure 11 is an elevation point cloud of Waldorf MD Near 38.640685N 76.930133W.

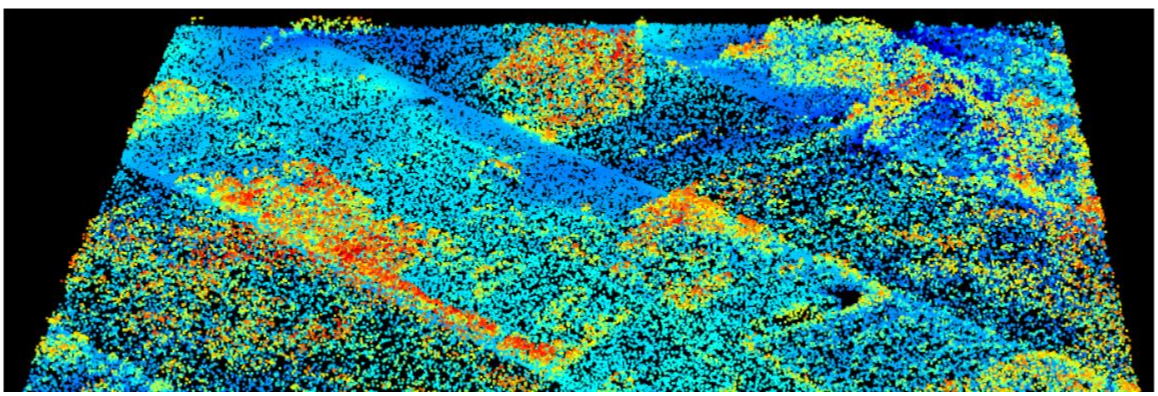


Figure 11 LiDAR Point Cloud, Copyright Free From data.gov

Although this type of technology would be extremely useful for this project the cost for one camera is well outside the budget of this project. Alternatively, a simple LiDAR based range finder as shown in Figure 10 was used and tested.

3.2.1.3. Microwave Range Sensor

The last distance sensor investigated is a microwave-based distance sensor. Microwave based range sensors also use the ToF basis for determining distance. The benefit that microwave has over LiDAR is that at a much lower frequency in the RF spectrum, allowing it to travel further at the same speed, the speed of light.

This solution was briefly investigated after LiDAR but since no commercial product within the project's budget was available it was quickly dropped.

3.2.2. Processors and Microcontrollers

Processors and microcontrollers are pivotal devices that can be observed in many robots, especially autonomous ones. As such, a careful look at many processors and microcontrollers must be had to choose the most appropriate one that matches the requirement specifications and allow for a high-performance bot. Below, a few of the many choices have been listed along with their strengths and weaknesses. The goal here is to highlight the decision-making process underpinning the choice in processor and microcontroller.

3.2.2.1. BCM2837

The BCM2837 is the processor which is used on the Raspberry Pi model 3B and Compute Module 3. The Model B is obviously not an appropriate device for this project as it is a complete computer with Wi-Fi, Bluetooth and Ethernet support. To purchase the BCM2837 by itself was not feasible since they are only available in large quantities. This leaves us with the Compute module which is a 200-pin breakout of the BCM2837 on a DDR2 package.

This was better suited for hardware integration as it gives the most freedom to the engineer to use in any needed application. Additionally, the module itself can be acquired for a low cost making it a perfect candidate for the master controller of the robot.

3.2.2.2. Atmel's ATMega2560

The ATMega2560 is the processor used on the Arduino Mega boards. Given how many subsystems and facilities will be accompanying this project, the ATMega2560 offers an expansive number of input/output options. Furthermore, this processor comes with much online support for troubleshooting and an interactive IDE with a simple coding scheme implemented through C. This was another viable option for the microcontroller implemented within this project but might offer too many input/output options. In that case, the ATMega328 can be another viable option, being very much like the 2560 model, but with far less input/output options.

3.2.2.3. Texas Instruments' Mixed Signal Processor 432 (MSP432)

The MSP432 is a processor that is based on an ARM Cortex-M4F CPU. Furthermore, in contrast to the MSP430, it has a larger address space and faster calculation speed. It provides another viable option, but within the domain of ARM microcontrollers. Furthermore, all members of the project have experience with the MSP430, so a transition to the MSP432 wouldn't be too complicated.

3.2.2.4. Texas Instruments Tiva TM4C123GH6PMT7 (TM4C123G)

The TM4C123G is another processor that is based on the ARM Cortex-M4F CPU. This controller has a large memory space with lots more addresses with the faster calculating speed at 80MHz. This microcontroller comes with four PWM modules each with four PWM generators, with a total of 16 PWM generators. The controller can be also programed like the MSP430 and would be relatively simple to program it. One notable advantage was that with 16 PWM generators, PWM lines can be used independently for each direction of each motor.

3.2.2.5. Comparison of Processor/Microcontroller

The following table compares important qualities of each of the presented processors and the various specs that were evaluated and taken into consideration.

	BCM2837	Atmel's ATMega2560	MSP432	TM4C123GH6PMT7
Operating Voltage	3.3V & 1.8V	1.8V - 5.5V	1.65V to 3.7V	3.15 V to 363V
Operating Speed	1.2 GHz	16MH	48MHz	80 MHz
Flash Memory	None	64K	256KB	256 KB
SRAM	1 GB	8KB	64KB	32KB
Sink Current	Variable	3mA	2mA	18mA
Power	Variable	1MHz per 500µA	95uA per MHz	
Cost	$35	$25	$13	$11.26

Table 2 Microcontroller Selection

3.2.3. Wireless Modules

The wireless module was a key component in the operation of the TankBot. It will be the bridge between the remote-control device and the robot itself. The wireless module should have a large bandwidth capability since it will need to stream the video from the robot's camera in real-time as well as receive commands. Since the main processor uses the Linux kernel, the module should also have drivers available to interface between it and the rest of the control system. Three devices currently meet these criteria, the WILC1000/3000 and the ESP8266.

3.2.3.1. WILC1000/3000 Wi-Fi and Bluetooth Module

The WILC1000 is a single chip IEEE 802.11 b/g/n RF/Baseband/MAC link controller with the 3000 variant supporting BLE 5.0. It supports single stream 1x1 802.11n with speeds up to 72 mbps. It is fully controllable via SPI, I^2C, UART, and SDIO interfaces and driver support for Linux is available at the manufacture's GitHub page.

This chip was our first choice for wireless integration as it has drivers already staged in the kernel source as well as first party manufacture drivers. Although the drivers will need a rewrite to work properly with the compute module.

3.2.3.2. ESP8266 Wi-Fi Module

The ESP8266 is an all in one highly integrated Wi-Fi-SoC made by Espressif Systems. The chip itself comes in various modules with integrated antenna's and communication interfaces including SPI / SDIO or I^2C / UART. The ESP8266 SoC features an integrated RF balun, PA, LNA, and filters making the module itself extremely compact. The module also features full Linux support via 3rd party drivers with speeds up to 72 mpbs supporting full IEEE 802.11 b/g/n. As with the WILC1000/3000 module the driver will need a rewrite to properly interface with the computer module.

3.2.3.3. Comparison of Wi-Fi Modules

Table 3 shows the three Wi-Fi modules to be investigated. While the ESP8266 was by far the cheapest, no 1st party drivers exist and the 3rd party driver is not very reliable. Moreover, the device does not officially support high speed SDIO and only through a hack can it be enabled; therefore, the other two devices will be investigated further.

According to the Microchip product page, the only difference in functionality between the ATWILC1000 and ATWILC3000 is that the 3000 variant supports Bluetooth 5.0. However, during testing, it was found that the 3000 variant performed 5 times faster than the 1000 variant on a local network speed test. Since local network speed was critical for high resolution video streams, the ATWILC3000 will be used despite it being the most expensive of the three devices tested. Figure 12 shows all three devices side by side, in the same order as Table 3.

	ESP8266	**ATWILC1000**	**ATWILC3000**
IEEE 802.11 Support	b/g/n	b/g/n	b/g/n
Bluetooth support	No	No	Yes BLE 5.0
Advertised max speed	72.2 Mbps	72.2 Mbps	72.2 Mbps
Tested max speed	4 Mbps	4Mbps	20Mbps
Driver Support	3rd party	1st party	1st party
Power Output	+20.5 dBm	+19 dBm	+19 dBm
Sensitivity	-98 dBm	-98 dBm	-98 dBm
Antenna type	Integrated Trace	Integrated Trace	Integrated Chip
External Interface	SPI	SPI/SDIO	SPI/SDIO
Cost	$2-$4	$7.34	$10.14

Table 3 Wi-Fi Module Comparison

Figure 12 ESP8266-12F, ATWILC1000, and ATWILC3000 Wi-Fi modules

3.2.4. Camera

Choice of camera was very important to the overall success of the project as computer vision will be the primary form of navigation and target detection/ tracking. As such, the decision of which camera system to use requires much discussion and that will be contained in this portion of the report. Primary considerations include resolution, frame rate, angle of view, and of course price.

3.2.4.1. Logitech HD Pro Webcam

The Logitech HD Pro Webcam is a high end easy to use option for computer vision projects and was seen on many of the senior design project in the past. It features 1080p resolution with an ultra-wide viewing angle and a frame rate of 30 fps. It connects to anything with a USB 2.0 connection which makes it every easy to implement in any project without having to design any specific connection for this sensor.

3.2.4.2. Raspberry Pi Camera Module v2

The Raspberry Pi camera module v2 is a high definition extremely customizable camera sensor. It has a very high resolution all the way up to 3240 by 2464 but this lowers its frame rate all the way down to 15 fps. If this frame rate was found to be too low for our needs the raspberry pi was capable of being modified such that the resolution drops to 1080p and the framerate increases to 30fps. This customizability was a plus because we don't yet know what kind of resolution we need to successfully identify targets at various distances.

3.2.4.3. GoPro Hero3 White Edition

The GoPro Hero3 White Edition is medium to large sized Wi-Fi compatible video camera. It was the most expensive of these options as it was the most commensally viable due to its durability, weatherproofing, and reputation for being very compatible with drones.

3.2.4.4. Camera Comparison

In the end we decided that the Raspberry Pi Camera Module v2 would be the best choice for our project for a couple of reasons, the first was its relatively low price point, allowing us to keep on budget. Also, the ability to decide what resolution and frame rate were needed was of great value to us. The last of the main reasons we choose this camera sensor was that its small size could be fit to the project in any way, so it would not impact the design of the project body. Below is a table comparing some of the most important variables for each of the camera systems that we were considering.

Name	Field of View	Frame Rate	Resolution	Price (USD)
Raspberry Pi Camera Module v2	Medium	15	3240 x 2464	$26.38
Logitech HD Pro Webcam	Ultrawide	30	1920 x 1080	$48.99
GoPro Hero3 White Edition	Wide	30	2592 x 1994	$69.00(used)

Table 4 Camera Comparison Table

Figure 13 Camera Comparison

3.2.5. Servo Systems

While the project may not have actual servo motors, there will be servo systems involved. A servo system can be implemented by having DC motors being fed-back to the MCU or other controller to handle error caused by the motor. To analyze the dynamics and the response of the motors, the characteristics of the motors is formulated:

$$J\ddot{\theta} + b\dot{\theta} = K_e i$$

The relation above shows the correlation between the moment of inertia (J), the motor viscous friction constant (b) and the electromotive force constant (K_e).

The following is the relationship between the passive electrical components that make up the motor and the voltage and the back EMF:

$$L\frac{di}{dt} + Ri = V - K_e\dot{\theta}$$

To model the motor being examined, the motor can simply be represented in the following correlation between the changing output angle and the input voltage:

$$\frac{\theta(s)}{V(s)} = \frac{K}{s(Js + B)(Ls + R) + K^2}$$

With the transfer function of this general DC motor, the poles of the system can be found. The response of the system can then be modified to meet the requirements desired of the system. To create a circuit to implement a feedback of the error, the circuit must be able to receive a PWM signal from its controller and the feedback from an encoder and compare the signal through a logic array that can send signal to the motor driver and keep the position of the motor at a desired spot.

To implement this, transfer function can be obtained by the varying the input of a motor and measuring its output. This will give us the transfer function of the motor as a relationship of angular speed over voltage. The only thing was to add a pole at the origin and we have the transfer function for the angular control of the motor.

3.2.6. Operating System

The main processor of the TankBot is the Broadcom BCM2837. This processor is a full ARM 64bit System on Chip (SoC) with a Memory Management Unit (MMU). Therefore, the main operating system (OS) that was run on the processor is a minimal Linux kernel and networking service along with the main control program. Using a Linux kernel also allows for the use of Portable Operating System Interface (POSIX) threads to allow for process segregation and better flow control with complex programs such as this.

3.2.6.1. Robot Operating System (ROS)

The Robot Operating System is an open source software library and tools used to build robots, while the software itself is not an operating system. In the Linux kernel, ROS is the middleware in which it gives package frames to help users produce robots. With ROS, the use of LiDAR and OpenCV can be integrated to detect objects in the real world.

Also, tools of ROS, Gazebo and Rviz, can help model and simulate the robot in a simulation. With it, the algorithms can extrapolate other bugs in systems that can exist while not using the actual robot. Though the use of ROS maybe left for researching the programing of the robot and simulation, and it may not be used in the final design.

3.2.7. Power

Power was a critical part of the TankBot design. The airsoft gun and servo motors draw large amounts of power and thus systems need to be in place to manage the large current draw of these components. The best way to manage was to use motors rated at the raw line voltage so the battery can be the main source of current instead of a regulator. To manage the large initial current draw of the motors, stiffening capacitors will be used in parallel with the inductive load to reduce dipping of the main line voltage.

3.2.8. Software Tools

The following sections describe the various software tools that will be used or a comparison of relevant choices for this project. The purpose of each software will be carefully considered as well as an analysis of the alternatives if available.

3.2.8.1. Advanced Design System (ADS)

ADS is one of the leading EDA software suites available for RF, microwave, high speed digital, and power electronics applications design. It is a powerful software suite which allows for the full analysis of things such as X-parameters, S-parameters, full-wave method of moment simulations, and non-linear harmonic balancing. ADS is used by leading companies in the wireless communication & networking, aerospace & defense, automotive, and energy industries.

For the TankBot project ADS will be used to calculate the high-speed trace dimensions and differential pair dimensions for the various communication lines.

3.2.8.2. PCB EDA.

There are a multitude of PCB EDA suites available, from amateur web browser based such as EasyEDA all the way to full professional suites such as Altium designer. However, for the purposes of this project, the two extreme ends of the choice range do not fit all the design needs while still being cost effective. Below is a table laying out all the PCB EDA options available to the team:

	KiCad	Eagle	Altium	DipTrace
Number of layers	2-32	2-16	Unlimited	2-unlimited
Board Area	Unlimited	$80cm^2$ - Unlimited	Unlimited	Unlimited
Number of Pins	Unlimited	Unlimited	Unlimited	500-Unlimited
Standard Part Library	Yes	Yes	Yes	Yes
Bitmap import	Yes	Yes	N/A	No
Gerber Export	Yes	Yes	Yes	Yes
Cost	Free	$15-$65* /mo.	$2995	$125-$348**

**Free for students for 3 years, equivalent to highest tier license with all features unlocked.*
***Free student version for lowest tier features 2 layers 500 pins.*

Table 5 PCB EDA Comparison Table

Although Altium designer is a full-fledged professional PCB software suite, it was obviously out of the budget of this project. DipTrace while offering a free version, was limited to only 2 layers which makes it un-usable for this project since we require a minimum of 4 layers to effectively route signals and power while having a solid ground under high speed signal traces. That leaves Autodesk Eagle and KiCad. The next two sections cover the features of each software suite.

3.2.8.2.1. Autodesk Eagle

Originally called Cadsoft Eagle and acquired by Autodesk, Eagle is a popular PCB EDA software for hobbyists and professionals alike. Originally, purchasing a license of eagle meant you owned that the right to use the version at the time of purchase for life, now Eagle has moved to the subscription model with the cheapest license starting at $15 per month. Students can get the full version of eagle professional for free for 3 years. Eagle offers like other PCB EDA software, a schematic editor, modular design blocks, ERC validation, push and shove routing, obstacle avoidance routing, and auto routing. The premium version of Eagle currently offers a maximum of 16 copper layers.

3.2.8.2.2. KiCad

KiCad is an open source GPL licensed EDA software for Windows, OSX and Linux. KiCad allows for the creation PCB circuit schematics, board layouts for free with the most advanced features offered by other paid software suites. Like eagle, KiCad features a schematic editor, modular design blocks, ERC validation, push and shove routing, obstacle avoidance routing, and auto routing. KiCad currently features a maximum of 32 copper layers and 32 misc. layers available to the designer.

3.2.8.2.3. EDA Decision

Since our group has no experience in using Eagle and some experience using KiCad, KiCad was chosen as our EDA software of choice. Along with the benefit that experience learned while using KiCad for this project will always be able to be used for free and not require money once the student license expires.

3.2.8.3. 3D Modeling software

3D modeling software will be used to design custom parts for the TankBot to be either 3D printed, or CNC machined at UCF. This section describes the various software tools used to accomplish this task or a comparison of available tools.

3.2.8.3.1. FreeCAD

FreeCAD is a parametric 3D modeler made primarily to design real-life objects of any size. In addition to being a zero-cost 3D cad solution, FreeCAD is also licensed under the Lesser GPL license with its source freely available.

3.2.8.3.2. SolidWorks

SolidWorks is another 3D parametric modeling software by Dassault Systems. This software is used by engineers to design and model 3D parts, simulate mechanical systems with constraints, and prototype designs before construction.

3.2.8.3.3. 3D Modeling Software Comparison and Decision

While there are many similarities and differences with FreeCAD and SolidWorks, for the purposes of this project, designing and modeling 3D parts for manufacturing, they perform the same. Thus, the main differentiating factor comes down to price, FreeCAD is 100% free while SolidWorks is base priced at $3,995. However, as a UCF student, we have free access through CECS. Then the choice was left to the usability of the software. Since none of our group members have experience in either, both software suits where downloaded and tried.

The learning curve of FreeCAD seems rather steep as we had trouble creating a simple multi shape 3D part. Drawing in 2D and extruding and/or revolving was also a difficult task to find how to do. In contrast SolidWorks UI was simple to navigate and learning curve was small. Our team member, was able to with no prior experience, draw and render a fully 3D correctly dimensioned part in SolidWorks in a few hours. The menu and options where clean and finding tools was simple.

Therefore, SolidWorks was chosen as our 3D modeling software of choice. This will be used as one of the software tools that will be used to design custom parts for the TankBot.

3.2.8.4. Computer Vision Framework

The computer vision software was to be the “eyes” and “mind” of the robot. As such to tackle the incredibly complicated problem of not only identifying human vs non-human targets, but also attempting to distinguish friend from foe, we will make use of a relatively new inclusion to the engineering discipline, Computer Vision. Computer Vision is an intersection of Computer Engineering and Artificial Intelligence that attempts to allow computers to turn visual inputs into workable information via machine learning, also called deep learning. In the following sections we will present three of the most prevalent software's on the market and discuss the pros and cons of each and talk about our choice of software that we will use on this project.

Careful consideration has been made particularly to cost and learning curve as none of the members of this project have any previous experience in CV, which was the most important software application that this robot will have at its disposal. After all, the autonomous decisions of the robot and the 'correct-ness' of these decisions, so to say, stem from the choice in software used in the final implementation.

3.2.8.4.1. MATLAB

MATLAB is a privately licensed programming platform used by many private and public institutions. The Computer Vision System Toolbox available for MATLAB includes code for Deep Learning, Object Detection and Recognition, and Motion Detection and Estimation among others. These three abilities are key to the operation of TankBot and will need to be investigated further.

The benefits of MATLAB are that it is relatively easy to learn, it was often used in our engineering curriculum, so all the members of our team have some experience with the software. Also, it is written in Java, C, and C++ so it is easy to find a language that we have experience in. In addition, MATLAB code is notoriously easy to debug. Lastly, it is usable on all the major operating systems, so we wouldn't have to limit our design based on the requirements of this software.

Some of the cons of MATLAB are its price, over $1300 for a single license to the software with access to the Computer Vision System Toolbox, and the difficulty in porting MATLAB files from one machine to another, due to the need for an interpreting software which can dramatically decrease performance un-necessarily.

3.2.8.4.2. OpenCV

OpenCV on the other hand is entire free to use, as an open source library it has been widely used and thus has many supporting resources. OpenCV was initially developed by Intel Corporation but is currently controlled under a BSD License which allows for businesses and other organizations to use and modify the provided codes freely. The library includes code to track movement, recognizes faces, and identify objects.

Pros to OpenCV are its cost, it was a totally free library of open sourced code. It also has a large community of users, making finding resources relatively easy. The generality was also a plus, in that it can be used in any programming environment and with one of many programming languages including Java, C, python, and C++. While OpenCV can be used with any of those languages, the native language is written in C++ with java and python using library calls. OpenCV being written in C/C++ was a plus since it was the language that our group has the most experience in and allows for a native implementation into the TankBot. OpenCV is also consistently updated with a new version to keep it as up to date as possible.

Cons are its initial complexity which makes for a large learning curve, since none of the members of our group have any experience with using this tool it makes setting up a working prototype a challenge.

3.2.8.4.3. Aforge

The last API (application programming interface) that we explored for our project is called Aforge. Aforge was developed by an individual named Andrew Kirillov for the .NET framework in 2006. It is licensed under both the GPL and the lesser GPL which means that the libraries are free to use, and modify, by both free and propriety programs.

The pros of using Aforge include its cost, being open source and essentially free to use and modify. It is also useable in many of the most popular programming environments including DevC++, Microsoft Visual Studio, and Codeblocks, due to it being written in C#.

Unfortunately, as we did more and more research, we found more and more drawbacks to Aforge, including but not limited to the fact that the last update to the software was published in 2011, nearly seven years ago. Also, the fact that not one of our group members have had any experience using this software, it also seems to be a relatively small community of users and developers meaning that most of the problems that we run into will have to be solved in house.

3.2.8.5. Software Framework Decision

In the end, our group decided that OpenCV would be the best choice for this projects Computer Vision needs. This decision was made based mainly on a couple of factors. First of these factors was the availability of a teaching software, provided by LearnOpenCV.com, to smooth the transition between having zero experience with a software and the first interaction with it. The second main point was the ability to use whichever coding environment we want thus allowing easy communication between systems. Lastly, but still important, was the issue of price, MATLAB’s $1300 price tag really puts a strain on our proposed budget. In the end OpenCV’s open source libraries of optimized and flexible code was the best option for us. The table below summarizes the numerous aspects considered in the selection of the computer vision software:

	MATLAB	OpenCV	Aforge
Cost	$1300	Free	Free
Experience	High	None	Some
License	Private	BSD License	GPL License
IDE	Private Built in IDE	Any	Many Common IDE's
Supported Languages	Java, C, C++	Java and python lib calls. C backports, currently used with C++	Aforge Language
Community	Extremely Large	Large	Small
Updates?	Yes	Yes	No

Table 6 CV Comparison Table

3.2.8.6. LiDAR Mapping

One of the main goals of the robot was its ability to not only move but move to a specific location in relation to not only itself, but also landmarks in the immediate area. To do this, the TankBot must to either have a map of the local area downloaded to it, or develop a map for itself, this was the method we chose for several reasons, the first was that we don't really have affordable access to high resolution images of UCF, which was where the demonstrations will be held. Secondly, and more importantly, most of the testing and demonstrations will be done indoors which would require us to either model the rooms we test and demonstrate in, or map it by hand, both options are time consuming and unnecessary when we can just have the robot do the job for us. All this leads to the need for TankBot to develop its own map as it was operating. There are 2 main methods of doing that and both will be discussed in the following sections.

3.2.8.6.1. Simultaneous Localization and Mapping (SLAM)

SLAM is a method of using LiDAR inputs to map surrounding terrain including but not limited to, surrounding walls, steep hills, and stationary objects in the vicinity. This is done through the utilization of several proprietary algorithms that can be found in several textbooks or online courses. These algorithms allow us to theoretically place the robot in a previously unknown room, turn on the machine, and have it "explore" its surroundings to create and store a map of the area that it can find itself on in a moment.

3.2.8.6.2. Simultaneous Localization, Mapping and Moving Object Tracking

SLAMMOT is a sort of extension of SLAM and can be implemented if SLAM is ineffective because of drift. The extra algorithms that SLAMMOT provides form the basis for counteracting the effects of drift due to movement by taking inputs from the wheels/treads/ locomotive systems to account for the movement of the machine. algorithms that SLAMMOT provides form the basis for counteracting the effects of drift due to movement by taking inputs from the wheels/treads/ locomotive systems to account for the movement of the machine.

3.2.9. Autonomous Detection Processes

One of the primary tasks for our robot to achieve to be successful is to be able to detect objects, namely, people, autonomously. There are many ways of doing this, many of which can be combined into a multi-step method of autonomous detection. This section will cover some of those methods and discuss the pros and cons of each.

3.2.9.1. Object Recognition

A subsection of autonomous detection is called object recognition, this is classified as a feature-based detection method. There are a couple of widely used techniques on achieving this process, and they will be discussed below.

3.2.9.1.1. Hough Transform

The Hough Transform is a method of extracting features from an image or object. It attempts to find parts or misshapen wholes of things that are known to the software in an image, and then “voting” on them to classify these objects or partial objects. By classifying these images, the software can then reference this information in the future to better identify new objects. Figure 14 displays an example of the application of the Hough transform by OpenCV.

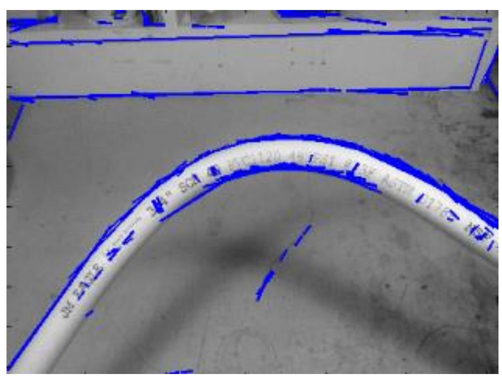


Figure 14 Hough Transform in OpenCV

3.2.9.1.2. Edge Detection

Edge Detection is a type of mathematical technique that tries to identify points in any image where the brightness values change drastically over a small space, these areas, when found next to one another in a line or curve, are likely to be the edges of objects in an image. This is very useful as one of the main features we need are robot to preform is identify the human *shape* and if we can determine where the edges of an object are, we are that much closer to being able to identify a person. For this reason, we will apply the edge detection method using our computer vision software in this project. An example of edge detection applied by OpenCV can be found in Figure 15.

3.2.9.1.3. Corner Detection

Corner detection is another type of mathematical expression attempting to define an image, this time by taking an image and breaking it down into its constituent *corners*. Here we define corners as a point where two edges of a color or brightness meet. During our research we discovered that corner detection is not very good at being the main technique for object detection due to the errors that arise relatively often during its use. For this reason, we will not be implementing corner detection in our project.

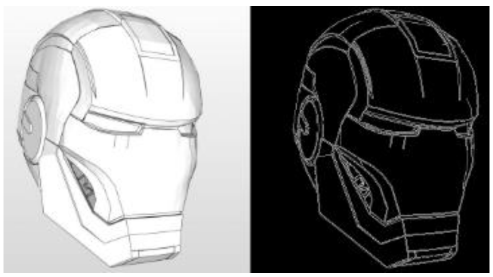


Figure 15 Edge Detection in OpenCV

Figure 16 Blob Detection in OpenCV

3.2.9.1.4. Blob Detection

Blob Detection is yet another method for detection objects in images. This method finds areas, rather than lines or corners, of differing properties, such as brightness and color, when compared to the nearby areas. Note that in addition to brightness, which is used by both Edge and Corner Detection as well as Blob detection, blob detection also see color, which could be useful in our application of trying to determine friend or foe, maybe we could use this method to even single out articles of clothing or patches on a uniform to further classify targets. For this reason, in addition to the relative robustness of the blob detection method compared to the corner detection method, we will use blob detection in our project in addition to edge detection. We feel these combined methods will almost certainly provide us with the accuracy of detection that we need. Figure 16 displays an example of this type of detection in OpenCV.

3.2.9.2. Motion Detection

Motion detection and tracking was one of the most important issues we faced during this project with regards to its success or failure. Success is being able to hit a target, which was most likely moving, and failure is not. For our autonomous machine to hit a moving target, it must first be able to track a moving target, and then adjust for that motion when aiming the gun.

For TankBot to be able to detect motion we made use of two separate systems simultaneously. Namely, the camera, and the range finding system, which in our case, is LIDAR. The camera, using object detection, attempted to find a target in its field of view always while the range finder provided depth readings to the targeting system. These combined gave the targeting system a solid footing with which to tackle the problem of firing on a moving target while the robot itself is moving.

3.2.9.3. Facial Detection

Facial detection was one of the easiest system to implement on our robot, simply because there is so much material and work already done on the subject out there on the web. During our research we found numerous example of computer vision software being purposed to detect faces that were free to use. For this reason, we were able to just download a pre-made facial detection software and with a few lines of code implement it into our final design. An example of face detection applied by OpenCV can be found in Figure 17.

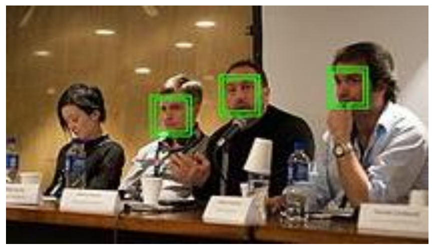


Figure 17 Automatic Face Detection with OpenCV

3.2.9.4. Human Detection

Human detection is a subsection of object detection that required much more thought and effort to implement correctly. As opposed to face, bodies have many, many more features to keep track of and use as decision points by a given program. For this reason, it was far more computationally demanding than just facial detection. While there are many resources available online, this one feature of the robot took more time and effort than any other single subsection of the project.

3.3. Standardized Communication Protocols

Various standardized communication protocols were available at our disposal. These communication protocols are implemented as hardware interfaces to communicate between the major controllers in the system. Effective communication between the various sections of this project was pivotal for the final success of the bot, so carefully peeling through the various options available was very important to the overall design and implementation of this project. The following sections describe the relevant standards as well as their implementation.

3.3.1. UART

Universal Asynchronous Receiver Transmitter, or serial, is one of the most common communication protocols used to relay information between devices in a system. UART can be implemented in software but more commonly implemented in hardware due do the zero processor overhead needed unlike software.

Hardware UART is handled by a dedicated hardware state machine independent of the main processor and information is retrieved and sent via special registers from the main processor. When a frame is received, the hardware will raise an interrupt, so the main processor can retrieve the data before returning to the main program execution. When a frame is to be sent, the main processor will write the frame to a send register, the UART hardware handles the sending of the data. A standard UART frame is shown in Figure 18.

UART communication uses a minimum of two wires called RXI and TXO with optional flow control CTS and RTS for full RS232 support, although flow control will not be used in this design. Data in UART is sent in frames, each frame contains between 5 and 9 bits of data, 1 start bit, 0-1 parity bits, and 1-2 stop bits. In UART, the default state of RXI and TXO is a pull-up to logical high.

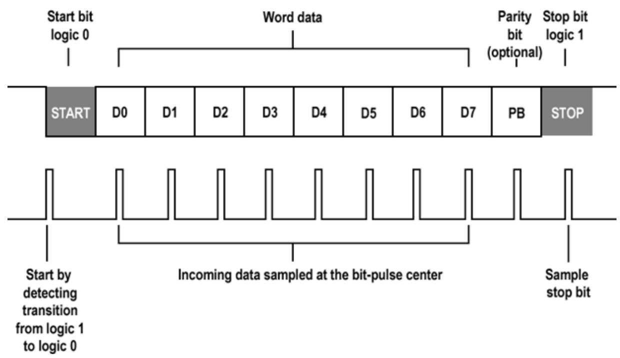


Figure 18 Single UART Frame[1]

Since UART is asynchronous, there is no master / slave control and either device can send data at any time, so the sender and receiver must be in sync in terms of timings. To manage the timings, a baud rate timer was used to set the clock timing for each bit. It was crucial that each device have the same timing to ensure data integrity. Figure 18 diagrammatically displays the organization of a UART frame.

Start Bit: Since it was impossible for both devices to remain in sync for all time, even at the same baud rate, a sync condition must be created to inform the receiving device that this is the start of the frame. The start bit signifies the start of a frame, this was required to start data transmission. To send the start bit, the sending hardware pulls their TXO line low for one clock cycle which signifies to the receiver that data will be sent on the next clock cycle. The next cycles that follow are the data frame.

Data Frame: The data frame is where the bits of information are sent by the sender on TXO and received by the receiver on RXI. The length of a frame can vary between 5 and 9 bits but is most commonly 8 bits or one byte. Data bits are sampled in the middle of each clock cycle as opposed to the rising or falling edge to give time for the bit to settle.

Parity Bit: This bit acted as a sort of check-sum for data integrity. Data sent over long distances or data effected by electromagnetic interference can cause bits to flip resulting in errors. The parity bit is a way for the UART hardware to know if the data is incorrect. It does this by summing all the “1’s” in the data frame and checks the LSB of the sum, if this bit does not match the parity bit the hardware knows there was an error in transmission.

Stop Bit: This bit signifies the end of a frame. The sending device asserts TXO low for 2 clock cycles signifying transmission end.

[1] Copyright Electricimp.com, See APPENDIX A for permissions

3.3.2. I^2C

I^2C is a communication protocol that uses two wires called SDA and SCL for communication between one master and multiple slave devices. The SDA is the data signal while SCL is the clock signal. The clock signal is always generated by the master device on the bus. Unlike UART where communication is bi-directional where either device can request data transfer, I^2C requires the master to initiate the data transfer. Additionally, the I^2C lines are open drain, requiring external pull-ups to the signal voltage unlike UART. The limit on the number of I^2C slaves on a bus is only limited by a maximum bus capacitance of 400pF as defined in the standard.

The process for data transfer between a master device and slave device is as follows:

First the data transfer is initialized by a start condition with SDA asserted low quickly followed by SCL asserted low. The next 7 bits are the address bits to which all slave devices on the bus listen in on and compare to their addresses. If a device on the bus has an address which is the same as the address transmitted by the master an ACK is sent which acknowledges to the master that the requested slave device is available. The next byte sent on the bus is the 8 bits of data followed by ACK. Once all data has been transmitted a stop condition is sent consisting of SCL asserted high followed by SDA asserted high. This process is shown in Figure 19.

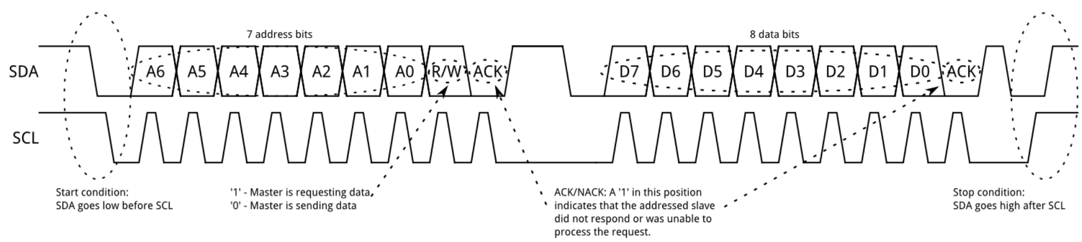


Figure 19 I^2C Communication Timings

3.3.3. SDIO

Secure Digital Input Output is a form of data transmission that is commonly used to communicate with I/O devices under the MMC protocol. The SDIO specification allows for the transfer of 1, 4, or 8 bits in a single clock cycle and due to its high bandwidth capabilities, it is commonly used for transmission from primary systems to subsystems such as Wi-Fi and Bluetooth controllers.

All SD bus transactions are initiated from the host controller through the CMD pin. All CMD and DATA signals are synchronized with the CLK pin. The beginning of a data transfer starts with a 48-bit command, the format for a command is shown in Figure 20.

1 start bit	1 direction bit	6 cmd index bits	32 argument bits	7 CRC bits	1 end bit

Figure 20 SD Command Packet

Some commands require a response from the slave device. If this is such a command, the response packet is shifted on the CMD pin. The size of a response is either 48 bits or 136 bits. If any data transfer is requested by the command, the DATA pins are used in either 1, 4, or 8-bit modes.

3.3.4. SPI

Serial Peripheral Interface is a protocol standard that is used to interface one master device with one or multiple slave devices. Unlike UART or I^2C, SPI uses separate lines for clock and data transmission between master and slave devices as seen in Figure 21. As such, more wires are needed. However, the benefit was that the clock frequency is higher allowing for faster data rates.

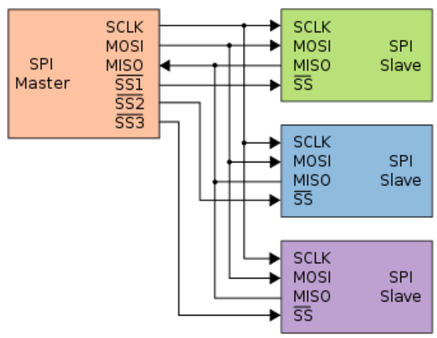


Figure 21 SPI Connection with Multiple Devices

When a master asserts the Slave Select pin low, the slave device is selected to begin data transmission. The master then controls the clock and uses the MOSI pin to send the command byte to the slave device. If command is a read command, the slave device sends the data via the MISO pin. For a Write command, the master sends 1-3 bytes of data per transaction to the slave device. This data transfer can be seen in Figure 22.

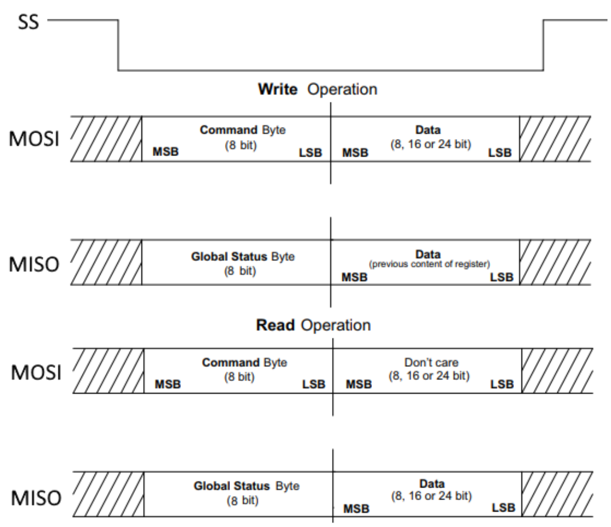


Figure 22 SPI Operation Modes

3.3.5. MIPI

The Mobile Industry Processor Interface or MIPI is a specification which defines the standard interface by which a device can connect to a host processor. The benefit of using this type of interface over others such as serial was that it allows for extremely high bandwidth.

3.3.5.1. MIPI M-PHY

The M-PHY is a performance driven physical layer definition of the MIPI interface. It uses standard differential pair signaling and the performance of M-PHY is defined as 3 high speed ranges, called gears, shown in Table 7.

M-MPY Speed	Clock Rate	Bit Rate
Gear 1	G1a	1.25 Gbps
	G1b	1.49 Gbps
Gear 2	G2a	2.5 Gbps
	G2b	2.9 Gbps
Gear 3	G3a	5 Gbps
	G3b	5.8 Gbps

Table 7 M-PHY Supported Speeds

The M-PHY layer is a full duplex design, which means it allows for concurrent data transmission in both receiving and transmitting. Lanes can also be implemented asymmetrically to serve and benefit various devices such as cameras, modems, or any other application that has highly asymmetric traffic.

Though the M-PHY later uses high frequency switching, it can achieve a low power consumption for a wide range of data rates, this is because of the low switching voltage during operation.

Typical applications for M-PHY are universal flash storage, mobile PCI-E and SuperSpeed USB Inter Chip which optimizes USB 3.0 over M-PHY. This allowed a broad range of USB-enabled functions to be used on mobile devices.

3.3.5.1.1. CSI

The MIPI specification defines an interface between a camera or multiple cameras and the application processor or image signal processor called Camera Serial Interface. CSI-2 first developed in 2005 saw widespread adoption in mobile phones and now virtually every smartphone built today.

CSI-3 is the next iteration of the interface specification, allowing for high resolution and high megapixel sensors with fast framerates. CSI-3 also supports bi-directional image transmission between hosts based around the M-PHY specification. The original release of CSI3 was in 2012 with V1.1 released in 2014

3.3.5.2. MIPI D-PHY

The D-PHY is a physical layer for high performance, low cost cameras and displays. This standard has become the industries primary solution for high-speed PHY applications in smartphones. It is typically used with MIPI's CSI-2 or DSI protocol specifications. Unlike M-PHY, D-PHY is only half duplex, data only travels in one direction at a time. Typical differential voltage swings are 200mV, though they can be operated at 1.2V at a lower speed of 10 Mbps

D-PHY delivers data at up to 1.5 Gbps per lane, typically 4 or 8 lanes are used. D-PHY also uses a standard clock forwarding technology which is suitable for most cameras and displays.

3.3.6. Networking

Various networking protocols exist within the Linux kernel space. Two of which stand out as potential candidates to use for communication between a client and server via local networks, TCP and UDP.

3.3.6.1. UDP

User Datagram Protocol is one of the main protocol standards of the Internet Protocol model. USD allows computers to send packets of information called datagrams to each other on a network. Unlike TCP, a UDP client requires no acknowledgement from the server and will send data regardless of if the server is accepting it. As such, prior communications are not required to setup a communication channel. UDP is suitable for applications where error checking and correction are either not necessary or performed by the application rather than the network stack. Time dependent applications will use UDP since dropping packets is preferable to waiting on packets being delayed due to retransmission.

3.3.6.2. TCP

Transmission Control Protocol is another of the main protocol standards of the Internet Protocol model. TCP allows for reliable, ordered, and error checked delivery of byte streams between applications running on hosts inside an IP network. The TCP accepts data from a data stream, splits it into chunks and adds a TCP header to create a TCP segment. This segment is then wrapped into an IP diagram and exchanged with peers. A TCP segment consists of a segment header and data block. There are 10 mandatory fields, and one optional field. Unlike UDP, TCP requires the server to acknowledge the client for data transfer to occur.

3.3.6.3. Sockets

Sockets allow communication between different processes on the same machine or between two or more machines on a network. In Linux, every I/O action is done by writing or reading a file descriptor which is just an integer associated with an open file and it can be a network connection, a text file, a terminal, or something else. In the programming space, a socket looks and behaves like a low-level file descriptor which allows for commands such as read() and write() to work with sockets in the same way they do with files and pipes. This allows for the sending of information in either UDP or TCP packets between hosts on a network.

3.4. Components and Part Selections

The final selection of parts and their analysis against other relevant, existing parts important for maintaining the requirement specifications as well as outputting a high-performance product. Below the selection process behind these components and parts are detailed. Often, comparisons are made to other products. In the cases in which this is not done, the authors believed it was irrelevant to display other choices because of how remarkable the object was amongst the crowd of others. The remarkability is highlighted by the performance, familiarity, or cost of the product when compared to other viable choices.

3.4.1. Battery Selection

To power the robot, a portable energy source was needed, the obvious solution was to use batteries. The three main types of battery technology available were lithium cobalt oxide, nickel-metal hydride, or lead acid. Each type of battery chemistry has benefits and drawbacks listed in Table 8.

	Lithium Cobalt Oxide	Lead Acid	NiMH
Rechargeable	Yes	Yes	Yes
Nominal Voltage	3.7V	2.1V	1.2V
Cutoff Voltage	2.5V	1.75	0.9V-1.05V
100% SOC	4.2V	2.23V-2.32V	1.3V
Energy Density(MJ/kg)	0.7 MJ/kg	0.11-0.14 MJ/kg	0.36 MJ/kg
Energy Density(MJ/L)	2.0 MJ/L	0.22–0.27 MJ/L	1.44 MJ/L
Specific Power	-	180 W/kg	250-1000 W/kg
Cost	1.99 W/$	3.6-4.2 W/$	1.54-1.66 W/$

Table 8 Battery Chemistry Comparison

The main points were energy density and cost. The best performing out of all three chemistries was the lithium cobalt oxide with an energy density 64% larger than NiMH and 133% larger than lead acid. Lithium made an excellent choice where energy density and weight are the most important factors. The robot will be mobile, so weight is a concern, but the cost performance of the lead acid chemistry makes it an attractive choice to power the robot. Therefore, lead acid batteries where chosen to power the robot despite it being the lowest in terms of energy density. Additionally, the robot will be drawing large amounts of current to power the drive motors, lead acid batteries make a good choice due to their ability to source large amounts of current.

3.4.2. Voltage Regulation IC's

The voltage regulator IC is the component that will provide power to all the control systems, IC's and processors. As such it needed to be versatile and robust. There were two common types of voltage regulation technologies, linear and switching.

3.4.2.1. Linear Regulator

Linear regulators are simple devices, the basic circuit is given in Figure 23. The basic theory of operation is the Op-Amp controls the BJT transistor load line to try and set the output voltage to the value set by the resistor voltage divider. While simple, the linear regulators efficiency is inversely related to the ΔV of the input vs output voltages, that is the larger the difference in voltages, the worse the efficiency. Moreover, since the voltage regulation is done in series by the transistor, the ΔV is dropped across the Collector-Emitter junction.

This created a thermal problem when the current draw is high. With low efficiency and most of the power being lost as heat, the linear regulator makes a poor choice for large load regulation. The linear regulator is better suited for analog or low power applications where low noise is more favorable over efficiency. As our application is both high power and battery operated, this failed on both accounts.

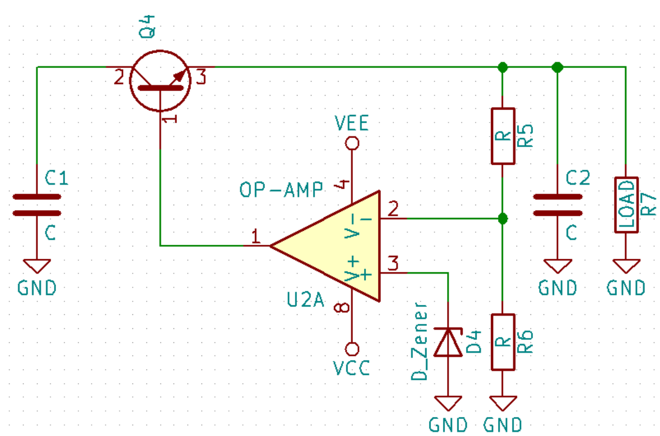


Figure 23 Basic Linear Regulator

3.4.2.2. Switching Regulator

The switching regulator is the next type of regulator circuit. Unlike the linear regulator, the switching regulator boasts high power efficiency and low thermal loss vs power output. Instead of using an Op-Amp to resistively drop the voltage across a transistor, a switching regulator turns on and off the input voltage at a specific frequency. During the on cycle the magnetic field is propagated across the inductor while current also powers the load. During the off cycle the magnetic field collapses generating forward current which powers the load. The output capacitor is used to stabilize the voltage during the off cycle. And Feedback informs the PWM circuit to increase or reduce the duty cycle to maintain constant voltage. The basic circuit is shown in Figure 24.

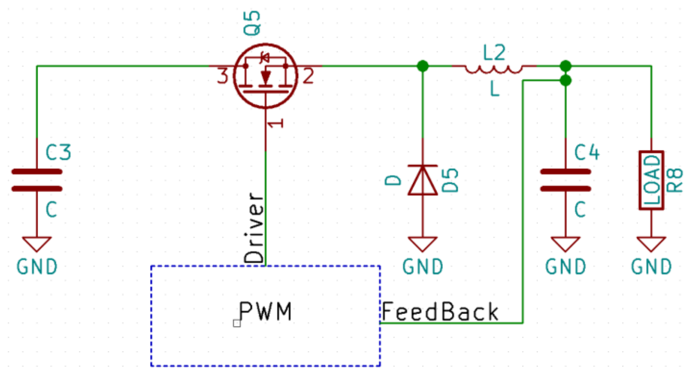


Figure 24 Basic Switching Voltage Regulator

Because the circuit uses a switching signal for voltage regulation, both voltage and current noise are significantly increased. Voltage noise comes from the output capacitor selection. Not only does the capacitor need to be properly sized for the switching frequency, but also its ESR needs to be low. High ESR capacitors such as aluminum electrolytic capacitors cause a reduced rate of charging and discharging which increases the ripple voltage on each cycle.

Current noise comes from the load itself and an improperly sized inductor. When the current draw is to large and the inductor is saturated. The current will ripple as the regulator struggles to maintain a constant voltage.

3.4.2.3. BCM2837 Power IC

The BCM2837 on the compute module 3 requires two main input voltages, 3.3V and 1.8V. Various switching regulators where considered for this task with the requirements being it must supply a minimum of 1A on each voltage level either with a two in one package, or two separate regulators. For this, three regulators where considered which meet the requirements, summarized in Table 9.

	AP3512EMPTR-G1	FAN53611AUC18X	PAM2306AYPKE
Input Voltage	4.5V-18V	2.3V-5.5V	2.5V-5.5V
Output Voltage	3.3V	1.8V	3.3V & 1.8V
Switching Frequency	500 KHz	6 MHz	1.5 MHz
Maximum Efficiency	90%	92%	96%
Nominal Output Current	1A	1A	1A, 1A
Package	SO-8EP	6-UFBGA	WDFN-12L 3x3
Manufacturer	Diodes Incorporated	ON Semiconductor	Diodes Incorporated
Cost per Unit	$0.61	$0.75	$0.71

Table 9 1.8V and 3.3V Switching Regulator Comparison

The important considerations were power efficiency, cost, and output current. All three of these regulators met the output current rating and have good efficiency, but the PAM2306 stands out as the winner of the three in terms of cost. The PAM2306 is a dual step-down DC-DC switching regulator at $0.71 per unit and can supply 1A per channel effectively beating both regulators by 50% since we would need both to meet the power requirements of the processor. For this, the PAM2306 was chosen.

3.4.2.4. 5V Voltage Regulator

Many of the sensors, components, and dedicated chip power supplies run on 5V nominal voltage, therefore a 5V rail was needed to supply sufficient power to these components. The following regulators shown in Table 10 list the different voltage regulators selected to fit this need.

	ACT4060ASH-T	AP65450SP-13	MIC24056YJL-TR	MIC261201YJL-TR
Input Voltage	4.5V-24V	4.5V-18V	4.5V-19V	4.5V-28V
Output Voltage	Variable	0.765V-6V	5V	5V
Switching Frequency	400 KHz	650 KHz	600 KHz	600 KHz
Maximum Efficiency	96%	-	95%	95%
Nominal Output Current	2A	4A	12A	12A
Package	SO-8	SO-8EP	5x6 QFN	5x6 QFN
Manufacturer	Active-Semi	Diodes Incorporated	Microchip	Microchip
Cost per Unit	$0.59	$0.83	$2.27	$3.28

Table 10 5V Switching Regulator Comparison

The first choice to power the sensors was the ACT4060ASH-T however during testing it was found that because there were no thermal pads on the device which made the device burn and fail when loads larger than 800mA where drawn. The next choice was then AP65450SP-13 which could source a maximum of 4A and only marginally more expensive. Again, testing revealed that the component was not up to spec. The issue being the device could not maintain the target voltage under any load and the PWM frequency was largely out of spec at 1.1 MHz. The final choice is the MIC series of regulators from microchip. Although these regulators are more than 2 times costlier than the other regulators, our team was able to obtain 3 of each as free samples.

3.4.3. MOSFET

Metal-Oxide-Semiconductor Field Effect Transistors (MOSFETs) are incredibly common electronic devices used in many different integrated and discrete circuits. MOSFETS come in 2 modes, with 2 variants available in each mode. These are depletion and enhancement mode which come in both P-Channel and N-Channel.

Depletion mode MOSFET's are very uncommon while enhancement mode MOSFETS dominate the industry in everything from digital computers to analog devices. Which is why for this project we focused exclusively on enhancement mode MOSFET's for all MOSFET transistor applications. Their primary use was as switches or in applications involving amplification of voltage or current. They are also commonly used in the design of input and output stages for various applications. Their incredibly commonality yielded an absolute ton of choices to pick from, in which even similarly named devices might be developed by several different companies.

As such, their selection was done carefully due to sheer volume of different implementations and creations.

3.4.3.1. Load Switch Power MOSFET

The load switch is an integral part of the power distribution as it connects the power source to the power regulation and as such, has the entire system load passing through it. Therefore, The MOSFET needed to be chosen such that the $R_{DS(ON)}$ was minimal such that the total power dissipation is low. Three of the lowest priced MOSFETS were compared which met these requirements shown in Table 11. Power dissipation values are given with the thermal pad soldered to ground, Thermal resistances are junction to case.

During testing, our team was going to use the SiS413DN as it offered superior performance over the DMP3010LK3. However, the new MOSFET, DMP3013SFV, available from Diodes Inc was slightly cheaper while still offering acceptable performance in all critical areas.

Our team had already acquired the SiS413DN but since the pinout and package of the DMP3013SFV is the same, if the need arises this would be our second purchase.

	DMP3013SFV	**SiS413DN**	**DMP3010LK3**
Breakdown Voltage	-30V	-30V	-30V
$V_{GS(\max)}$	$\pm 25V$	$\pm 20V$	$\pm 20V$
$V_{GS(th)}$	-1V to -3V	-1V to -2.5V	-1.1V to -2.1V
$R_{DS(ON)}$	$9.5m\Omega$	$9.4m\Omega$	$6.5m\Omega$
Gate-Source Leakage	$\pm 10\mu A$	$\pm 100\mu A$	$\pm 100\mu A$
Power Dissipation	31W	52W	3.4W
Thermal Resistance	$4°C/W$	$1.9°C/W$	$37°C/W$
Continuous drain current	-35A	-18A	-17A
Package	PowerPak (Type UX)	PowerPak (Type UX)	TO252
Manufacturer	Diodes Incorporated	Vishay Siliconix	Diodes Incorporated
Cost	$0.49	$0.60	$0.60

Table 11 P-FET Comparison

3.4.3.2. NX7002AK N-Channel MOSFET

This MOSFET is used to invert an active high signal to an active low signal. No cost analysis was performed with this part against other parts as this was already in our teams' possession in moderate quantities before the beginning of this project and as such the acquisition cost is zero. The cost of each of these MOSFET's was approximately $0.04 each. With their incredibly low cost, excellent switch time, and a high gate source/ drain source voltage this MOSFET makes an excellent cheap device for a variety of uses in this project including logic inverting, level shifting and general fast signal switching.

3.4.3.3. BSS84 P-Channel MOSFET

This MOSFET is used to invert an active low signal into an active high signal. As with the NX7002AK N-channel MOSFET, this P-channel MOSFET was already in the possession of one of our group members in moderate quantities and thus the cost of acquisition was zero. The average cost payed to purchase this MOSFET was approximately $0.09 each.

3.4.4. Voltage Supervisor IC

The voltage supervisor IC is a critical part of the overall design of the robot. This is the device that will control the load MOSFET to connect or disconnect the power source to the rest of the circuit. The voltage monitor should also be able to inform the system about the voltage status and in the event of a fault, be able to initiate a graceful shutdown of the system. This was key as a hard power down of the system could cause unrecoverable data corruption if the processor is performing a writeback operation during power off. A comparison of voltage supervisory IC's is shown in Table 12.

	LTC2953	MIC2786	MAX16122
V_{in}	2.7V-27V	1.6V-5.5V	1.6V-5.5V
Supply Current	$14\mu A$	$7.4\mu A$	$10\mu A$
Push Start	Yes	No	No
Push Reset	Yes	Yes	Yes
Programmable Delay	Yes	No	No
Package	$3mm^2$DFN	$2x2mm^2$ MLF	2x6mm WLP
Cost	$4.72	$0.95	$2.51

Table 12 Voltage Supervisor Comparison

The IC which most closely matches these requirements is the LTC2953 by Linear Technology. This IC has an input voltage range from 2.7V-27V as well as internal timers and circuits that trigger when the input voltage falls lower than a user defined level. Our team was able to acquire free samples as well, so the parts cost was zero.

Additionally, the supervisor is push button triggerable for both turn on and turn off. The typical application circuit with the timing diagram for start and shutdown is shown in Figure 25.

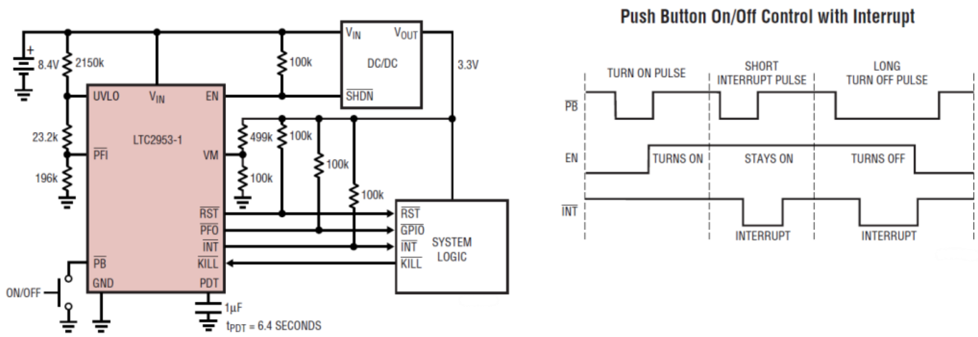


Figure 25 LTC2953 Push Button Timing

When the push button is pressed, the LTC2953 begins startup, pulling up EN and starting the watchdog timer configured by the capacitor on PDT. The default timeout is set to 64ms if left floating or additional time can be configured by $6.4s/\mu F$. When the push button is short pressed, the LTC2953 pulls INT low which can be used to request a shutdown from the main system. The long pulse duration for forced power down is determined by the same time constant $6.4s/\mu F$. Once the system has gracefully powered down, the KILL pin is asserted low and the LTC2953 asserts EN off, turning off the system.

The last feature of the IC is the voltage monitoring on UVLO. When the falling voltage on this pin goes lower than 0.5V the watchdog timer is started, and the INT is asserted LOW requesting that the system shutdown. Once the timer runs out or KILL is asserted LOW, whichever is first, the LTC2953 will then release EN shutting down the system. This watchdog timer feature is extremely valuable as it allows the under-voltage protection of the battery independent of the main system.

3.4.5. Motor Drivers

The motor driver IC is what powered the servo motors for the wheels of the TankBot. This component needed to be robust and able to drive an inductive load. The type of motor driver is called an H bridge. An H bridge is a circuit that allows a voltage to be applied differentially which lets the motor move in either forward or reverse direction. allows for Various IC’s where investigated including single and quad H bridge drivers. The comparison is shown in Table 13.

	L293(D)	DRV8842	L298	DRV8412	DRV8432
H Bridge count	Quad Half	Single Half	Dual Full/Quad Half	Dual Full	Dual Full
Max current per channel	1A(600mA)	5A	2.5A	6A	7A
Drive Voltage	4.5V - 36V	8.2V - 45V	2.5V - 46V	-0.3V – 70V	-0.3V – 70V
Internal Diodes	No(Yes)	No	No	Yes	Yes
Cost	$3.68	$3.62	$5.66	$7.01	$11.52

Table 13 H Bridge Comparison

Initial preliminary testing was done with the L293D driver, but future testing and design incorporated the DRV8432 Dual Full Bridge Driver. This selection will give the maximum, if needed, of 7A. But the system would need a current supply of 4A. Another addition to selecting this chip is that the chip came with 2 bridges that controled 2 DC motors. The following figure is the recommended schematic of the driver.

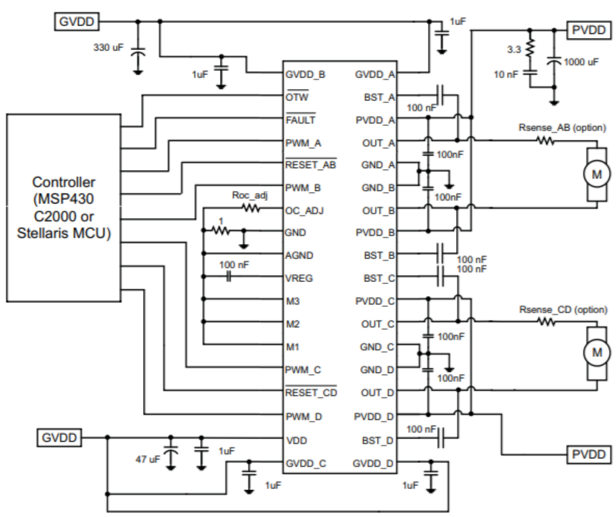


Figure 26 Driver Circuit of Two DC Motors.

The PWM signal will be driven at a 200 kHz frequency to keep the efficiency of the driver at about 95%. As for the microcontroller to maintain connections with the driver, there needed to be a great deal of general purpose input/output pins to control one such driver. Therefore, a microcontroller with a lot of pins are required. The TM4C123G was a candidate under consideration for this task.

3.4.5.1. Airsoft Guns

The gun turret is one of the most important systems present in this project as it allowed for the neutralization of the target once a target has been acquired. Choosing the right gun for the project was incredibly important; below are listed the top three considerations of the many considerations made for the project. The value of all these guns was determined from its weight, durability, user acclaim, cost, muzzle velocity, and capacity. The being said, it needed be low-weight, durable, reviewed highly by purchasers and users, low-cost, high-velocity, and high-capacity. Many guns were reviewed and considered, but as has already been stated, these are the top three choices, with the M4 being the ultimate decision due to its attractive specifications and reviews.

3.4.5.2. Pulse M74DP

The pulse M74DP is an airsoft gun with fully electric automatic and single shot modes. It's gravity-fed and bullet discharged travel at speeds matching 150 feet per second. It also holds 250 BBs, therefore it's an excellent choice for this project. It is low weight, but most of the gun is plastic, meaning it wears down quite quickly over time. However, in exchange it's quite thrifty at a mere $30.00 and it boasts a weight of a mere 0.75 lbs. However, most of the gun being plastic was a huge deterrent to the project, which strives for mechanical sustainability. Otherwise, this gun could have been a viable option for the project.

3.4.5.3. TACR71C

The TACR71C is an electronic airsoft gun with semi and full automatic modes. It carries 500 rounds and is gravity-fed. The muzzle velocity is approximately 200 feet per second. However, again, it is synthetic plastic all the way through, so not durable. However, the upsides of the gun are the price at $59.99 and a weight of a mere 2.00 lbs. This gun wasn't selected due to its plastic build not being durable, which, again, deters it from use since one constraint for the project pertains to mechanical sustainability. However, other than the lack of durability, it could have been a viable option.

3.4.5.4. M4 Airsoft AEG w/ Metal Gearbox

While there is a plethora of electronic airsoft guns available that have metal internal components (for durability's sake), the M4 was a good option for a couple of reasons, including a cost a total of $45.00. It boasts a muzzle speed of approximately 400 feet per second (the fastest of the guns presented), as well as a capacity of 300 rounds (the moderate of the two other proposed guns). However, this gun is completely metal and consists of a high-torque motor, making the gun attractive for this project's purposes. Furthermore, the gun runs at approximately 8.4V, which fit within the power limitations imposed by the battery. The major downside to this gun is its weight of 8.24 lbs., being the heaviest of the guns presented. However, most of the gun was stripped down, so the weight decreased during the stripping process. The motivation here was that this gun would normally cost $200.00, but Evike's website offers some guns under a category named the Bone Yard. This means that the guns are often malfunctioning in some way or another. However, the gun typically comes with high acclaim, which made it even more attractive to the group. An image of the gun can be found in Figure 27.

Figure 27 M4 Airsoft AEG w/ Metal Gearbox

3.4.5.5. Gun Decision

In the end our group decided to purchase an M4 from the Bone Yard. The main reasons we picked this gun over the others were the high durability of the interior components (being made of metal rather than plastic) and the inclusion of a high torque motor. The other guns were viable options, but their plastic, non-durable builds disappointed the group. Ultimately, a more expensive gun will reduce the number of times it will be replaced. The weight is the highest, which is a huge downside, but can be worked around. The capacity isn't the highest, but a capacity of 300 BBs is still quite good when weighed against the several other options that weren't discussed here. The muzzle velocity is the highest, which harkens on the potential accuracy of this gun. The gun was received and after an investigation it was found that the problem with the gun was purely aesthetic. This wasn't an issue since most of the gun was going to be stripped down for the project.

	Pulse M74DP	TACR71C	M4 Airsoft AEG w/ Metal Gearbox
Muzzle Velocity	150 ft/s	200 ft/s	400 ft/s
Capacity	250 BB's	500 BB's	300 BB's
Material	Plastic	Plastic	Metal
Power			8.4V
Weight	.75 lbs.	2 lbs.	8.24 lbs.
Cost	$30	$59.99	$45

Table 14 Airsoft Gun Comparison

3.4.6. Rectifier Diodes:

The rectifier diode is a key component in the design of the TankBot. Many parts such as switching regulators or inductive motors generate a back EMF that if unhandled would severely damage the sensitive silicon components such as MCU's. Therefore, rectifier diodes were used in parallel with these switching devices to rectify the reverse voltage generated before it can harm the system.

3.4.6.1. ES2J SMD Diode

The ES2J is a surface mount, glass passivated super-fast recovery rectifier diode in a SMB package. This diode can withstand up to 600 Volts across the diffused junction. Furthermore, it can receive up to 260°C for up to 10 seconds at up to $\frac{1}{16}$ in. from the case, making it ideal for soldering and reflow. It's a standard diode that our team already has an abundance already in possession. The ES2J is a multipurpose diode which can be used way to rectify incoming signals with a reverse recovery time of 35ns or as a power diode with an average current of 2A and surge current of 50A. Its role in the project was for general purpose rectification and fly back protection for switching regulators and motors. One of the main fly back protections it provided was for the airsoft gun which internally has an inductive load. It was used in conjunction with a driver circuit which controls the fire rate of the gun. The ES2J costs approximately $0.36-$0.39 but was acquired for $0.12 at the time of purchase.

3.4.6.2. BAT54GW

The BAT54GW is a planar Schottky doped with an integrated guard ring for stress protection, it comes in a SOD123 package with an average forward voltage < 400mV. This specific diode features low forward voltage, leakage current, and capacitance while be approved by the Automotive Electrical Council. This part was used as a feedback diode for the MIC series of voltage regulators as recommended by the datasheet.

3.4.7. Bipolar Junction Transistor

The bipolar junction transistor was the first silicon amplifier created by bell labs in 1947. The basic theory of operation is when forward current is applied to the base of an NPN BJT, it causes electrons to flow from the emitter to the collector. This flow of electrons creates a positive current in the opposite direction from collector to base. The relationship between base current and collector current is given by the classic formula $i_c = \beta i_b$ where β the gain of the transistor is. Typical gains for single NPN transistors are anywhere between 100-300

3.4.7.1. Darlington Pair

The Darlington pair is a special integrated circuit which utilizes two NPN BJT's in cascade. Current which enters the base of transistor 1 causes current $i_{e1} \cong i_{c1} = \beta_1 \, i_{b1}$ to enter the base of transistor 2. i_{e1} is therefore equal to i_{b2} and the output current $i_{c2} = \beta_2 i_{b2} = \beta_2 i_{e1}$ then $i_{c2} = \beta_1 \beta_2 i_{b1}$. This means that by using a Darlington pair, the gains of the transistors can be multiplied together allowing for incredibly large current gain on the output with a small base current input.

For the Darlington pair transistor, we tested and we chose the TIP110. The TIP110 is a plastic, medium-power complementary silicon transistor. It consists of a 2 Amp Darlington pair which can be used at up to 60 Volts on the collector-emitter junction and 50 Watts of power dissipation. At up to 1 ADC collector current, it can produce a gain of up to 2500. This transistor was tested in conjunction with a rectifier diode to allow for a switching mechanism controlled by the MCU's digital ports. The TIP111 and TIP112 were also worthy choices from the role of this BJT but were only unnecessary improvements upon the TIP110. It prices at approximately $1.3622 each. This BJT was used for the switching the gun on and off.

3.4.8. Motors

Motors provide the robot with the ability of translational motion as well as turning. However, they were also pivotal to the aiming mechanism of the embedded turret. So, the two major categories of motors analyzed below are differentiated by their use, rather than their design: locomotive or turret motors. As such, the choice in motors made below stemmed from user experience, performance, cost, and provider reliability. Direct testing of each motor was important to determining whether its implementation was feasible within this project. The authors found that some motors didn't live up to their initial expectations and that others needed to be considered.

3.4.8.1. Locomotive Motors

The selection of the motor was chosen by calculating the payload of the system, the weight of the project, and finding the motor with a torque greater than the stall torque that the weight of the system produces. The approximate expected distribution of the various masses of the parts is shown in the table below.

Weight Distribution of the Project	Measured Weight
Airsoft Gun	8.00 lbs.
Robot body	17.00 lbs.
Electronics & Sensors	3.00 lbs.
Battery	5.00 lbs.

Table 7 Robot Weight Distribution

The robot body consists of the frame and the motor mounted in their slots. The electronics and sensor comprise the Printed circuit boards along with its electrical components and the sensors used to get feedback to controllers. The total weight of the robot is 33 lb., 15 kg in mass. The motors of the robot would need to overcome the stalling torque of the system, which is the static friction between the wheels and the surface it is on. Assuming the system is rubber on dry asphalt, the frictional force is 132.4 N. Mecanum wheels were chosen and are explained in 3.4.9.3 and in 3.4.9.5. The Mecanum wheels have a radius of 50.0 mm, therefore the stall torque of the system is 6.6 Nm. Since there are four motors producing forces to move it forward, the needed torque for each motor needs to be greater than 1.65 $N \cdot m$. The following table are the motors that were examined.

Motor	Operating Voltage	RPM	Type of Motor	Torque (N•m)	Cost
Maker motor	12	50	DC - Geared	6.00	$69.99
Uxcell	12	25	DC - Geared	1.96	$32.99
GW370	12	7	DC - Brushed	4.90	$ 9.99
Siemens 2L147827144AB	12	55	DC - Geared	6.00	$49.99
Kenshi 29606	9	100	Stepper	1.21	$15.99
TMCG QSH6018-56-28-165	3.36	200	Stepper	1.68	$86.92
Mopar 05067591AE	12	90	DC - Geared	4.00	$40.00
Tetrix Max Torquenado 44260	12	100	DC - Geared	4.94	$29.95

Table 8 Motor Selection with Torque

The Mopar 05067591AE was selected was selected for several reasons. The characteristics evaluated were the amount of current drawn, the speed of the rotor, and the efficiency of the motor. The motor selected had a free load current of 1.02 amps and lock current of 7.98 amps. The maximum rated torque was at 4 $N \cdot m$. The current drawn can be modeled by the following function:

$$i(\tau) = \frac{(i_{max} - i_{no\ load})}{\tau_{max}} * \tau + i_{no\ load}\ (A)$$

This curve is the blue line in the plot of Figure 14.

The next evaluation is the speed of the motor and how it relates to the torque applied increases, the orange curve on the plot. The following equation models the speed of the motor as a function of torque.

$$\omega(\tau) = -\frac{\omega_{no\ load}}{\tau_{max}} * \tau + \omega_{max}\ (rpm)$$

The orange-dotted curved line is the outputted power of the motor. This is modeled by:

$$P(\tau, \omega) = \frac{\pi * \tau * \omega}{30}\ (W)$$

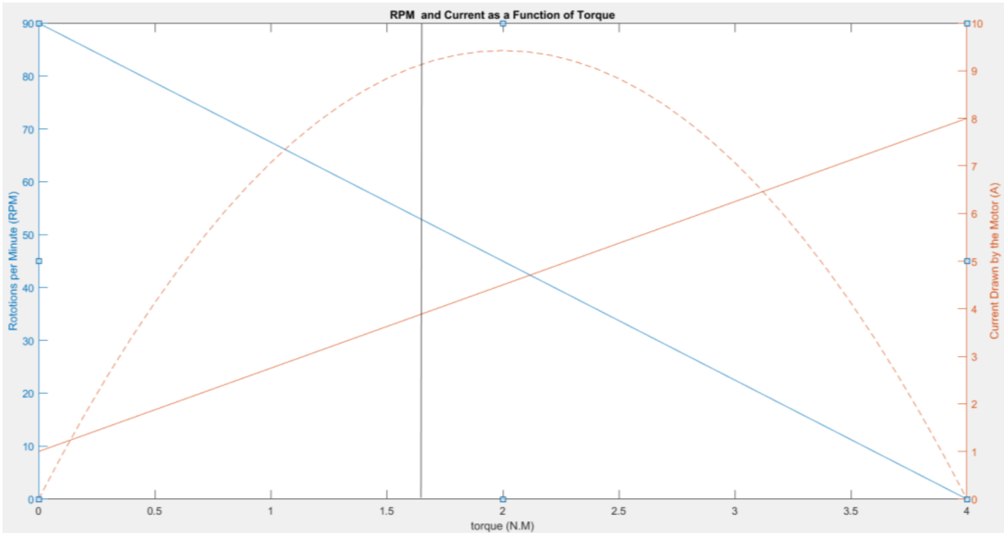


Figure 28 Characteristics of Chosen Motor

The black vertical line is the minimum torque required for each motor to perform. When comparing all the other motors, with cost considered, the Mopar 05067591AE was originally considered. When the robot was being built, the Mopar motor was inconvenient to mount to the robot. The Tetrix Max Torquenado 44260 motor was better suited to mount on the robot. Also, the Tetrix motor had the encoder built in to the motor. The following graph is the motor analysis using the equations used from above.

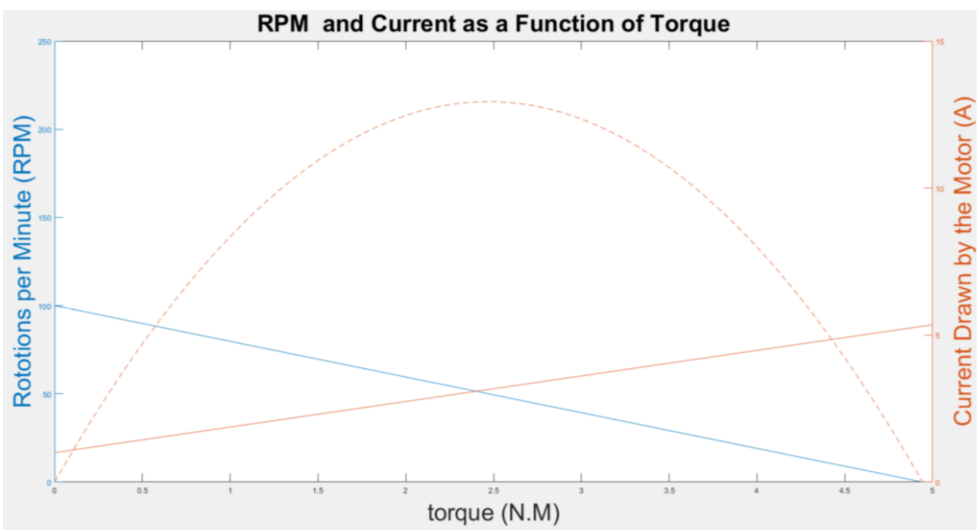


Figure 29 Tetrix Max Torquenado

3.4.8.2. Turret Angle Motors: VEX EDR 393

The VEX EDR 393 is a rotational motor produced by VEX EDR. It is the primary actuator used in the VEX EDR and is used to build rotational mechanisms in robot designs. It markets itself as being a useful motor for anything that needs to spin. It provides 60% more output power than the standard motor in the VEX EDR. It's constructed with strong steel internal gears, making it quite durable. It operates at 100 RPM and 1.67 $N \cdot m$ of stall torque. They also offer a 3-wire servo motor that can be used to face a specific direction. It operates at 100 degrees of rotation and can perform exact positioning. Both the 2-wire motor and the 3-wire servo were worthy contenders for use in the turret's angular control scheme. One of the only downsides was that a 9.6VVEX battery is typically required, but this project used its own battery sourcing, so testing was done and the motors were able to run on the battery using regulators to step it down.

Figure 30 VEX EDR 393 Motor

3.4.9. Uxcell 12V DC Geared Motor 10RPM

The Uxcell motor is a 12 volt geared motor. The gear ratio of this motor is 506:1. One benefit of this motor is that I has a built in magnetic encoder. With this selection, the system has been controlled like a servo. The motor changes to the sets or pulse width modulated signals and locking torque helps the weight load of the turret be fixed when not in use.

3.4.10. Wheels

The robot must be able to traverse terrain when it's on patrol or being relocated somewhere else. Therefore, four methods were brought up to accomplish this feat. These were the two tank tread system, the three tank tread system, the Mecanum wheels system, and the rubber tire system. All systems were evaluated to see if they are economical and that the implementation was feasible with less drawbacks. Another attribute to keep into consideration was the whether the wheels or tread can be easily maintained.

3.4.10.1. 2 Wheel Tread System

The first method of mobility that we considered was to have the robot be more based on tank treads, this would give the ability to traverse many terrains and not be hindered by it. This would give us two degrees of freedom since the robot would be able to move forward, backward and turn on the spot of its axis. This configuration also has some draw back. Like tanks, debris can get into the tread mechanism and stop the gears. Which also leads to the maintainability of these treads being less. These would be more complicated since the tread is now another element to be maintained apart from the geared wheel that connects to the motor. Also, the power to move the gears and the treads will increase the power consumption of the robot.

Figure 31 Dual Wheel Treads

3.4.10.2. 3 Wheel Tread System

The three-wheeled method would three sets of treads in a triangle configuration and would allow the robot to have three degrees of freedom. Also, this configuration would also be susceptible to debris in the gears. This system will also be heavily maintained, as like the two-wheeled tread system. Another drawback was that the with three tread mechanism, the power consumption would be even higher than the two-tread system.

Figure 32 Tri Wheel Treads

3.4.10.3. Mecanum Wheel

The Mecanum wheels are two disc plates connected by several rollers connecting across. The diameter of the wheels is at 100mm and are made of aluminum. Mecanum wheels allow for three degrees of freedom, in the x, y and θ axes. The Mecanum wheels will allow the robot to perform forward and backward motion, along with the ability to move side to side. Each wheel will have its own motor for its operation with independent signals from the MCU. Therefore, the microcontroller, or controlling unit must be able to accommodate many PWM signals, at a minimum of 8, to control the full movement of the Mecanum wheels. The Mecanum wheels degrees of movement are shown in Figure 33.

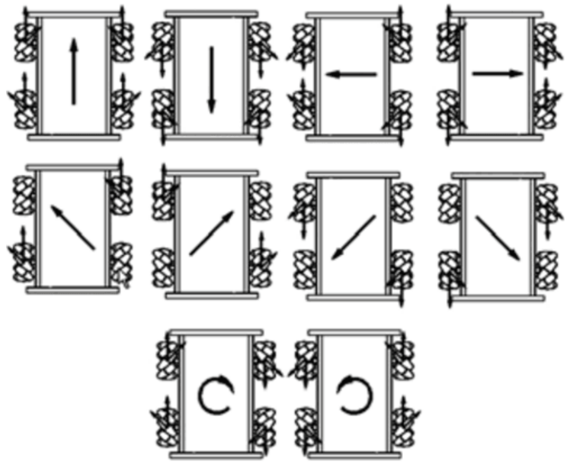


Figure 33 Mecanum Wheels Movement

3.4.10.4. Rubber Tire

The rubber tire would be configured like the Mecanum, with each tire getting its own motor, and it will give the robot 2 degrees of freedom. This also means that the microcontroller would need about 8 PWM signals to control the for wheels. These would be easy to maintain, since the wheels are a single object that can be pulled out easily. While it's still limited, this would also be the more economic decision and be cheaper than those above.

3.4.10.5. Final Wheel Choice

The wheels selected to move the project is the Mecanum wheel. The Mecanum wheels were chosen since the performance of three degrees of freedom allow the robot to be versatile. To control the system, a microcontroller with many signal generators will be required.

3.4.11.Rotary Encoder

There are three feasible methods to measure the number of steps a motor takes and how fast the robot is going. The first is to use an external optical rotary encoder. The other would be using a Hall effect sensor with a magnet in the rotor of the motor and measure its rotation. The use of hall effect encoders will be used over the optical encoder. The hall effect encoder is a electro-magnetic based encoder that measures the change in magnetic fields as the motor rotates. The output is noisy but can be overcome by shielding the connections.

The alternate solution is to use free rolling potentiometers that do not stop at an end. While this potentiometer cannot give us two signals, to determine direction, it would give a crisp signal since the oscillations will be at the resolution of the potentiometers. The drawback to this is that it will need to be read and an analog signal, since the output will be a ramp function. To get around this would be to use a Schmitt trigger comparator and set it to the allowable levels. Or, the use of a differential circuit to output a square wave rather than a ramp. Another drawback to a potentiometer is that the period of the signal will be very large and be too slow for it to update to the microcontroller.

The optical encoder does not have much noise interference since the encoder is sealed completely. This gives a clean signal. The only drawback is that it has mechanical parts that would have to be replaced since the part wears down and can break down. The price of each type is far apart, the optical encoder being the more expensive at $19.02, while the hall sensors are only $0.35. The free roll potentiometer is prices at $3.59. The optical encoder is still selected for reliability, despite increased price.

A specialized optical encoder is an incremental encoder. The incremental encoder has finer lined holes that light can pass and give a finer resolution to the number of rotation. These would be optimal for measuring position of the wheel, but its usage will be give greater accuracy. The following image is the incremental encoder used in the project.

Figure 34 Rotary Encoder Qauick 400p/r

Above all, the magnetic encoder was used to the project since it easily mounted to the robot and that the motors used in the project come built with magnetic encoders.

3.4.12. Robot Frame

The base of the robot is one of the essential parts of the project. This will house all the electronics and the turret. The base must be sturdy and be economical for this project. There were several considerations for this project. Some requirements of the base are that it must be within the limit of 2 feet (0.6096 m) x 2 feet (0.6096 m) x 2 feet (0.6096 m) volume, be able to carry the payload, which is the airsoft gun and be light weight to not exceed the limit of 50 pounds (22.6796 kg), weight includes other components.

3.4.12.1. OTTFF Robot Tank Base

This frame was the first consideration for the robot's base. While the base comes with treaded wheels, these can be interchanged with the Mecanum wheels and the motors can be upgraded to the ones selected. This base measure within an area of 8 inches by 7.5 inches and the height at 3 inches. The reason it was not selected is that the base's small size limits the motor placement and would need to have been raised to accommodate. Along with this, additional materials would have needed to be bought to accommodate the turret to sit on top and would have pressed everything together. This platform would have meet the requirements but is confined to too small of an area. The cost to obtain this frame would be at $59.99 and would consist of 5.99% of the budget allocated to this project.

Figure 35 Robot Tank Base

3.4.12.2. 4WD Mecanum wheel Mobile Robotic Platform

This robot platform was another under consideration. The platform was measured at 1 foot 3 inches (400 mm) by 1foot 2 inches (360 mm) by 4in (100 mm). The platform has enough space to place a sentry on top. The motors would have to be modified to the ones selected. The platform would have to be raised or make a hole opening for the motor to fit. The platform would be an excellent candidate to house the project. The only drawback of choosing this is that the price is at $770.00. This would consist of 23 % of the budget allocated to this project.

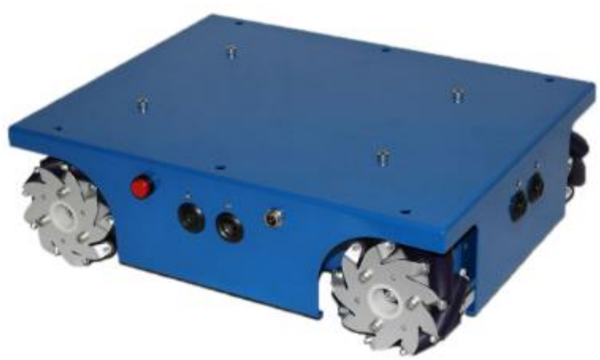


Figure 36 Mecanum Wheel Mobile Robotic Platform

3.4.12.3. Self-Made Base

The next option for this project would be to build a base from scratch. With this opportunity, the base can be sized up to the maximum dimensions, if needed. Also, this give this project flexibility when creating the base. If a self-made base were to be accomplished, the following would have to be taken into consideration when constructing. The surface area that has the sentry mounted, the housing for the motors and wheels and the space for the electronics to be kept. An open flat surface will be given for the sentry to be mounted on. This will consist of an 18 inch by 16 in surface. Another requirement will be housing the motors to the base. The motors will be connected to a gear box that connects to the frame of the robot. This robot consists of encoders to measure the speed of the robot. Therefore. the encoder must be connected in series with the motor. Therefore, the motor will contain a sprocket gear, along with the encoder, and be connected by a link chain. This will keep a study connection and have reduced slippage between the motor and encoder.

The wheel is then connected to the rotor of the encoder. The sprocket gear teeth for both the motor and encoder will be the same. This gives a one to one ratio between the angular velocity of the motor's rotor and the encoder's rotor. All four wheels will have this type of gear box. The small deviation of speed recorded from the encoder will be corrected with the microcontroller getting the feedback data. The cost to implement this is at $452.26.

3.4.12.4. Pitsco Tetrix Modular Kit

The Tetrix kit option is the use of prefabricated aluminum-based parts to build the frame. The use of this method allows us to make the frame to the needs of the robot without having to fabricate the pieces. To accommodate the turret, the robot will have to have the turret mounted on the robot or be built into the robot. The aluminum parts is that the kit consisted is the U-Bent bards. The bars also come with common hole sizes that allow rods to go through them. This would allow the bars to be mounted on wheels.

3.4.12.5. Platform Selection

The platform that was implemented to the project was the Tetrix kit.. This will give us more flexibility in both cost and design. While the original intent was to build a custom frame, time and resources were limiting this approach. The plan of building a custom frame was a novel idea, but the evaluation of this project would pertain to the electrical hardware and software. Therefore, the Tetrix kit has given us the flexibility to make any frame with the upmost reliability.

3.4.12.6. Modifications to the Base

There are several modifications that, in general, that will need to be done to the platform of the robot. The robot platforms selected were all aluminum-based frames. With this, the frame of the robot can act as a ground distribution node for all electrical grounds in our circuit. Then that ground can be connected to the negative end of the battery and act as a reference point. This should reduce the use of lines that go to negative end of the battery by at least one-third. Another part of the base that would be created is a housing for the electronics, the printed circuit board. The housing should keep the electronics, PCB, from getting wet or get impacted by the environment, while keeping it cool controlled temperature to have smooth operations. Therefore, a plastic box with slits to the side will suffice. The box will protect the circuit from environmental factors, such as rain or getting impacts from external forces. But with slits, this keeps the flow of air into the electronics. By adding a fan, this will also help control airflow and cool down the device.

3.4.13. Turret Platform

The turret platform is another module that will work in conjunction with the robot locomotion. There are several requirements to take into consideration. The sturdiness of the platform and the mechanism of operation. The price remains a huge consideration since turret platforms have wildly varying prices based on the quality of the platform and the additional features included. The selection of the turret platform will observe the difference between the pre-built models and a self-built model. A self-built model was chosen in the end for the reasons laid out below.

3.4.13.1. M74DP Airsoft Gun Mounting Kit

The M74DP Airsoft Gun Mounting Kit is a premade mechanism that uses two servo motors that control the angle of the airsoft gun. The kit does not include the airsoft gun or the microcontroller to operate it. But the airsoft gun we bought to make the turret and the microcontroller that will be used will come either from an integration with another controller, the motor locomotion system, or just a minute microcontroller that will communicate with the main central controller. There are some flaws with this platform that will not work for our project. For one, load on the mechanism is for light-weight loads. Since the mechanism demonstrates its at capacity at a small plastic airsoft gun, it is sufficient for this project. Therefore, the airsoft gun we planned to use would overwhelm the mechanism. So, the servos would need to upgrade to handle the payload of an aluminum air soft gun. The kit is sold at $99.95.

3.4.13.2. PhantomX Robot Turret

The PhantomX Robot Turret is the next prebuilt mechanism that is under consideration for the robot. While the project is using airsoft, rather than paintball, that can be replaced so that it is airsoft. The turret consists of two middle end servos and can carry a medium-payload. The airsoft gun that the project uses is going to be stripped down to the bear minimum. Therefore, the mechanism may hold the airsoft gun without overwhelming he servo motors. Another attribute is that the weight of the mechanism will not greatly affect the overall gross weight. The cost of this kit $209.95. This kit may serve as a dependable kit that will be used.

Figure 37 PhantomX Robot Turret

3.4.13.3. Self-Built (Pitsco Tetrix Modular Kit)

The next option is to self-build the turret platform. There are a few problems with the prebuilt that are overcome by being made specifically for this application. The basic construction of the frame will involve mounting the gun between four rails. A front-view of the frame will reveal a bottom-most, horizontal rail that will provide a general foundation for the yaw rotation of the turret. Perpendicular to this rail will be two vertical rails which serve to stabilize the turret in its angle position as well as holistic encapsulation of the gun itself. Further stabilization of the gun will be ensured by a rail that is perpendicular to all the other rails, where the turret itself will ride. This rail itself will be driven by the angular rotation motors, allowing for the aiming of the turret. A summary of this frame reveals a basic U-shape frame with a penetrating rail where the gun is mounted. The penetrating rail will be attached to a gear system that allows for angle control, and the U-shape rails will be attached to a gear system which allows for yaw control. This explanation of the frame design serves to highlight its advantages. First and foremost, cost of the frame is severely reduced. Rails and gears are generally quite cheap. As such, constructing from just the materials removes the overhead costs of labor construction or manufacturing. Furthermore, the frame can be built precisely as the group prefers and adjustments can be made on the spot to further improve the frame as needed. Say, for instance, one of the other frames exceeds the maximum volume requirement of the bot, a self-built frame can be reconstructed to smaller dimensions quite easily. Thus, adjustments such as these can be made far easier. One disadvantage of a self-built frame, though, is the aesthetic appeal. Given that nobody on the project is a specialized expert in frame construction, some aesthetic qualities of a pre-built frame are discarded by ignorance. However, application of the bot will not directly necessitate high-quality aesthetic appeal, so this disadvantage can be largely ignored.

3.4.14. ADXL345 Digital Accelerometer

The ADXL345 digital accelerometer is the perfect device to use to measure the current angle of the turret. It draws $40\ \mu A$ while measuring and $0.1\ \mu A$ while idling. It has up to 13-bit resolution for acceleration measurements. It has activity monitoring to adjust power consumption as needed and can be easily driven from an MCU port. The bandwidth and measurement ranges are selectable, and it can perform in a wide range of temperatures, making it quite usable near hot components. It has a $98.1\ \frac{km}{s^2}$ shock survival, meaning that the robot cannot damage it by bumping into something or by jostling it. Furthermore, it’s encapsulated in a $3\ mm\ x\ 5\ mm\ x\ 1\ mm$ LGA package, so the component is quite small. It can measure up to $157\frac{m}{s^2}$, which is more than sufficient for this project. As such, it meets the minimum requirements of this project.

Furthermore, it digitally communicates through SPI or I^2C with a 16-bit, twos-complement data transmission. It can measure changes in angle that are less than a degree, making it more than sufficient for the minimum range of the turret (30 feet). With this consideration of effective range, the device can account for position changes of a target within 3 inches. A functional block diagram of the accelerometer can be seen in Figure 39. The main things to notice here are that the 3-axis sensor is run through an ADC and digital filter, making the output of this device completely digital. This is valuable because it makes interfacing with the device far simpler. Furthermore, the ability for this chip to serially communicate makes it easy for the MCU to communicate with the chip. Including interrupt logic also furthers the simplification of communication with the MCU. Conclusively, this accelerometer is a good option for angular measurement because it functions as a sort of ‘all-in-one’ package for the sensor, taking weight off the MCU and allowing it to function entirely without direct operating instruction from the MCU. To that degree, the MCU needs only to interpret the results of the sensor and adjust the angle of the turret appropriately. The device can be used to find the angles of the sensor itself through the following equations:

$$\alpha = \cos^{-1}\frac{\hat{a}_x}{|\vec{v}|} \qquad \beta = \cos^{-1}\frac{\hat{a}_y}{|\vec{v}|} \qquad \gamma = \cos^{-1}\frac{\hat{a}_z}{|\vec{v}|}$$

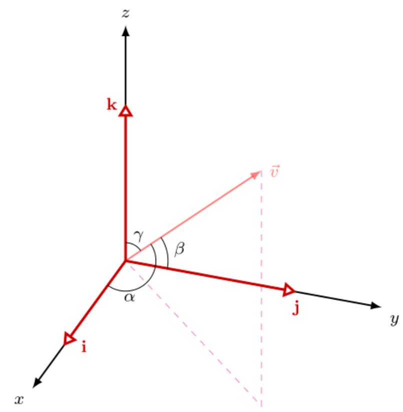


Figure 38 3-Axis Angle Detection with Acceleration Vectors

Where $\hat{a}_x\hat{a}_y\hat{a}_z$ are the component vectors reported by the sensor and $\vec{v}$ is the gravitational vector which always points down toward the earth. Proper mechanical placement of the sensor can greatly simplify the angular calculation by setting one axis, for example x, such that $\hat{a}_x$ will always be zero. Then the angle of the gun becomes simply

$$\phi = \tan^{-1}\frac{\hat{a}_z}{\hat{a}_y}$$

Then the obvious choice is to orient the sensor such that the acceleration vectors correspond to the planar coordinate system that will be used.

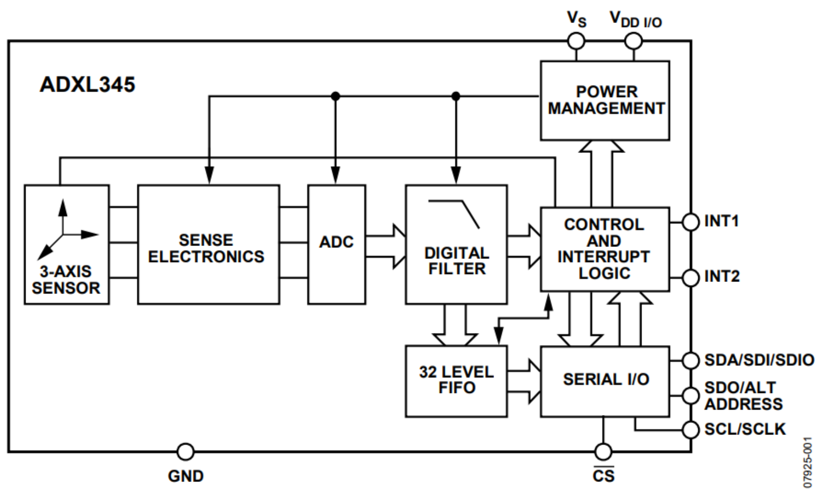


Figure 39 ADXL345 Block Diagram

3.5. Parts Selection Summary

Though the final parts have been selected, it is possible that during the implementation stage of the project, parts and designs will need to be changed. The final decision and summary of those decisions are detailed below. The mechanical components will remain as the products chosen, due to the lucrative advantages that they provide the requirement specifications of the project. These include cost, performance, and reliability, all of which are summarized below. Should any problems arise, we can look back and re-evaluate the part selection into the alternative candidates. The full testing strategies can be observed in sections 7 and 8.

Battery: Lead Acid.

Lead acid batteries where chosen as a balance between energy density and cost. Though the best performing chemistry is the lithium cobalt oxide, the cost for large amounts makes it not practical to use in this application. Additionally, the robot has be drawing large amounts of current to power the drive motors, lead acid batteries make a good choice due to their ability to source large amounts of current.

5V Voltage Regulator: MIC261201YJL-TR

The main reason the MIC261201YJL-TR was chosen is simply because it was free. Moreover, it meets and exceeds all the requirements needed for a 5V regulator in this project while performing as advertised.

3.3V / 1.8V Regulator: PAM2306AYPKE

The PAM2306AYPKE was chosen after careful considerations of power efficiency, cost, and output current requirements. Of the three regulators researched, the PAM2306AYPKE meet all the requirements while remaining cost effective at $0.71. Not only has this regulator been tested and verified to perform within datasheet specifications, it is proven to work in large deployments as it is what currently powers the Raspberry Pi 3b.

Voltage Supervisor: LTC2953

The LTC2953 which most closely matches these requirements is the LTC2953 by Linear Technology. This IC has an input voltage range from 2.7V-27V as well as internal timers and circuits that trigger when the input voltage falls lower than a user defined level. Our team was able to acquire free samples as well, so the parts cost is zero. Additionally, the supervisor is push button triggerable for both turn on and turn off.

Load Switch MOSFET: SiS413DN

The load switch MOSFET that was chosen is the SiS413DN. It has a low $R_{DS(ON)}$ is to keep the total power dissipation low. Also, if was one of the lowest priced MOSFETS where compared which met these requirements.

Our team has already acquired the SiS413DN but since the pinout and package of the DMP3013SFV is the same, if the need arises this will be our second purchase

Motor Driver: DRV8412

This Full bridge driver was chosen because of its ability more than supply the power we needed to drive our devises while being free due to being able to get samples for free from Texas Instruments. Other options had a higher max current, namely, the DRV8432, we did not have access to them in a reasonable time frame, as each order would take 4 weeks to receive from TI due to issues in their supply chain for this specific product.

Locomotion Motor: Tetrix Torquenado 44260

This motor was chosen because the availability and the output performance were desired to operate the robot. The output torque being at 2.45 N-m and that four of them will produce 9.8 N-m is enough to move the robot. The selection of the motor also meant that the operating voltage is at 12 volts and that the average current consumption when the motor is operating is 3.1 amps. Therefore, the motor driver selection have been meeting these requirements.

Wheels: Mecanum Wheel

The Mecanum wheel system was chosen to maximize our freedom of motion since it was the only system that boasted the 3rd degree of freedom, namely, being able to move on the diagonal, which is important while keeping the gun pointed at its target while also moving to a location. The cost of these wheels was generally higher than other options we had considered but we felt that the extra freedom was valuable enough to be considered over the others.

Robot Frame: Pitsco Tetrix Modular Kit

The platform used for this project is the self-made frame using the Tetrix Build kit. This gave the project the flexibility to accommodate changes to the electrical design as need be. The use of the Tetrix kit was also chosen because a kit was readily available from the Texas Instruments Innovation Laboratory. Now, one of the major disadvantages of a self-built platform is the aesthetic quality, but the group is generally unconcerned with this detail. The other major reason for self-building involves the adaptability. The group can re-build the platform as needed to perfectly suit the project at hand. This adaptability has allow for easier troubleshooting in the testing phases of the project.

Turret Platform: Pitsco Tetrix Modular Kit

We decided to self-build the turret platform using the Tetrix kit. This was the logical step since the base was built using the Tetrix kit. The original plan was to use a rotating turret. Since the wheels of the machine were the mecanum wheels, we fixed the turret and have the rotation of the turret be part of the mecanum wheels' movement. This simplified the load of having two rotational axis into one. Another benefit is that when using the Tetrix kit is that the modular parts also come with common hole sizes that will allow the robot to have addons as the project needs them.

4. Related Standards and Realistic Design Constraints

This section of the report will cover some of the many standards that were adhered to during the design, development and testing of TankBot in addition to the most important design constraints that we had to consider. Sticking to the standards and being aware of the constraints are necessary to successful design. By doing so, the project's interoperability will be increased. Future designers might choose to implement different schemes and components, and this will be made easier through adherence to the standards and understanding of the constraints.

4.1. Impactful Standards

Impactful standards detail the standards that most greatly affect the choices made in the final implementation of the project. These standards exist to allow for successful communication between engineers about their projects. They provide a sense of universality between the product being developed, those already existing, and those to be created in the future. Thus, adherence to these standards is important for effective communication between other engineers and to avoid any legal mishaps that could occur from negligence of aspects related to this project. Not every standard is mandatory by law, but it was important for the group to realize which ones are. Furthermore, the design constraints might reveal complications between the project and the law that will need to be eliminated before design can proceed.

4.1.1. List of Relevant Standards

All over the world developers are constantly innovating and creating in a race to be the next big thing on the market and make a ton of money. However, for an engineer to make sure that the next amazing invention he or she is creating will be applicable to the average Joe in any house in America, they need to know about the power supply, the temperature control, the local laws regarding safety, and many other things. If every engineer had to investigate the individual circumstances of each point of sale across the globe there could be no global trade, no innovation, and no creativity. To make these things possible, standards have been developed such that contractors building your house down the street and the engineer building the next HVAC system are both using the same outlets, voltage regulators and current supplies. Standards are the glue that holds the national and even global markets together which is why it is incredibly important to be aware of the standards that apply to each part of any project you may be working on so that your product can be used, and thus bought, by as many people as possible. Below are just a couple of the many hundreds of standards that could be applied to our TankBot project.

Standard	Name	Description
IEEE 1872-2015	IEEE Standard Ontologies for Robotics and Automation	Specifies the main concepts of Robotics and Automation plus defines the vocabulary for communicating knowledge between robots and humans.
IEEE 2700-2017	IEEE Standard for Sensor Performance Parameter Definitions	A common framework for evaluating sensor performance in addition to units, conditions, and limits of the sensors.
IEEE 1118.1-1990	IEEE Standard for Microcontroller System Serial Control Bus	Describes a serial control bus for inter-device interconnection of microcontrollers,
IEEE 208-1995	IEEE Standard on Video Techniques	Describes the method for determining a camera's resolution, specifically, where the fine details of an image are no longer captured by the camera.
IEEE 7007	Ontological Standard for Ethically Driven Robotics and Automation Systems	The standard establishes a set of definitions and their relationships that will enable the development of Robotics and Automation Systems in accordance with worldwide Ethics and Moral theories.
IEEE 1873-2015	IEEE Standard for Robot Map Data Representation for Navigation	A map data representation of environments of a mobile robot performing a navigation task is specified in this standard. It provides data models and data formats for two-dimensional (2D) metric and topological maps.
IEEE 1872.1	Robot Task Representation	Provide a robot task ontology for knowledge representation and reasoning in robotics and automation.

Table 15 Table of Standards

Each of the above listed standards follows the same standard formatting making them very easy to follow and implement. Many of the standards contain references to other several related standards that can also be applied to the project. Overall, the group will need to study each of the standards to increase overall adherence to the standards. While only summary of the standards is provided here, it should be noticed that the standards involve more specific directives on how to design with adherence.

4.1.2. Design impact of relevant standards

Engineering standards are incredibly important to make sure that regardless of who is using a product, they will have access to the proper power supply, equipment, or understanding to effectively make use of the item. Standards do this by setting a guideline for how things should be done such that each person who encounters a thing will be able to understand how it was designed and how that person can further their understanding of it.

The above standards were relevant to this project but in some cases the impact they have on the way we designed and developed TankBot will be minimal. One of these cases is the CV system, which is sourced from an open source library in an almost fully optimized condition, thus the need to adhere to the proper standards while altering or writing new code for this purpose was negated. Another system that will not be needing standards are the individual sensors we will be using, as these come straight out of the box with a datasheet and all we need to do is keep to the data sheet and the default settings of the sensor should be good enough for our project. If not though, we will need to revisit some of these standards to make sure that any changes we make to the sensors correspond to the standards listed above.

Places where the design standards basically told us how to act to maintain it to code include the USB connectors that will relay information between systems in our robot: USD has a very strict standard about how to use the connection, what kind of information can be sent over the channel, and even at what temperature the connection must be kept at in order to maintain working order.

4.2. Realistic Design Constraints

The realistic design constraints are the inhibitors to the project which exist from various sources. The ideas behind these constraints is to limit the creators in such a way that a high-performance product is made without endangering certain concepts surrounding the product. They exist to protecting the engineer who adheres to their guidelines and rules. As such, they provide various limitations on implementation and design of the bot. They must be carefully considered to understand what is and isn’t possible when creating the product.

4.2.1. Economic Constraints

Prior to beginning any project, engineering especially, one must consider the economic constraints placed upon you by whomever is determining your finances. In our case, there was no sponsor, professional or otherwise. This meant that our financial constraints are determined entirely in house. In section 9.2 we present a part by part breakdown of our decided upon budget including our discussion of what costs we were willing to part with and what was needed despite the costs. Overall, our budget was a big source of discussion at the onset of the project and indeed remains a source of discussion as the project continues to evolve. As needed, the budget will be stretched to its absolute maximum to provide a feature-rich, high quality bot at the time of the demonstration. Further minimization will not be intensely sought, to increase the quality of the project throughout design and implementation.

4.2.1.1. PCB Size Constraints

The cost comparison in section 7.3 details a comparison of various PCB vendors. One thing they all have in common is that their pricing scheme is only valid for boards less than or equal to $100\ mm^2$ or slightly less than $4\ in^2$. Once this threshold is passed, the cost per in^2 drastically increases. As such it is imperative that to maximize our purchasing power, we keep the board within this size dimension. Doing so might require using components that are smaller, which often comes with a larger price tag. However, considerations will be made as to whether a smaller component that costs more, or a PCB that costs more will be implemented to minimize total cost. It must be remembered that one of the requirement specifications of the project is to keep within the budget, so choices between more expensive components must be weighed carefully against more expensive PCBs.

4.2.2. Parts Constraints

As detailed in section 4.2.1.1, due to the maximum dimension limitation imposed by the cost constraint; an additional dependency is created. To keep the board as small as possible smaller parts will need to be used. Surface mounted parts will be preferred over through-hole parts. The closeness of parts also increases the interference between circuits such as high frequency power switching and signaling. As such, focus will be maintained on using smaller parts and not creating an incredibly tight and compact final board. This minimalization of size of the PCB further reduces the size of the total project, which is one of the requirement specifications. With the PCB taking up less room, it's quite possible that the robot can be made more compact without introducing thermal problems.

4.2.3. Thermal Constraints

The constraints described in section 4.2.1.1 and section 4.2.2 also create another constraint dependency: thermal constraints. Due to the small size of the board and the miniaturization of the parts, the amount of heat generated per in^2 will be increased. As such, a thermal design will need to be considered. Possible solutions include designing separate power PCB's as well as 2oz copper plating, heat sink usage on the hottest components, and careful spacing from components that get quite hot. By taking precautions in the PCB design, future troubleshooting of errors can be reduced, meaning that error-finding time is reduced. As such, the group will have more time increase the quality of the robot, which is highly demanded within this project.

4.2.4. Time Constraints

The time constraints of this project are mainly determined by UCF, given in terms of the two semesters we have, and other deadlines provided by the university. In the first semester we are asked to provide initial design parameters, get feedback from our professors, and update our design parameters before continuing with the design. After our design is approved we move to the planning stage, during which parts are ordered, tested, and re-ordered until satisfactory. In addition to physical parts, the time to develop the required software is allocated to this time. At the end of the first semester a preliminary design document is required. Moving into the second semester a process of development, testing, re-assessing, and re-developing begins until the final deadline which culminates in a final presentation and demonstration. Concurrent to all this is the student's personal schedules which will affect progress on each of the student's individual parts of the project. A date by date breakdown of our production schedule can be found in section 9.3. Time constraints will act as an absolute maximum and minimizing the time required to complete the project will not be intensely sought. Early completion of the required major aspects will yield time for extra relevant features which improve the overall quality of the project.

4.2.5. Environmental, Social, and Political constraints

The environmental constraints of this project were mostly limited to the disposal of various parts of the finished product, this includes disposal of the battery, which is lead-acid, and the machine parts, which include all sorts of heavy metals. The main exception to this is the energy efficiency of the power system which has become a hot button issue in the social and political areas. As such, disposal of parts will be carefully examined within part selection, as well as general energy efficiency.

To highlight the recent concerns over energy efficiency, consider NASA’s predictions on future climate change:

“Scientists have high confidence that global temperatures will continue to rise for decades to come, largely due to greenhouse gases produced by human activities. The Intergovernmental Panel on Climate Change (IPCC), which includes more than 1,300 scientists from the United States and other countries, forecasts a temperature rise of 2.5 to 10 degrees Fahrenheit over the next century.” [8]

These types of predictions have become more and more frequent, lately, leading to a surge in the market on energy efficient products that claim to save money over less efficient options. The group is aware of the environmental considerations behind increasing the overall energy efficiency of the bot and will take strides to reduce the overall power consumption of the bot (which, is one of the requirement specifications). In doing so, the battery will be changed less often and harmful effects that might be made to the environment because of the battery’s creation will be reduced. However, the user must take part too, in proper disposal of the battery. It is up to the designers to notify future users of this.

Another interpretation of environmental constraints applies to how the robot will have to interact with its environment. For the testing of the mobility systems we want the robot to be able to move on various surfaces and types of ground. This means that the robot will have to be able to navigate not only smooth indoor floors, but also concreate parking lots, all the way up to the dirt and sand that make up most of the outdoor space in Florida. Throughout the testing and re-design phases of the project, emphasis will be placed on ensuring the bot’s ability to navigate all general terrains effectively.

Regarding social constraints, there has been pushback against the use of AI in warfare due to the ethical question of allowing machines to make life and death decisions when they have no moral conscience to weigh the decisions against. Legally though, there has been no decisions on this issue, so this issue is of less importance to us, especially as our main consumer, military and security personnel, are already solidly on the side of allowing AI into the realm of war with little to no restrictions. Furthermore, the team believes that reducing the risk to human lives within the critical situations that this robot would perform in has a positive effect on society at large.

Other social and political constraints include the gun involved within the project. Using a real gun would cause many hurdles for the group to jump through and would prevent demonstration due to the laws against guns on school grounds. These laws have most definitely become more intensified considering recent school shootings. In response to this, an airsoft gun will be utilized, but certain rules must still be maintained on its transportation and the demonstration that will be inevitably performed. Concealing the gun throughout transportation of the bot and the gun itself in the preliminary building stages will be of the utmost importance to prevent unnecessary interventions and misunderstandings with law enforcement.

It is important to highlight here that this type of project falls under a category of objects named ‘lethal autonomous weapons’ (LAWs) if ever implemented with an actual semi-automatic or automatic gun. As of the time of this report, no law currently forbids the design, implementation, and general creation of LAWs. However, many standards exist providing guidelines behind how they should be created. One of the most important harkens on the concept of human judgement over the use of force. Essentially, humans should be allowed to judge whether the robot should use force in a situation or not. In this project, the robot will have a standby mode, as well as a follow mode. In both modes, the robot will not shoot any targets unless explicitly instructed to do so (by, for instance, switching the robot’s mode). In that sense, the group has adhered to this concept in the creation of the robot. One critical example of this concept in actual application involves DoDAMM’s Super aEgis 2 (see section 3.1.6.). DoDAMM’s turret takes quite an extreme stance on this. Upon thermal detection of an unauthorized target within the secured zone, the turret will not fire without clearance from a human whose duty is to monitor the turret’s decisions. As such, while the turret can act with autonomy and fire at will on thermally sensed targets within the zone, human judgement over the use of force can be exerted instead. This human judgement was adapted in by request of stakeholders who planned to use this weapon to guard secured zones.

4.2.6. Ethical, Health, and Safety constraints

Ethical constraints for this project include but are not limited to that of originality, meaning that while work like this project has been done before, using without citation or copying any of the code, research or documentation of another group or company would be highly unethical. To prevent this sort of behavior, our group must be incredibly conscious of what our sources are and make sure to cite and give credit to all sources drawn from during the project. In addition, making sure that all our work is original is imperative to a successful outcome.

As the result of the project is a machine that is potentially capable of taking lives, there are quite a few restrictions that conveniently line up with the health and safety of users and those on the other end of the TankBot. Furthermore, the project itself requires implementation of some type of gun which must adhere to local and national laws for demonstration in the final presentation of the project. The issues are pivotal to even considering the design of a robot whose intention is to kill, and as such must be carefully justified.

One of the ways our group intends to prevent injury during the testing and operation of the robot is to default the robot to a “Follow” mode, wherein the tracking, targeting, identification, and motion systems are all still active, but the firing system is disabled such that it cannot start shooting on start-up. Other modes defining the activities the robot is capable of will keep the robot from accidentally harming people during testing and development.

Another object of consideration when talking about safety is that of the testing procedure, to make sure the robot can hit a moving target, we will of course have to move a target for it to shoot at. This could be dangerous for whomever is or is moving the target. To mitigate the risk associated with this specific line of testing we will be implementing the use of safety goggles, in addition to heavy clothing and closed toed shoes to lessen the impact that the bb rounds have on the target.

One further ethical and health issue related to the bot is whether the bot should necessarily kill the target or incapacitate it by (by shooting the leg of the target). However, being that the bot is intended for defense use, absolute neutralization of the target could be desired and required. Future implementations of the bot might seek to disarm instead of killing the enemy target, but current implementation of the bot will seek a sure kill on the target. This is justified by the key audiences being that of critical situation law enforcement and military. For civilian audiences, the sure kill aspect of the robot can be interchanged for simple disarming or incapacitation.

To highlight the ethical issues surrounding autonomous weaponry, consider Sharkov’s argument written in Newsweek:

> “In an open letter to the U.N., Musk, Google’s Mustafa Suleyman and 114 other specialists from 26 countries called for a ban on autonomous weapons.
>
> The experts, all of whom work in the development of artificial intelligence, robotics and relevant industries, warned that the possible arrival of killer robots in the ranks of world militaries could be as significant to warfare as gunpowder and nuclear arms.” [9]

This article not only highlights the ethical arguments surrounding autonomous weapons that could kill, but their effect on modern warfare. The invention of gunpowder and nuclear arms didn’t only carry the ethical implications of nearly sure death, for their time, but brought with them the press for countries to commit acts of war due to their new-found confidence in the, then, modern weaponry. The promotion of autonomous weaponry, in our day in age, could do the same. It isn’t impossible that countries withholding acts of war out of lack of confidence in technology might find this technology to be the tipping point of confidence, thus spurring war. However, a viable counterargument is the fact that nuclear arms carried far more lasting implications of eminent destruction than that of autonomous weaponry, such as turrets. So, the lack of confidence that potentially warring countries might have will probably not dissipate with the promotion of autonomous weaponry due to the continued fear of nuclear weaponry.

Rosenberg and Markoff of the New York Times further reports:

> "Beyond the Pentagon, though, there is deep skepticism that such limits [referencing human intervention on acts of murder] will remain in place once the technologies to create thinking weapons are perfected. Hundreds of scientists and experts warned in an open letter last year that developing even the dumbest of intelligent weapons risked setting off a global arms race. The result, the letter warned, would be fully independent robots that can kill, and are cheap and as readily available to rogue states and violent extremists as they are to great powers." [3]

This only furthers the argument by reinforcing that even the worst of the autonomous weapons would set off a global arms race. However, as mentioned before, an easy counterargument would be the fear of nuclear weapons. Furthermore, this project will not be implementing complete and utter autonomy to the robot. The robot will not be able to fully and comprehensively 'think', per say. More specifically, the robot can 'see' and make decisions from this vision concerning whether the target should be fired on based on a list of criteria. Humans set these criteria, which doesn't give the robot full autonomy and control. It is more of a perceived autonomy due to the robot's ability to make critical decisions in critical situation. As highlighted before, quick human judgement might not be as reliable as judgement made by a machine. Furthermore, a machine can be adapted off hindsight situations. If the robot has made a decision that is deemed faulty, further research and work can be done to adapt the robot to make a better decision next time. While this can still be done with humans, an emotional factor still exists that can inhibit immediate and precise action.

The entire group is fully aware of these ethical arguments and will leave future adaptation of the decision-making process of the robot up to future implementation, stakeholder interest, and law. The group is critically interested in showing that a decision-making process can be implemented within the autonomy of the robot to carry out simple guidelines. In doing so, the group shows general success of the design of the robot.

4.2.7. Manufacturability and Sustainability constraints

The manufacturability of this project is an important consideration because by our own expectations the robot needs to be portable, repairable, and durable to be considered a success. The reason behind this stem from the potential users. In a military or critical incident situation, an expert on the robot might not always be available, so personnel must be able to transport the robot without difficulty. As for repairable, a designer should not be required to execute repairs on the robot. Schematics should be in place throughout this report to guide a future repairer on the various elements of the robot, so that they can, in effect, repair it without being an expert on every individual component. Finally, the durability affects how often it must be repaired, so that increasing durability means decreasing the frequency of required repairs. As such, the robot becomes far more valuable to the market and the potential users don't get frustrated with repairs.

To be portable the size and weight of the robot must be tracked to make sure neither exceeds our expectations. However, the size and weight are a part of the requirement specifications, so by meeting the specifications, this constraint will be simultaneously met. In addition to the overall size and weight being tracked, the robot should be dismantle-able, and each of the individual parts must not be unwieldy or hard to handle by a single person. This is important because it affects the deploy-ability and thus mobility of the robot. This also ties into the ability for the robot to be repaired on the fly but by being able to dismantle the robot, a user can isolate problems and get closer to what may be the source of an issue rather than just looking at the hard metal outside. In that sense, future repairs can be done by thorough troubleshooting of each embedded element individually, which will decrease the time required to repair the robot. Furthermore, it reduces the chances of the robot being holistically discarded because an issue cannot be found.

Durability is also important so that the robot can be used on more than one mission or assigned to a duty for a long period of time, durability is affected by many things, including battery life, heat dissipation, and physical sturdiness. The battery life is discussed in another section of this report as is heat dissipation, so lastly, we have the physical sturdiness. In order make sure the robot is physically sturdy we will test each system of the robot thoroughly by bumping and jostling it while it is running through its processes. Thus, we will be able to ensure the durability of the robot to the satisfaction of our advisors.

Objects that will require maintenance over the life of the robot would be the mechanical parts, such as that motor being replaced when the brushes are worn down. Other things that require maintenance would be greasing some gears to lessen that wear and prolong any advancements of rust when gears wear down. Furthermore, the battery must be periodically replaced, as well as the bb rounds. As such, continual, periodic maintenance is required for the overall success of the robot and must be noted by potential users. The designers, then, must make the periodic replacements easy to perform, so that experts aren't required for simple, repeated tasks.

5. Hardware Design

This section details the design processes and choices implemented in the design of the final product to be demonstrated. The overall system will first be discussed at a high-level to explain the basic structure of the system at hand. Then, specifics will be given on the design of the various interconnected parts. The goal is to relate to the reader how the designs were created and implemented. Careful consideration of constraints, standards, and requirement specifications were made in the design processes discussed below. Furthermore, implementation of the parts to their maximum capabilities related to the performance of the bot is investigated.

5.1. Overall System Design

The block diagram shown in Figure 1 on page 10 describes the hardware control system that was used to implement the TankBot. The master control system works as the hub for all communication and decisions across the various subsystems and block components of the system.

The motion control system can be seen on the right side of the diagram. The encoder feedback controls where exactly the turret is aiming, to accurately aim the turret at the correct target. The motor adjusts as needed controlled by the TIVA microcontroller.

In the bottom of the figure, the wheel controls can be seen. Mecanum wheels where implemented, which require independent motors for each wheel. These motors are controlled by a motor control subsystem. Thus, communication takes place between the wheels and the motor control subsystem, then between the motor control subsystem and the master control system. The motor control subsystem acts as a buffer between the master control system and the wheels.

At the top of the figure, the wireless link system for communicating with a computer can be seen. The purpose of this set of block components is to allow the robot to interface with humans to some degree. Users can see through the eyes of the bot, so to say, which can allow users to possibly understand the motivations behind the decisions of the robot, as well as the process of the decisions. It also allows for wireless control of the robot.

To the right within the figure, the CV, LiDAR, and gun fire control subsystems can be seen. The CV operates on a camera which is interpreted and controlled by the master control system for universal control of the robot in its decision-making processes. The distance LiDAR allows the entire system to know when a target is in range for shooting. The gun fire control system works as a buffer between the master control and the gun, communicating with the master control system on when to fire and when to idle.

5.2. Power Subsystem

The power supply subsystem is the circuits which turns the raw battery input into usable power for the entire system. It was arguably one of the most important as a bad power system will not allow the robot to function. The estimated load breakdown for the robot is shown below in Table 16. These values are the maximum estimated load requirements. To estimate the power requirements of each piece of the TankBot, various sources were used including datasheets. When the datasheet information was not available, experimental data was used such as measuring the continuous current draw of the motors.

Volts	Watts	Amps
1.8V	0.495	0.275
3.3V	5.302	1.61
5V	6.5	1.3
12V	324	27

Table 16 Maximum Load Breakdown per Rail

For both the 5V rail and 3.3V/1.8V dual rail power distribution systems, switching regulators will be used instead of linear regulators to maximize efficiency. Since switching regulators require an inductor to provide stabilized current during PWM operation, a proper inductor must be calculated. To choose the proper inductor for each regulator, the general formula is used throughout this section.

$$L = \frac{V_{OUT}\left(V_{IN(MAX)} - V_{OUT}\right)}{V_{IN(MAX)} * f_{sw} * \Delta I_{OUT(MAX)}}$$

Since ΔI_{OUT} is not known as L is not known an estimation is made based on a percentage of $I_{OUT(MAX)}$ which is usually set to 20%.

5.2.1. Battery Protection Circuit

The first step in designing a power system is the power source, for this application we used a single lead acid battery rated at 12V and 7A for 20 hours. For lead acid batteries, the cutoff point where the battery should no longer be used is 11.6V to avoid permanent damage to its recharge life.

Therefore, the first circuit to be designed is a cutoff circuit which cuts off battery power in the event the battery voltage drops below a threshold voltage, in this case 11.8V which is slightly higher than the minimum to act as a buffer. This circuit is shown in Figure 40.

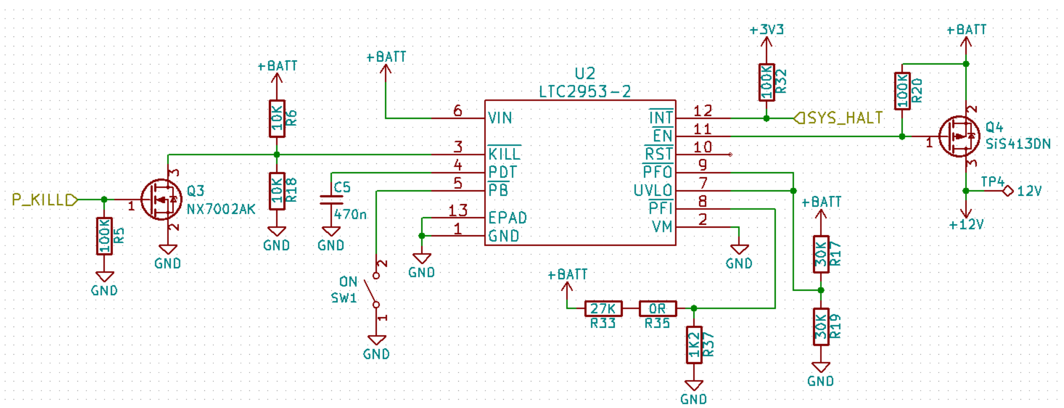


Figure 40 Battery Low Voltage Cutoff Circuit

The main component is the LTC2953 push button on/off controller with voltage monitoring. The main features of this IC are the ability to monitor the system voltage and assert INT low requesting a power down from the main processor in the event UVLO falls below 11.8V. This also starts the hard power down timer configured by C_5 currently set to 3.5 seconds to force the main power off in the event the processor is stalled. The timing formula for the power down timer is given by:

$$t = 6.4C_5$$

Where the value of C_5 is in μF. Additionally, a power down can be requested by pressing SW2 or a forced shutdown can be initiated by holding SW2 down for longer than 3.5 seconds.

When the processor initiates a software shutdown, once cleanup is complete P_KILL is asserted high. Q_4 then pulls KILL low and cuts the main power requiring a button press to power on again.

The next component is the high side P-FET which controls the connection from the battery to the rest of the circuit. Since the entire system load will pass through the P-FET, it needs to be chosen such that the power dissipation is low which translates to a low $R_{DS(ON)}$. The SiS413DN P-channel MOSFET from Vishay which has a max $R_{DS(ON)} = 9.4m\Omega$ which is used to reduce power dissipation across the device. More details on the specifics of this MOSFET can be found in Section 3.4.

5.2.2. 5V Power Distribution

The next power circuit is the 5V rail. This circuit needed to be able to source enough current to power all the sensors and controllers in the robot. The processor alone uses ~1A under full load so the 5V rail needs to supply at minimum 2A of current with no voltage drop. For this task we chose the MIC261201 which can supply a maximum 12A of continuous output current. The 5V rail schematic is shown below in Figure 41.

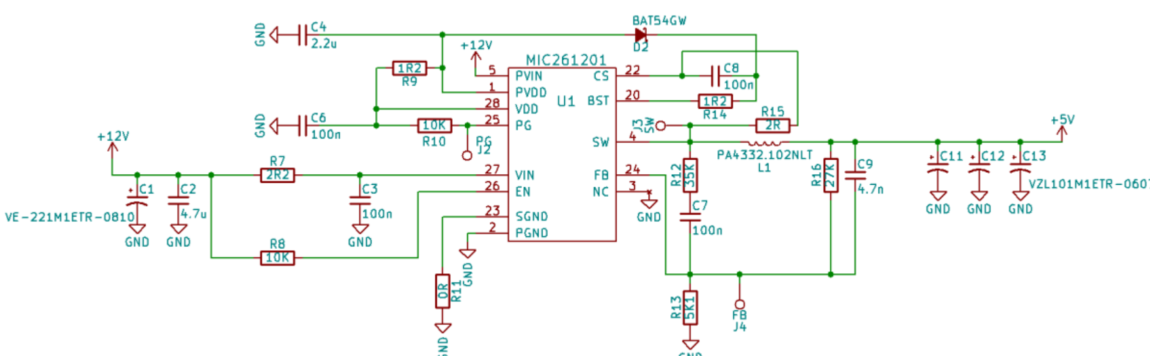


Figure 41 5V Rail Regulator Circuit

Additionally, this regulator supports a "power good" function. The PG pin will become open and drain low when the voltage output has become stable. This function will allow for the fine control of the other control systems to only start when the voltage output has stabilized if needed.

One of the benefits of this regulator is that it is advertised as “any capacitor stable” meaning it can work with both low ESR multilayer ceramic capacitors or high ESR aluminum capacitors. Normally low ESR is preferred for lower voltage ripple however using a bypass capacitor or ripple injector allows for the use of high ESR capacitors.

Since ripple on the feedback pin is required for proper operation, even though aluminum capacitors will be used, a ripple injector was used in tandem for optimal performance since ESR in parallel add in admittance, lowering the effective value.

To design the ripple injector, the following formulas where used. First to see if a ripple injector is needed, the ripple voltage at the feedback pin was calculated.

$$\Delta V_{FB(pp)} = \frac{R_{13}}{R_{16} + R_{13}} * ESR_{C_{OUT}} * \Delta I_{L(pp)}$$

The equivalent ESR is given in the datasheet as an effective impedance and dissipation factor. From these two values the effective ESR can be derived via the following formula.

$$DF = \frac{ESR}{X_c} = ESR * 2\pi f C$$

Solving for ESR we get the following formula.

$$ESR = \frac{DF}{2\pi f C}$$

With $DF = 0.14; f = 120Hz; C = 100\mu F$ the ESR resolves to 1.85Ω per capacitor. The total $ESR_{C_{OUT}}$ is then the sum of the admittances of each individual capacitor, then $ESR_{C_{OUT}} = 0.619\Omega$.

Since we do not know the exact value of ΔI_L we must make an estimation based on acceptable values, it is quite common to use a maximum $\Delta I_L = 20\%$ therefore, that is what we will use for this calculation. Plugging in all these parameters into the first formula results in $\Delta V_{FB(pp)} = 0.0458mV$. The MIC261201 required a minimum ripple of 20 mV so an injector is indeed needed.

To design the ripple injector, the assumption is made that the time constant τ is greater than the switching period, $\frac{1}{f_{sw} * \tau} \ll 1$. Then since R_{13} and R_{16} are in the $K\Omega$ range, at 1nF-100nF capacitor can easily satisfy the time constant requirements A 100nF capacitor is used to achieve the required time constant requirements.

Then using the following equations to solve for R_{12}. η is the efficiency of the regulator given as approximately 92% at 2A.

$$D = \frac{V_{OUT}}{V_{IN(MAX)} * \eta}$$

$$\tau = \left(R_{16} || R_{13} || R_{12}\right) * C_9$$

$$K_{div} = \frac{\Delta V_{FB(pp)}}{V_{IN}} * \frac{f_{SW} * \tau}{D(1-D)}$$

$$R_{12} = (R_{16}||R_{13}) * \left(\frac{1}{K_{div}} - 1\right)$$

R_{12} is therefore found to be $\cong 35K\Omega$.

5.2.3. 3.3V and 1.8V Power Systems

The last power circuit is the dedicated power rail for the bcm2837 on the Compute Module 3. The 3.3V rail provides power to the majority of I/O while the 1.8V rail is required but not used for any logic. For this we chose the PAM2306 to power the Compute module. It features dual outputs at 3.3V and 1.8V and can source up to 1A per channel. This regulator is what is also used commercially on all Raspberry Pi boards, so it is proven to work. The schematic for these rails are shown below in Figure 42.

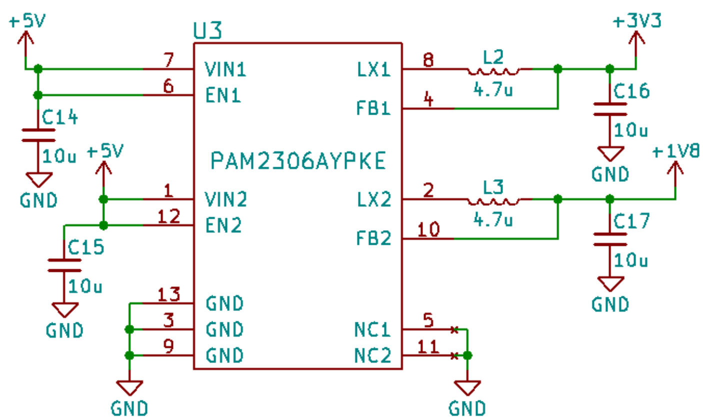


Figure 42 3.3V and 1.8V Dual Regulator Circuit

Inductor selection is found by using the standard inductor formula in section 5.2 on page 13. This gives an approximate $L = 3.74\mu H$ for 3.3V and $L = 3.84\mu H$ for 1.8V. since both values are not standard inductor ratings, the next highest value at $4.7\mu H$ is used. For ESR considerations, low ESR ceramic capacitors are used as the datasheet does not specify if aluminum high ESR capacitors can be used. Additionally, since this circuit will power the "brains" of the TankBot, a stable power source is required and low ripple on the power rails is needed.

5.3. Communication Subsystem

The communication subsystem for the TankBot consists of high speed SDIO, CSI, and USB connections for the Wi-Fi, camera, and USB port respectively. As such, the design of the transmission lines for SDIO are unbalanced and needed to be designed such that the lengths of the lines are relatively equal compared to the wavelength of the transmitted signal to avoid significant timing issues. For the differential pair transmission lines, they needed to be designed at a specific characteristic impedance such that reflections in the transmission lines are minimized. For the CSI and USB communications, differential pairs of characteristic impedance 100Ω and 90Ω were needed respectively.

5.3.1. Differential Pair Calculations

To design the differential pairs, ADS LineCalc was used to calculate the exact dimensions of the differential pairs. CPW lines were chosen over edge coupled microstrip lines since it allows for a lower differential impedance with thicker lines, to avoid unrealizable dimensions. Figure 43 shows the dimensions required for a differential CPW transmission line, for a 4-layer FR4 board with a $\epsilon_r = 4.4$ and a dielectric thickness of 0.15mm between the top and first inner plane, the dimensions in mm where calculated to be:

USB $W = 0.35; G = 0.2; S = 0.1; Z_{diff} = 90\Omega$

CSI $W = 0.24; G = 0.2; S = 0.11;\ Z_{diff} = 100\Omega$

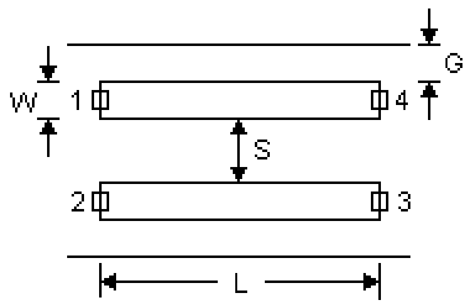


Figure 43 Differential CPW Dimensions

5.3.2. Signal Routing

Routing of signals was an important part in maintaining the integrity of the signals between devices. Arguably of equal importance as correct transmission line design. A poorly routed transmission line even designed correctly will not perform correctly. To route the differential signals, the shortest path is chosen such that the length and bends of the individual transmission lines is minimized.

5.4. Thermal Design

This section describes the hardware engineering that relates to the management of heat produced by the various components in the system. Parts such as switching regulators, processors, drivers, and dynamically switched MOSFETs all produce heat that must be managed properly.

5.4.1. Compute Module Heatsink

The Compute Module features a fully integrated SoC that contains both a processor and GPU in a small $14mm^2$ package. Since the TankBot is doing heavy video encoding and video streaming to the remote-control program, the CPU usage is at 100% most of the time. As such, the SoC becomes very hot to the point that the chip will throttle itself, reducing core clock to reduce temperature. To prevent this, a heatsink was needed to efficiently transfer this heat from the chip to the ambient environment. Unfortunately, we were unable to find the specific $R_{\theta JC}$ junction-to-case thermal resistance. We tried both contacting the manufacturer SoC Broadcom or the device manufacturer but were unable to ascertain any information. Thus, a custom Heat-Sink will still be used based of an analysis of current commercial heatsinks for this SoC.

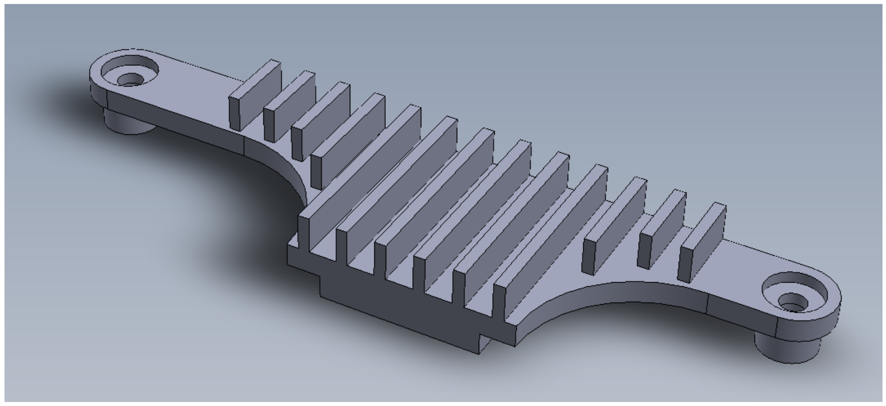


Figure 44 Compute Module 3 Heatsink

Figure 45 Manufactured Heatsink

Figure 44 shows the design of the custom heatsink and Figure 45 shows the completed heatsink. Unlike other commercially available heatsinks for this processor, this utilizes the in-built mounting holes to secure the heatsink instead of glue. Not only does this provide better thermal performance as glue is a poor heat conductor, it allows for the easy installation and removal of the heatsink with screws. Thermal paste was used between the heatsink and SoC to further improve thermal performance.

5.4.2. MIC261201 Thermal Design

The MIC261201 being a high efficiency regulator creates very little heat. Additionally, with there being an exposed thermal pad directly connecting the junction to the PCB ground plane allows for minimal thermal considerations if the recommended layout is adhered to. From the datasheet, at 3A, the power dissipation for $V_{IN} = 12V; V_{OUT} = 5V$ the power dissipation is approximately 0.75W.

5.4.3. PAM2306 Thermal Design

The power dissipation formula is given in the datasheet as 1.66W maximum. Since the power dissipation is low enough that a heatsink is not required, the IC can be connected to the PCB ground plane through thermal vias to dissipate the heat throughout all the ground plane layers.

5.5. Motor Subsystem Design

The motors need a driver to control the various operations needed to operate the motor. The following circuit is the driver for a motor to have bidirectional movement.

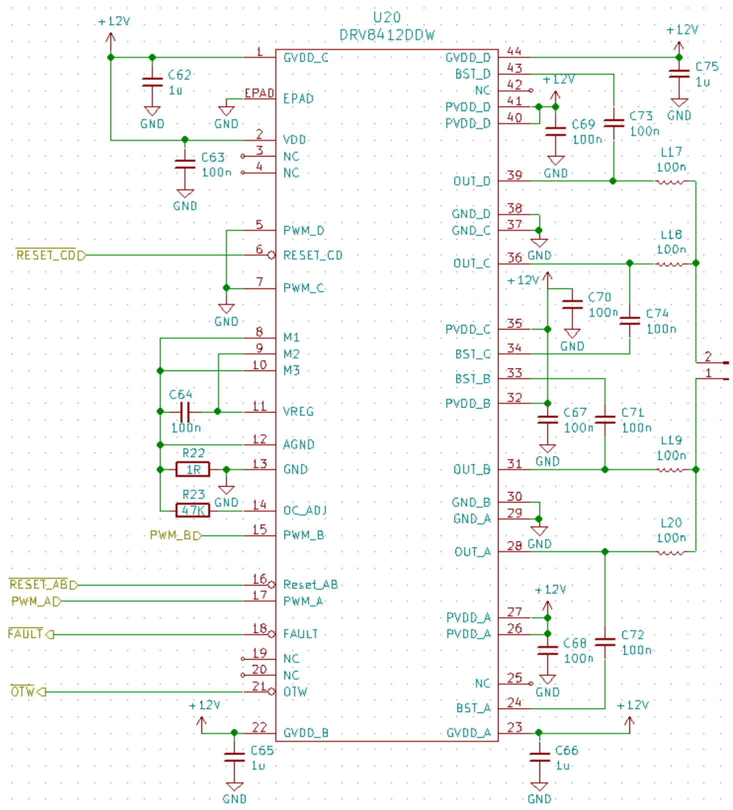


Figure 46 DRV8412 Driver Circuit

The DRV8412 is a driver that was used in this project. The main frame of the integrated circuit remains the same, except it comes equipped with over current protection and fault sensing. The connection of this circuit will be that same as the L293D. The pins of Output_1 and Output_2 corollate with the pins of the of OUT_A and OUT_B of the DRV8412. The connections from the DRV8412 are mostly to use the built-in features such as overcurrent protection and the fault chip line. Another thing is that the DRV8432 does not need diode protection for the motors since it comes inside the chip when it was manufactured. With the L293DD, the use of fast response rectifying diodes would need to be used on the motors end. Another requirement that the DRV8412 will need is for thermal dispersion. At the current draw of 5Amps, the chips itself could not cool itself without external support. To reduce heating when it operations, the circuit has a pad underneath with wave of cooling the chip. The PCB where its mounted is would need large copper areas and vias on the chip to disperse its heat. The following figure shows the five pads allocated to disperse the heat of the motor driver.

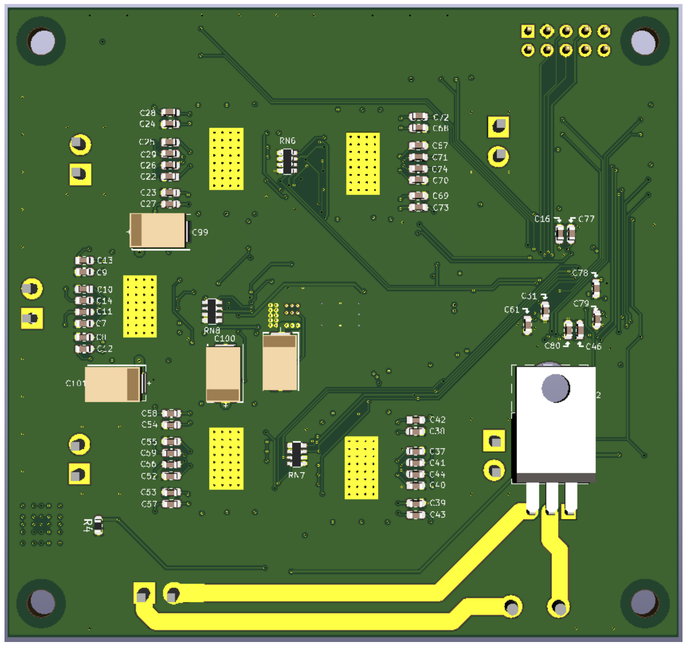


Figure 47 The Copper Layout to help Heat dispersion

Because the copper areas themselves are not be enough to disperse the heat, a heatsink was used connecting it to the exposed pads with thermal pads. The motor outputs of the driver would require, as a recommendation, to have the load as close from the DRV8412 as possible to reduce the, EMI. The DRV8412 also needs to have its settings set before the operation, like programing the overcurrent protector. To program this, this requires the pin of connect a resistor to ground. The resistor is used as an analog overcurrent programmer. To have a fast recovery response time of 250 ns, rather than using the ES2J, the resistor connected to this pin will be a $27K\Omega$ resistor to ground. This programs the driver to have and recovery time of 250ns.

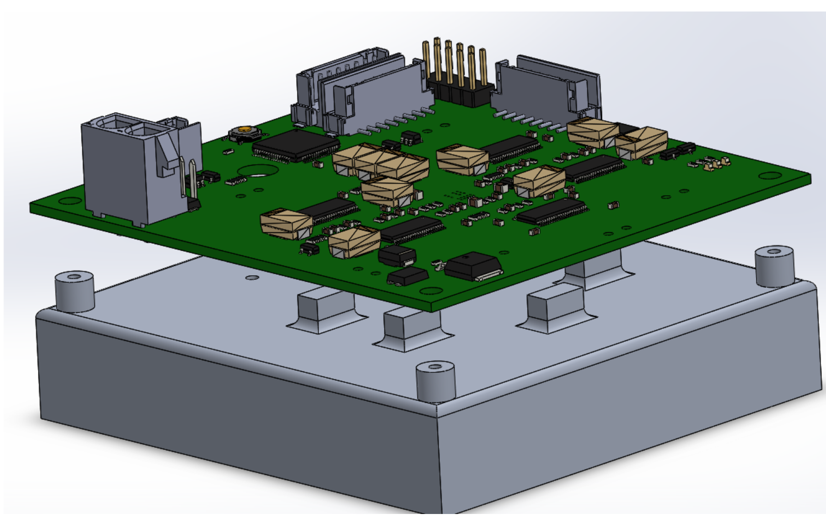


Figure 48 Heatsink Attached to PCB

5.6. Gun Fire Switching

The gun fire control is the circuit which will control the motor of the airsoft gun as well as the firing system. Because of the large current draw, it cannot be controlled directly by our circuit. Therefore, a MOSFET controlled relay was implemented to control the motor, in this case a low side switch was used. Additionally, since the motor only has two states, on and off, it was easy to digitally control via GPIO.

Figure 49, below, shows the gun fire switching circuits that was used. The switching is done through a power MOSFET. The diode is in place to act as a flywheel diode removing the back EMF and preventing it from destroying the rest of the system, since the gun motor is a large inductive load.

The push pull amplifier would be used as a driver to drive a power MOSFET in and out of cutoff. In the configuration shown in, the push-pull driver is configured as a high side driver to the power MOSFET Q_2, controlled by the GPIO. During a logical high, Q_4 is on and Q_2 is then on and the motor is on. At a logical low, the inverse is true, Q_4 is off and Q_2 is off then the motor is off.

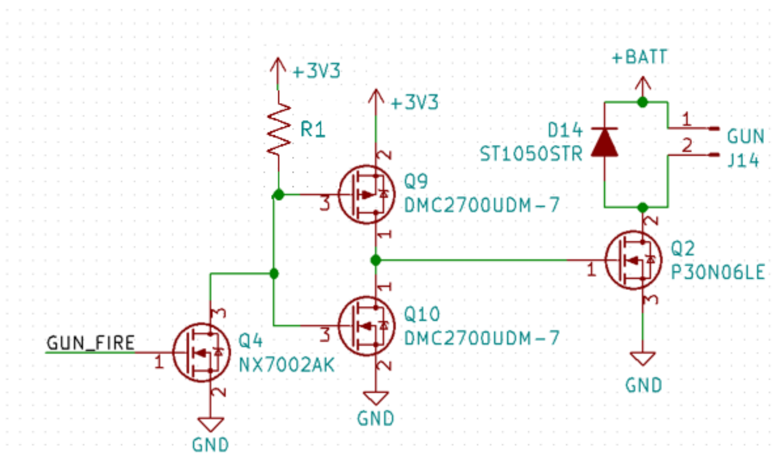


Figure 49 Gun Fire Switching Circuit with Push-Pull Driver

5.7. Turret Control

The turret control is the mechanism for aiming the gun at the target received from the MCU via the CV. They receive move instructions from the MCU control which receives aiming information from the Raspberry Pi. The motors will then orient the turret as necessary to aim at the target.

The turret system is controlled by a geared DC motor with encoder feedback. The motor driver drives the turret using a 1:1 gear system at 80rpm. Using software counting encoder steps, the turret will be limited in the maximum angle it can rotate to. The gun will then be mounted to this rotating axis to provide a pitch angle degree of motion.

Originally, an ADXL345 accelerometer sensor was to be used on the gun to provide the MCU with information on the current angles that the turret is aimed, thus allowing the MCU to calculate the changes required to aim the turret successfully. To do this, the ADXL345 sensor was to be configured as a dual-axis tilt sensor, as described within the technical documentation for the sensor. I2C will be used to communicate with the accelerometer and set the relevant registers for initialization of the accelerometer. Figure 50 shows the connection guidelines for I2C communication provided by the technical documentation of the ADXL345. The same scheme will be directly implemented for the bot. Then, the angles can be measured from three axes. This information will be sent over SPI to the central MCU, which can process the information and compare it with the desired angles for aiming the turret. This constant checking system will be used to adjust the yaw and pitch rotation motors which control the aiming of the turret. Adjustment will be made to the aim of the turret until the desired angles and actual angles are satisfactorily close.

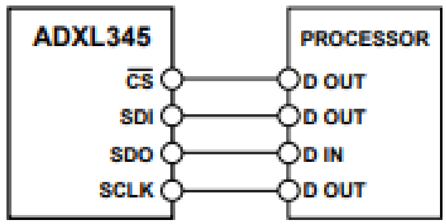


Figure 50 4-Wire SPI Connection Guidelines

Though from testing we found that the use of hall sensor encoders would provide faster feedback and finer control of the angle of the turret.

5.8. Feedback Encoder System

The feedback system of the robot was originally to be recorded by the Qauick incremental encoders. The Qauick encoder is a form of optical encoder that has 400 divisions for each rotation. The following is the circuit used to achieve a two-feedback signal that is sent to the microcontroller. However, the 400 divisions per rotation proved too sensitive, thus we decided to use hall encoders with smaller steps per rotation.

The feedback encoders will measure the velocity of the motor by measuring the phase difference between the two signals. The output signal would be a square wave output that connects to the microcontroller. The input for the microcontroller will be a 3.3V signal and could be read digitally, without the need of being digitally converted from as an analog signal. To test the following system, the circuit will be connected to the motor of the frame and have a motor run, at a given velocity. With this, we can have the controller display the calculated speed in RPM. Then to verify the readings, the use of a professionally fabricated tachometer will used to either confirm the readings of the controller. Should the velocities differ, the gains given in the programming can be tweaked to approach the tachometer readings. This process will need to be applied to each motor individually. Once the motor is individually calibrated, then the examination of having the robot move to abstract points may be taken.

6. Software Design

The software design for this robot encompassed all of the control and logic needed to achieve the goals we set for ourselves along with the requirement specifications set earlier. The master control for TankBot came from the compute module which allows for the use of threads in Linux to achieve concurrent control of various subsystems of the robot simultaneously. The overall system design was first considered, then details concerning each of the various parts of the overall system were considered in far greater detail. The software is divided into two main categories: the main processor software and auxiliary control software. The idea behind this division is to show the base upon which the various interconnected systems sit when explaining the main processor software. Then, the auxiliary control software will explain operation upon the pedestal provided by the main processor software.

6.1. Overall System Design

Figure 2 on page 10 describes the software flow logic that will govern and process the hardware inputs and outputs. The main program make use of multithreading to allow for simultaneous movement and target. Notice that the software control diagram identifies the 4 major components of the system at large: remote command, wheel control, turret control, and CV. The software control diagram allows users to see exactly what major components of the system exist within the system. The remote command allows for wireless communication with the bot to control it. The wheel control allows for planar motion of the bot. The turret control allows for aiming the turret at targets. The CV control allows for acquiring targets.

6.2. Main Processor Software

The main control software runs on the BCM2837 on top of the of the Linux kernel. Since the BCM2837 is such a powerful ARM processor, much of the control software for the TankBot can run on the device in parallel threads. The use of threads allows for processes to be executed asynchronously and the multi-core architecture allows for simultaneous execution.

6.2.1. Threading

The main control software needs to read information from the LiDAR, send commands to the movement controller, and stream images from the camera. While this could be done in a single process, it would be extremely slow and inefficient. A better solution is to segment each task into an asynchronous thread which handles the interfacing with the various subsections of the robot and only reports back pertinent information to the main process. The Linux implementation of threading is called PThread short for POSIX Thread, which allows a program to control multiple different flows of work in the same overlap of time. By moving the LiDAR probe, turret control, fire control, and movement control into separate threads, they can run in parallel and the main can manage the data flow from each thread executing decisions based on received information.

6.2.2. Web Sockets

To communicate between the main program on the BCM2837 and the remote control application in section 6.2.3 web sockets were used along with the UDP protocol described in section 3.3.6.1. UDP was chosen over TCP since the bot will be autonomous. When not connected to the remote control, the bot should not care that the socket is disconnected so a constantly checking TCP server was not an elegant solution. All the control software for the TankBot is written in pure C, while the remote-control interface is written in C++, the benefit of using web sockets for communication between these two programs is that it makes the source language irrelevant and only the data structures matter.

6.2.3. Remote Control Interface

The remote-control program is what will send commands from a host computer to the TankBot. Additionally, the remote command program will show the view from the TankBot's camera to allow for control without directly viewing the bot. During initial development, our team had the choice of two GUI based application development environments, QT and .NET. QT is a GUI framework written in C++ and licensed under GPL while .NET is made by Microsoft with C# and has native GUI frameworks built-in. Table 17 shows a comparison between the features offered by each framework.

	QT	.NET
Language	C++	C#
Native code runtime	Yes	No, runs on .NET VM
VLC support	Yes, native code.	Yes, ActiveX plugin
Native cross platform support	Yes	Partial
Xbox 360 Controller Support	Yes	Yes

Table 17 GUI Framework Comparison

While .NET provides a its rapid development environment, QT provides fast code execution and cross platform support, since the control application will run on Linux, QT was chosen in favor of .NET.

Figure 51 shows the final remote-control interface. When an Xbox 360 controller is connected, the 2-dimensional position of the analog sticks will be used to control the TankBot remotely. The drop-down menu is used to select the different modes of the bot and when set to attack mode, the fire button is enabled allowing for firing of the gun. Reload stream is used to reload the CV stream.

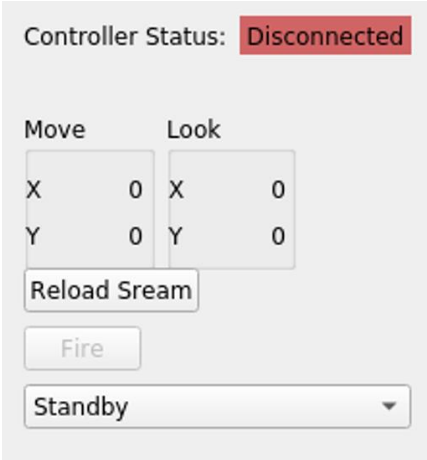


Figure 51 Remote-Control Interface

6.2.4. Wi-Fi Driver

Although the Wi-Fi module connects through a standardized MMC interface, it is impossible to know how to communicate with every single possible type of hardware both past, present, and future. Therefore, drivers are needed which give a standard interface for the kernel to create the abstracted interfaces that higher-level software can expect and use. The driver is a kernel extension that translates the specific and sometimes proprietary controls of the device to the kernel.

For the WILC3000 wireless module, the manufacturer's driver was used as a base and then modified to properly support the application case. The driver initializes the device, waking it from sleep, then registers the device as a standard interface for higher level code to access. Any time a program accesses the WLAN interface, the driver will communicate with the device over the standard SDIO protocol described in section 3.3.3.

6.2.5. Computer Vision

As discussed earlier in this document, we chose to use OpenCV as our computer vision library. OpenCV's libraries include the code needed to execute many of the most important aspects of this project including target detection, target identification, motion detection, and many others.

6.2.5.1. Motion Detection

The Motion detection systems works by observing the differences between any two frames in sequence and ignoring anything that is the same. After taking the difference between the two frames the output will be shown as a screen of white pixels indicating that those pixels are not the same and thus something has moved in that location. By doing this process repeatedly we can track the motion and the shape of whatever is moving.

6.2.5.2. Target Detection

Target detection is a much more complicated process called the Histogram of Oriented Gradients and Object Detection. This process consists of showing the software, which is provided by OpenCV, several positive samples, showing several negative samples, where the number of negative sample images is much greater than the number of positive samples. A set of descriptors was applied to each of the images such that each image is codified: this was important to the next step. Apply a support vector machine to each of the images descriptors, this machine will draw a "line" between what is a positive image and a negative image. After this step, we will apply what is called Hard Negative Mining, which is the process of letting your program attempt to find which *parts* of an image make that image a positive, by correcting the false positives found during this process you teach the classifier how to better differentiate between those negative and positive images. You can stop here, but the more you implement hard negative mining the better your classifier will be and thus your percent of false positives and false negatives will continue to decrease.

6.2.5.3. Target Identification

The process for identifying targets is very similar to the process for detecting a target in the cameras vision, it just has a stricter view on what is a positive hit and what is not such that rather than any human-shaped thing being flagged as positive, only human-shaped things that also are holding a gun-shaped thing are flagged. In addition to this method, you can also have the machine pay attention to how close the human shaped thing is to the robot, or even how fast the human shaped thing is moving in any direction to further classify targets and differentiate friends from foes.

Figure 52 shows an example of how the OpenCV would function on the TankBot. In this example, no filtering is performed, and any humanoid target is identified. Work must be done to not only classify targets but also to increase the detection rate and precision. The example was performed with a test video of pedestrians walking but the final software will use a live video stream instead of a file stream.

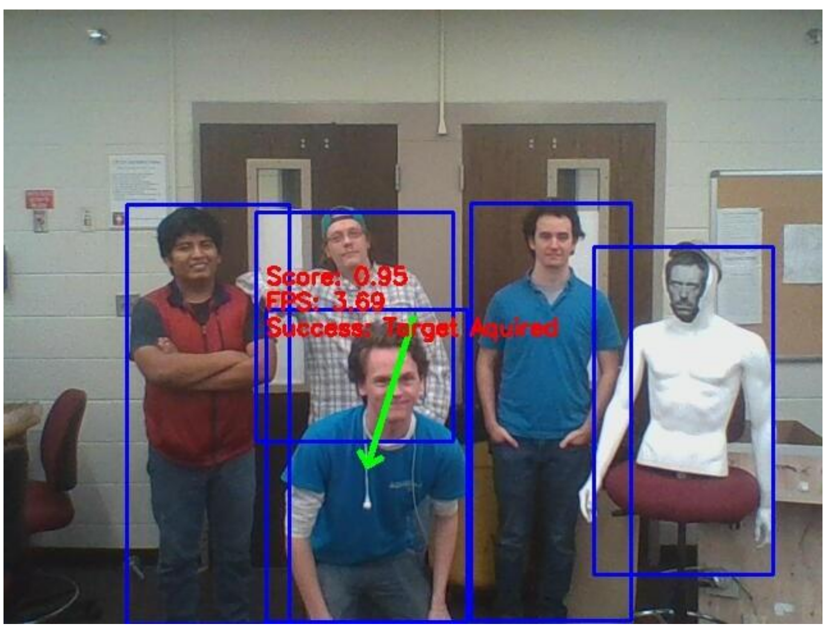


Figure 52 OpenCV People Detector

6.2.6. Simultaneous Localization and Mapping (SLAM)

SLAM, or Simultaneous Localization and Mapping is the acronym of the name for a concept developed by R.C Smith and P. Cheeseman. It applies to the computational problem of mapping a given environment while also tracking the machines own position in that environment relative to other significant landmarks. By making use of the inherent nature of a LiDAR scanner and by compiling several LiDAR scans over each other, we can produce a map of a given area. With the robot able to move, there would need to be some additional processes to reduce the randomness of scans and the inherent drift of the mapped boundaries. The following figure is an example of a small system going awry due to drift caused by the motion of the machine and thus the LiDAR system that is collecting information.

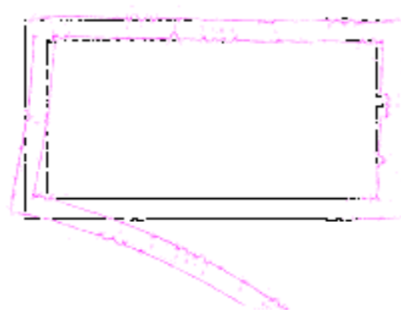


Figure 53 SLAM Mapping

The point cloud map in black is the actual dimensions of the hall, while the purple point cloud map is the sensor output with no correction or refinement to its telemetry. To correct this, the program must be able to know when it should correct and when it should leave it be. One solution is to have the robot make markers of the map so that it's another layer of competence. Also, if the markers contradict each other, then the robot will either shift the map to align and eliminate the contradiction or delete the point since it does not adhere to the rest of its markers. Another solution is to map the same area multiple times and have the points recorded connect to its corresponding counterpart. This can be like taking the average of two maps to get one. The resulting map will the merging of the two or more maps.

6.3. Auxiliary Control Systems

Auxiliary control systems consist of many critical logical systems of TankBot including the entire turret control system and all the logic associated with firing the gun in addition to the general locomotive systems. As such, the next subsections will go into a deep discussion of these systems, how they are developed, and how they are going to be implemented.

6.3.1. Gun Firing Logic

The gun firing logic is quite simple. After the CV has acquired a target and centered on the target by sending control vectors to the turret, the MCU needs only to drive the port that the fire trigger is connected to fire the gun. When the firing is to cease, the MCU needs only to clear the port. Further details of why this scheme for the MCU is so simple can be found in section 5.6. The logic itself was written entirely in C code.

6.3.2. Turret Logic

After a target has been acquired through the CV, the MCU will need to take note of the relevant angle that the turret presently occupies and determine what direction to move based on the received movement vector. The MCU then moves the turret while counting the steps to adjust the turret as needed.

6.3.3. Locomotive Logic

The locomotive control of the robot will operate in 4 primary modes:

- Standby
- Patrol
- Follow Target
- Go to Location

In the standby mode, the robot will simply halt in place. The locomotive motors will not be active, so the robot will not drive around. As such, the robot will simply be holding its ground in the area. The robot will also not shoot targets while in this mode. Essentially, it's a sleep mode for the robot.

In the patrol mode, the robot will trace a perimeter using the CV and LiDAR to guide it about. The MCU will control the patrolling actions to activate and deactivate the locomotive motors as necessary to allow the robot to seem as if it is patrolling an area. If a target is acquired by the CV, after rating distance with the LiDAR, the robot will aim the turret and shoot the target.

In the follow target mode, the robot will have acquired a target and will follow it without shooting unless the user specifies otherwise, or the target imposes an approaching threat to the robot. The MCU will control the following using the CV and LiDAR as a guide for deciding on how to drive the locomotive motors and whether to active the firing system to shoot the target.

In the go to location mode, the robot will simply travel to the target location. The locomotive motors will be activated or deactivated, as necessary, by the MCU to allow for routing to the location. The robot will be guided by CV and LiDAR. As the robot approaches the location or arrives at the location, if an enemy is determined by the CV, the robot will fire if the enemy is within the robot's effective range.

These modes will be specified by the user through the remote command logic software subsystem across the wireless link block displayed in Figure 1. As such, the user will have direct control over the actions of the robot and whether it is acting autonomously or not.

The method to accomplish this is to receive bits, two bytes of data, from the master controller. These data hold the coordinates for the next position in relation to the robot. With this the robot will align to a straight path to it and follow the straight path. Should the path be obstructed by any immediate object, the reactive sensor will push the robot to another direction. As a result, the displacement would need to be considered. To reduce the amount of error, the robot will need to be able to decide the path with the least cost to its effort. Therefore, as an overlay to the robot moving, a vector field with all the vectors pointing to the desired location can be implemented to move the robot to its desired location. The following is such implementation of a vector field.

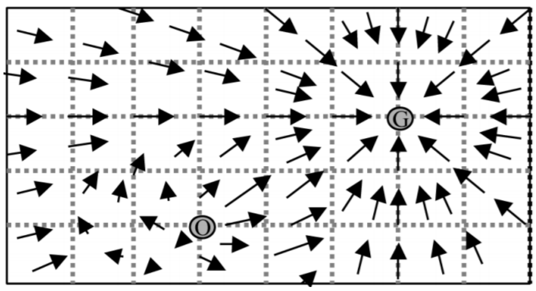


Figure 54 Vector Field of Robot(O) guided to the Goal(G)

One thing to not do is use absolute directional vectors and add noise to these fields. By not having noise inserted, should the robot enter a three-walled corner, then the robot may be perpetually get stuck trying to recalculate the directional vectors. By adding noise, this adds randomness to the vector field and can help the robot eventually leave these chokes. The tools required of this would be the LiDAR scanner with odometry feedback coming from the encoders.

6.3.4. LiDAR Logic

The LiDAR module is a device that operates on the Time of Flight principal. The module measures the distance and reports the information to the host via UART with a 115200-baud rate. The structure of a single packet of data is shown in Table 18. The received packet is compared against the checksum for data integrity, the checksum is the lower byte of the sum of bytes 1-8.

	Byte 1-2	Byte3-4	Byte5-6	Byte 7	Byte 8	Byte 9
Type	uint16_t	double	double	Reserved	uint8_t	uint8_t
Value	0x5959	Distance	Strength	0	Raw Quality	Checksum

Table 18 Lidar UART Packet

There exists a documented list of commands to interface with the module. These commands give access for the main program to control the data flow, output format, detection pattern, range limit, and many others. The full list is in Table 19.

Config Item	Command	Description	Default
Output data format	42 57 02 00 00 00 01 06	Standard format, as shown in Table 18	Yes
	42 57 02 00 00 00 04 06	“Pixhawk” data format	No
Data output period	42 57 02 00 EE FF 00 07	EE FF: setting of output period (ms) it must be the integral multiple of 10ms	10ms/ 100Hz
Unit of distance	42 57 02 00 00 00 00 1A	Output unit of distance data is cm	Yes
	42 57 02 00 00 00 01 1A	Automatic detection pattern	No
Distance mode	42 57 02 00 00 00 02 11	Short distance mode, applicable for 0-5m	No
	42 57 02 00 00 00 07 11	Long distance mode, applicable for 1-12m	No
Setting of range limit	42 57 02 00 00 00 00 19	Range limit disabled	No
	42 57 02 00 EE FF 01 19	EE FF: threshold of ranging limit (mm)	Yes, max 12m
lower limit of signal strength threshold	42 57 02 00 EE 00 00 20	EE: setting of the lower limit of signal strength threshold. If true signal <set threshold output = “FFFF” as the distance value which means invalid.	Yes, 20 DEC
upper limit of signal strength threshold	42 57 02 00 EE FF GG 21	EE FF: setting of the upper limit of signal strength threshold GG: output distance value (cm)	No
Setting of baud rate	42 57 02 00 00 00 GG 08	The baud rate corresponding to GG is shown in Table 20.	115200
Setting of trigger sources	42 57 02 00 00 00 01 40	Internal trigger, 100Hz as default	Yes
	42 57 02 00 00 00 00 40	External trigger	No
External trigger setting	42 57 02 00 00 00 00 41	command for one single measurement	No
Reset	42 57 02 00 FF FF FF FF	All settings are reset to the default	No

Table 19 Lidar Module Commands

The list of baud rates available varies from the standard 9600 to 512000. The full list of baud rate command codes is shown in Table 20.

GG	0x00	0x01	0x02	0x03	0x04	0x05	0x06
Baud Rate	9600	14400	19200	38400	56000	57600	115200
GG	0x07	0x08	0x09	0x0a	0x0b	0x0c	
Baud Rate	12800	230400	25600	460800	500000	512000	

Table 20 Baud Rate Configuration Codes

7. Project Prototype Construction

This section highlights the details behind the construction of the prototype bot as well as the various testing phases it undergoes to achieve a high-performance, well-design bot coherent to the requirement specifications, constraints, and standards. Without a working prototype, a final design isn't accomplishable, so it's important to carefully criticize the prototype until it is in a well-operating condition that is acceptable to the stakeholders and engineers.

7.1. PCB Schematics

The main control board that will be 4-layers and house the Compute Module as well as the ATWILC3000 wireless module, I/O ports, and power regulation for all logic components. Thus, the main schematic is divided into logical pages with each page accomplishing a specific task such as power regulation/distribution or I/O. This top-level diagram is shown in Figure 55. The following pages contain the schematics for each individual page.

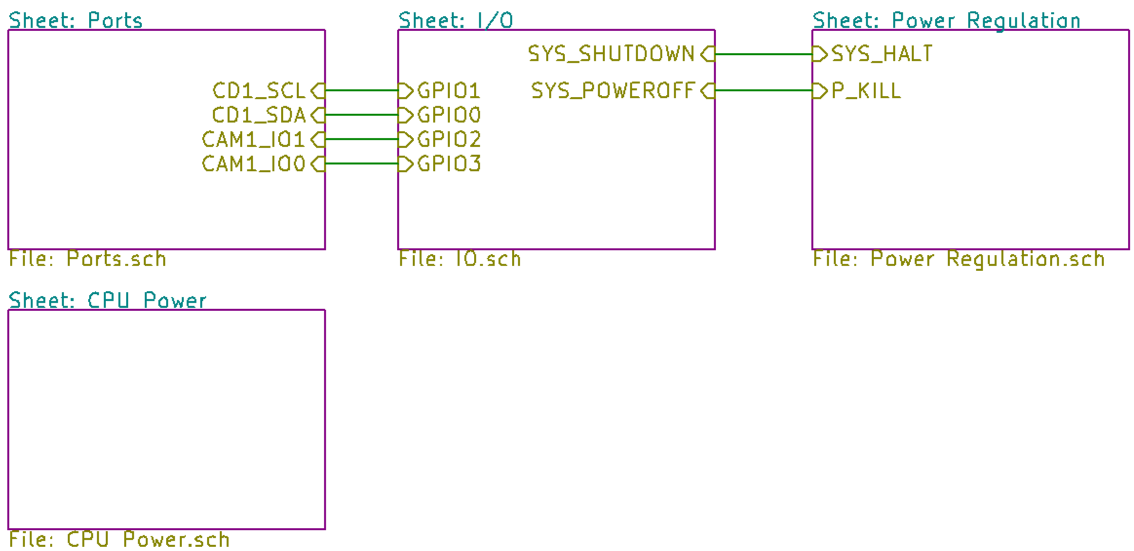
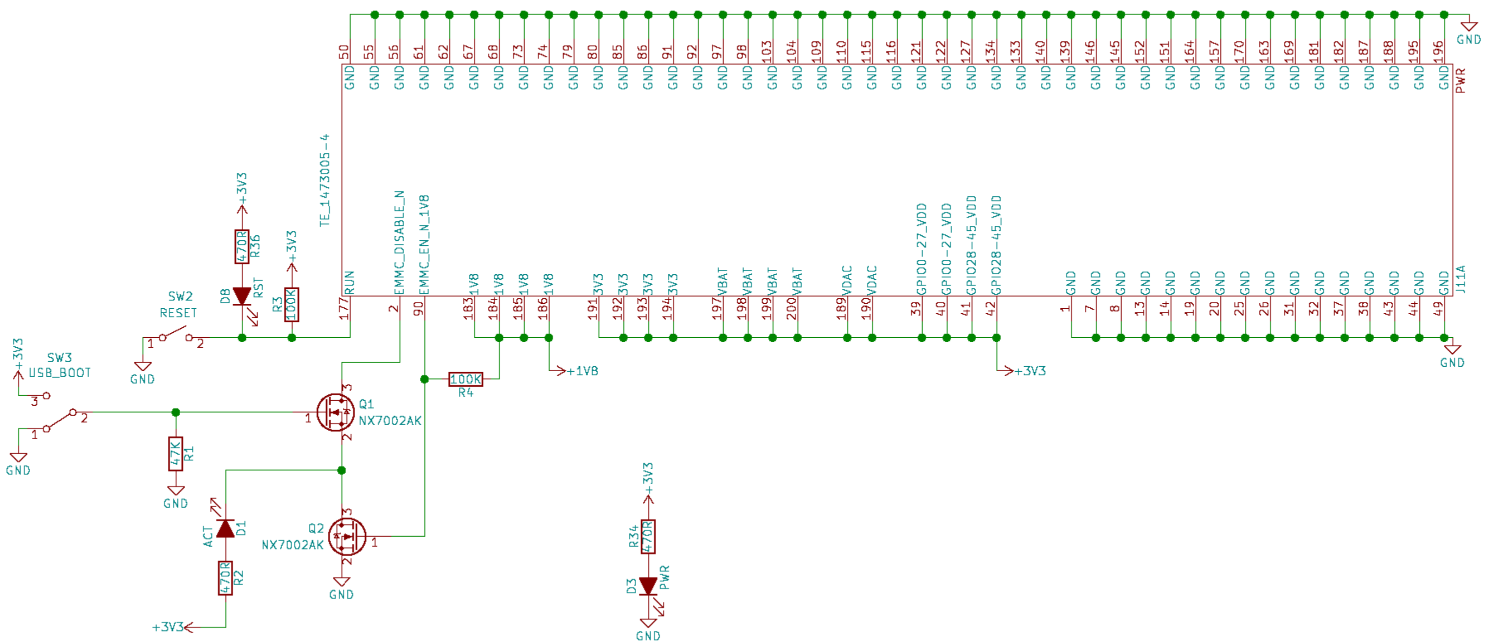


Figure 55 Overall Schematic Top-Level Diagram

Meanwhile the motor control board due to the high noise and current environment will be housed on a separate 4-layer board. The last page of schematics shows the schematic for this board.

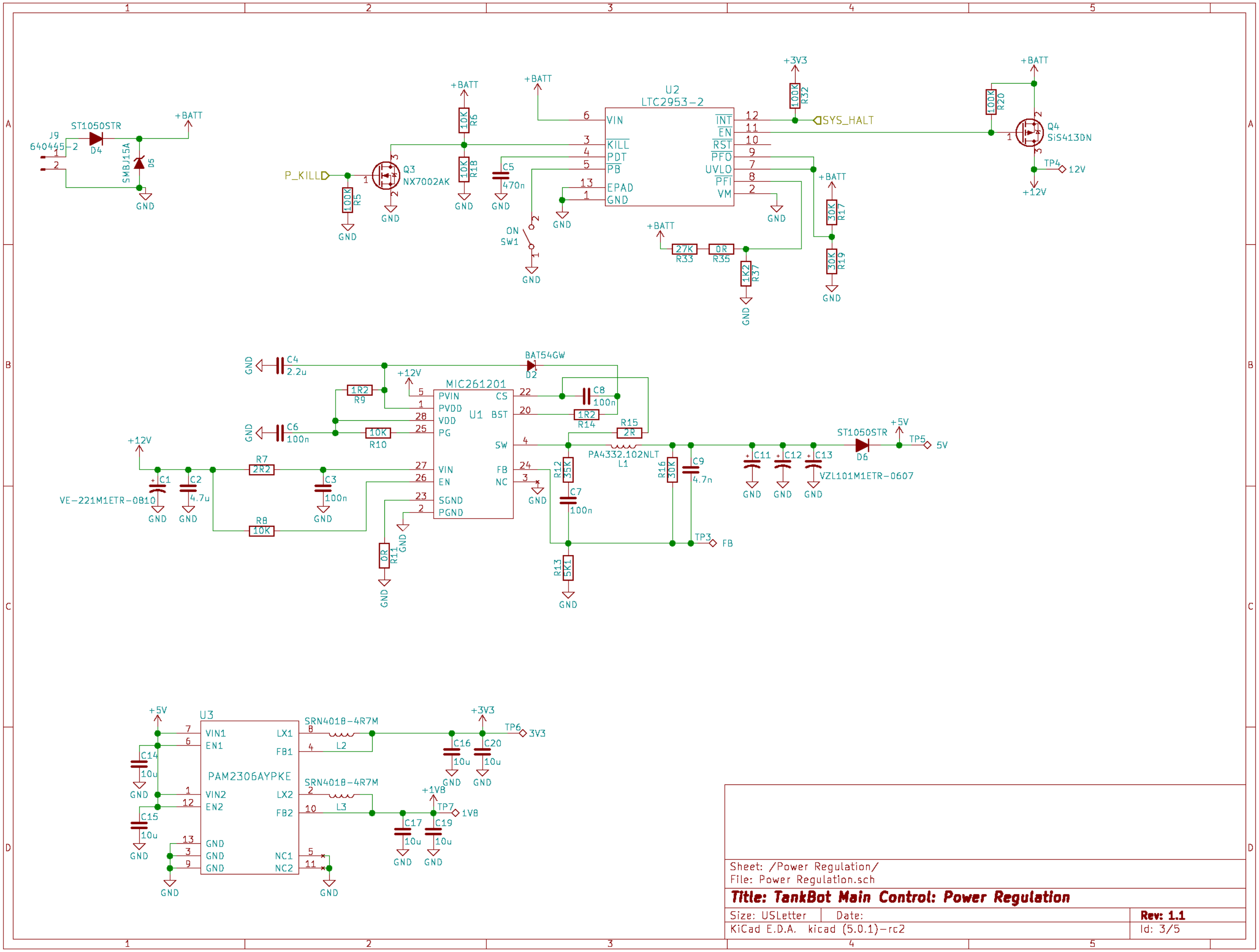


Sheet: /CPU Power/
File: CPU Power.sch

Title: TankBot Main Control: CPU Power

Size: USLetter | Date: | **Rev: 1.1**

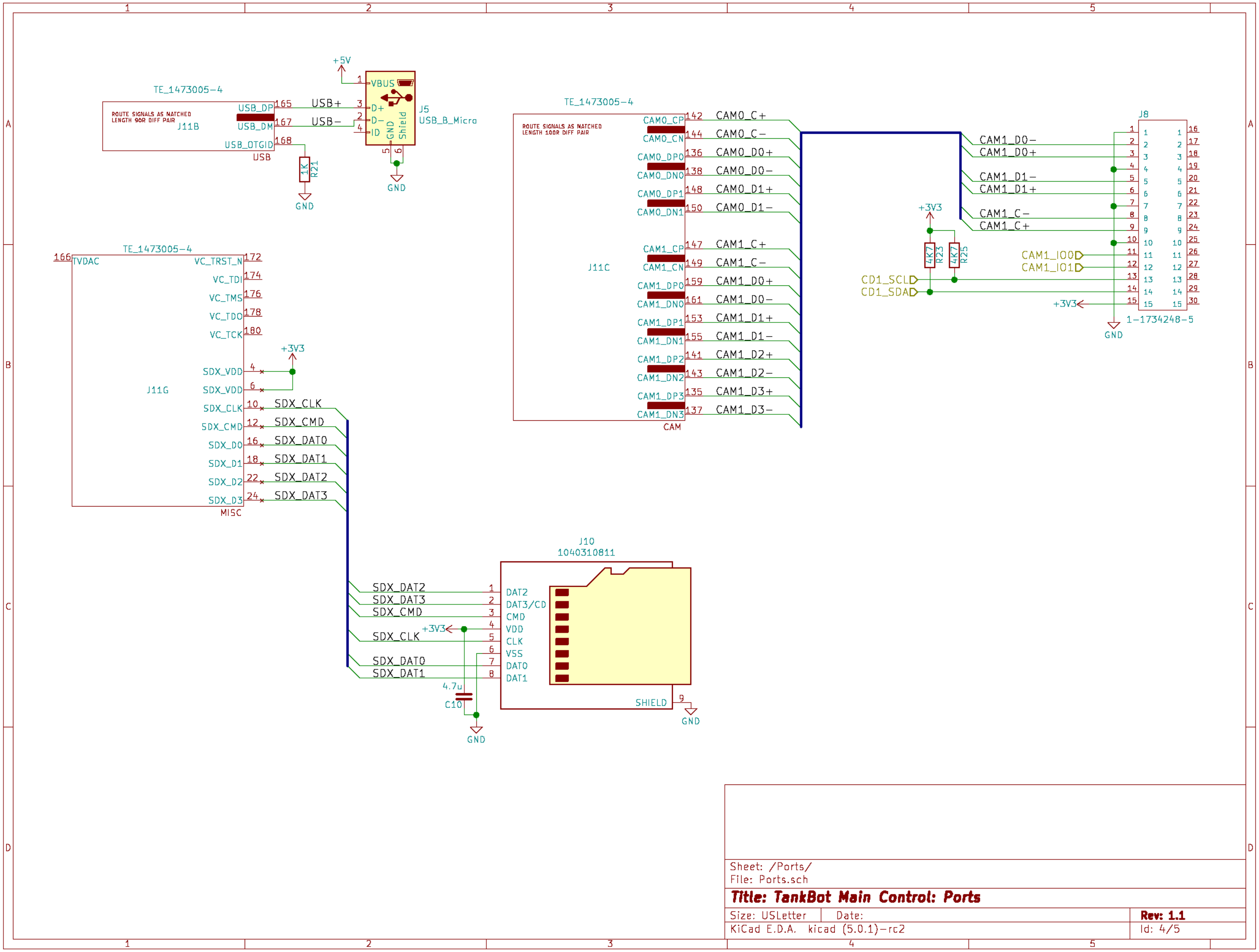
KiCad E.D.A. kicad (5.0.1)-rc2 | Id: 2/5

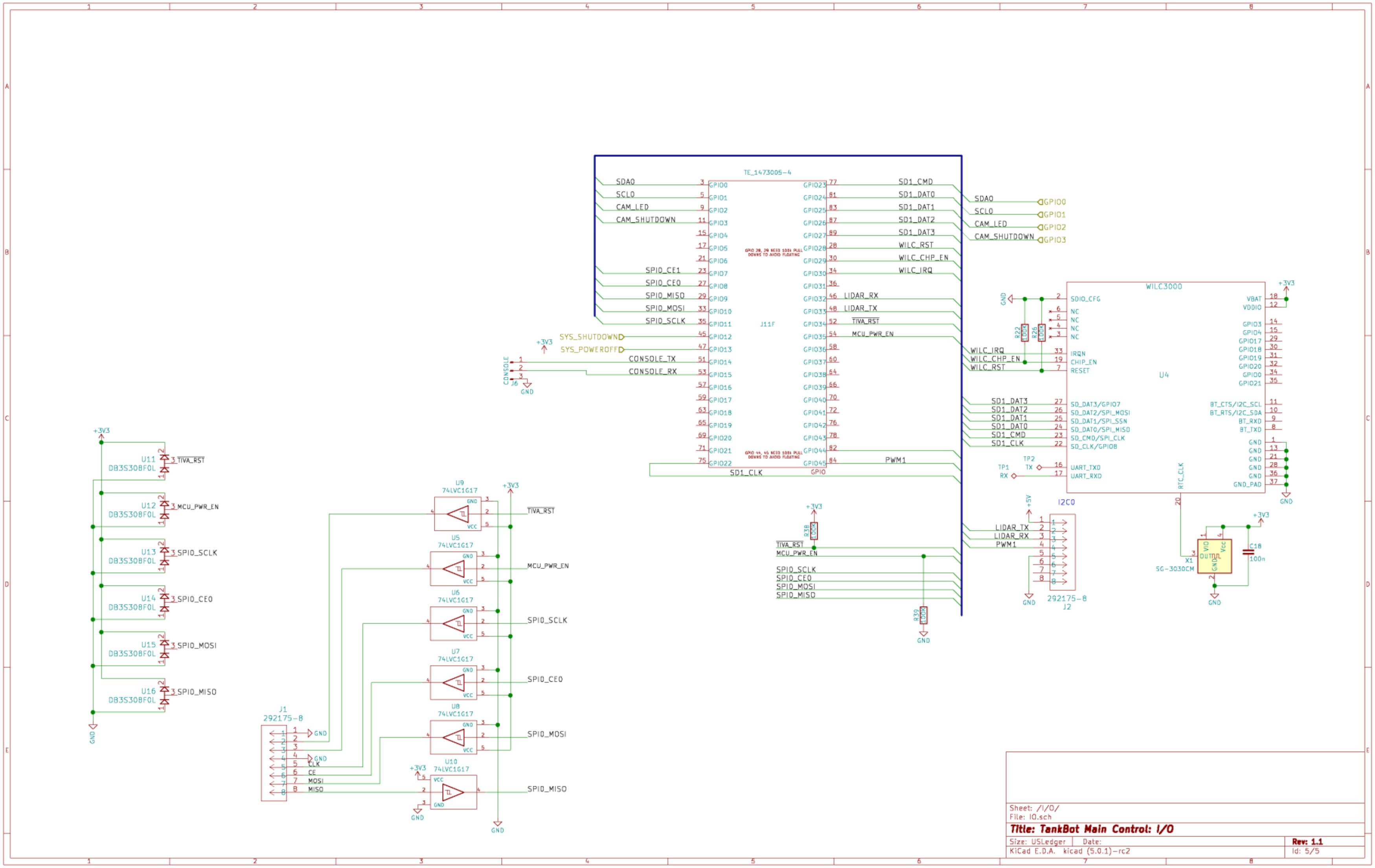


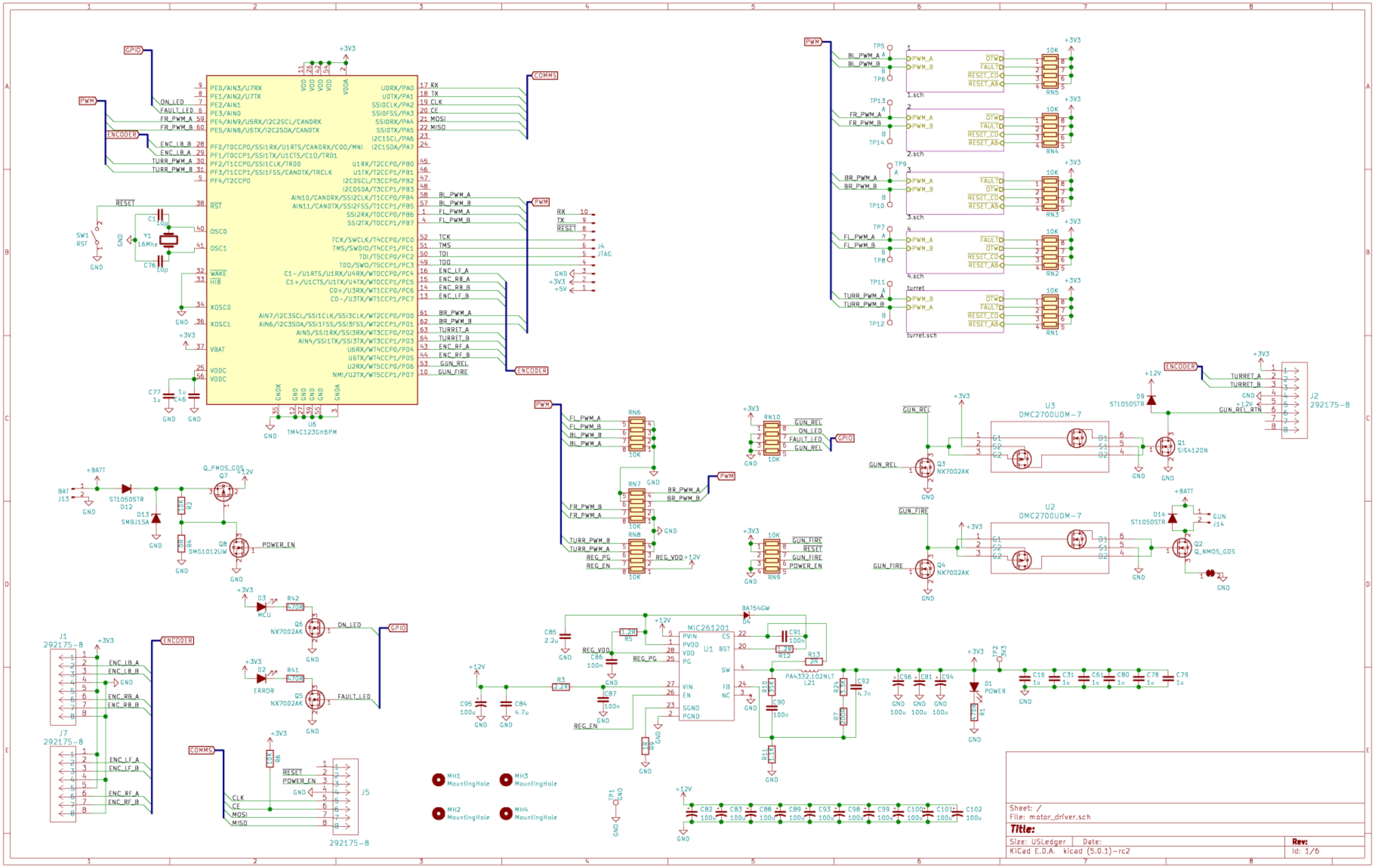
Sheet: /Power Regulation/
File: Power Regulation.sch

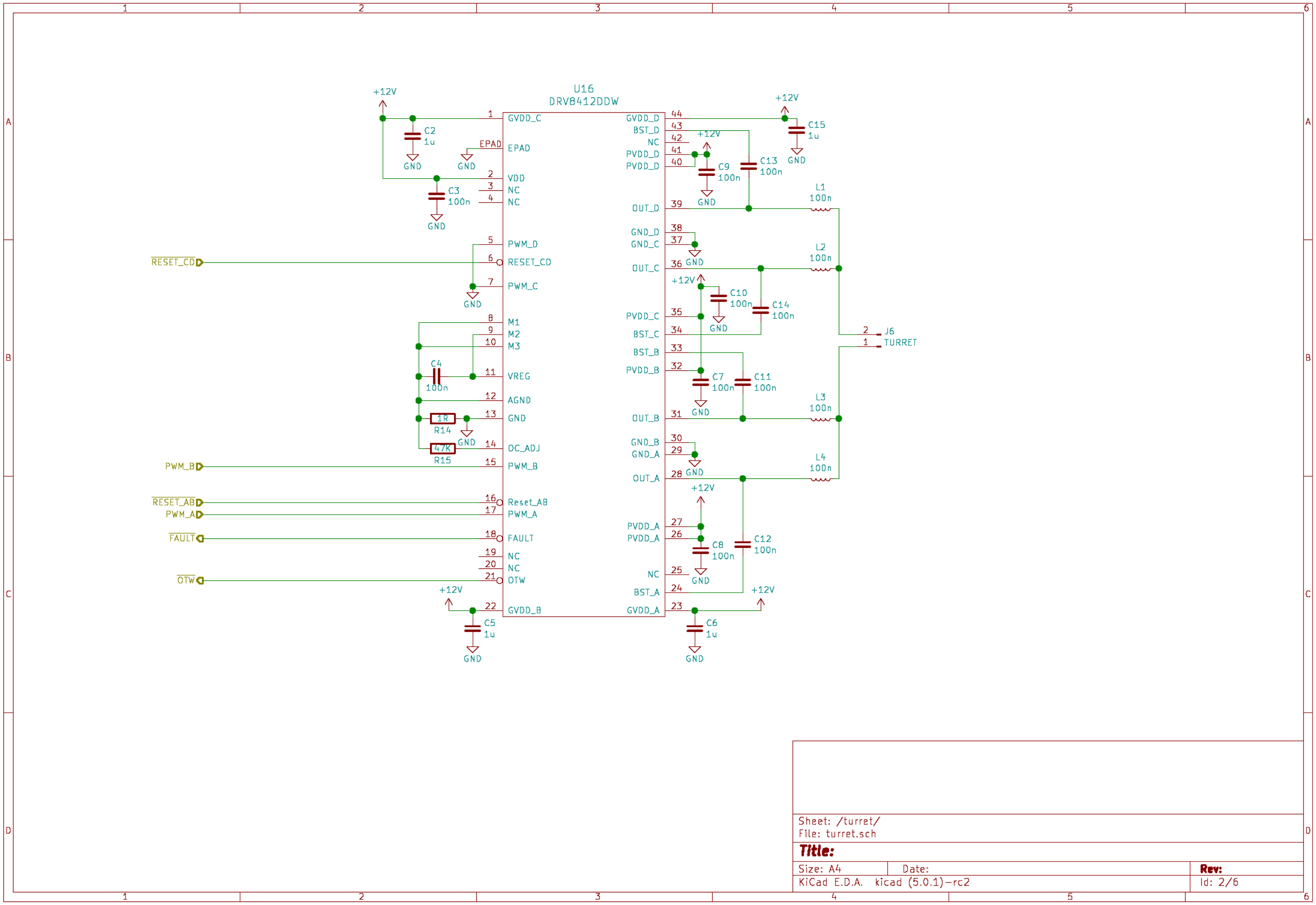
Title: TankBot Main Control: Power Regulation

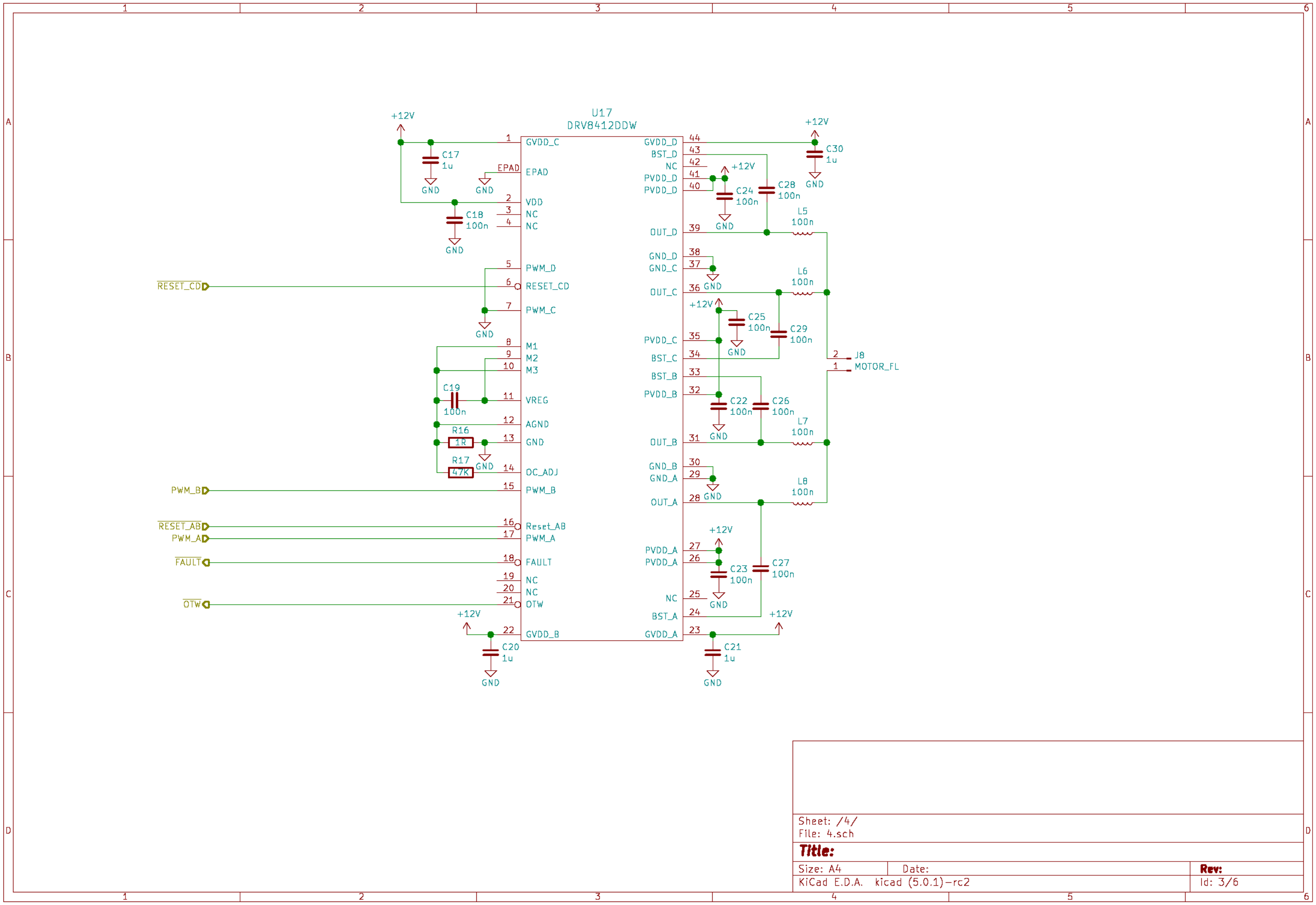
Size: USLetter | Date: | **Rev: 1.1**
KiCad E.D.A. kicad (5.0.1)-rc2 | Id: 3/5

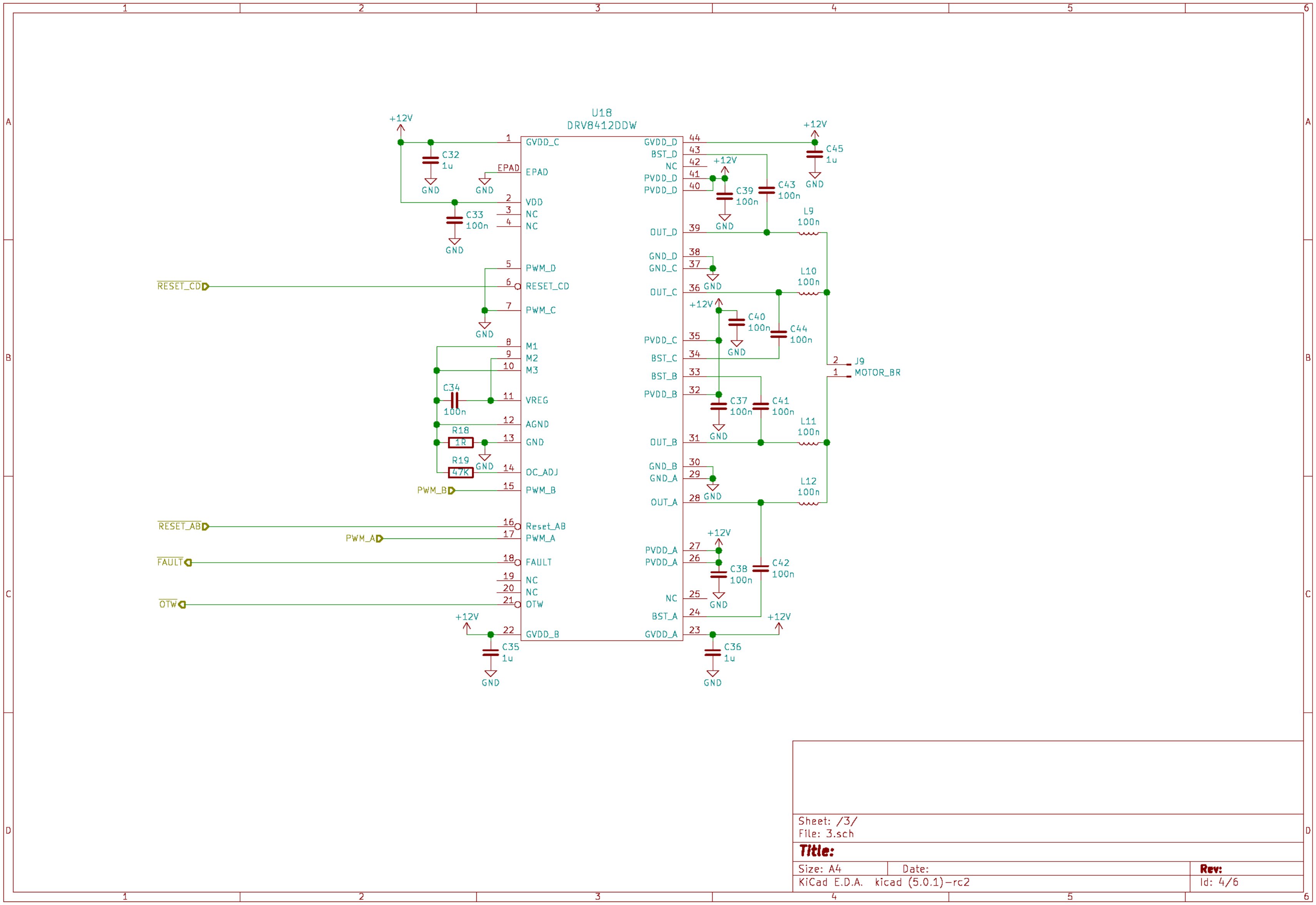

TE_1473005-4
ROUTE SIGNALS AS MATCHED LENGTH 90R DIFF PAIR
J11B
USB_DP 165 USB+
USB_DM 167 USB−
USB_OTGID 168
USB
1K R21
GND
+5V
VBUS
D+
D−
ID
GND
Shield
J5
USB_B_Micro
TE_1473005-4
ROUTE SIGNALS AS MATCHED LENGTH 100R DIFF PAIR
J11C
CAM0_CP 142 CAM0_C+
CAM0_CN 144 CAM0_C−
CAM0_DP0 136 CAM0_D0+
CAM0_DN0 138 CAM0_D0−
CAM0_DP1 148 CAM0_D1+
CAM0_DN1 150 CAM0_D1−
CAM1_CP 147 CAM1_C+
CAM1_CN 149 CAM1_C−
CAM1_DP0 159 CAM1_D0+
CAM1_DN0 161 CAM1_D0−
CAM1_DP1 153 CAM1_D1+
CAM1_DN1 155 CAM1_D1−
CAM1_DP2 141 CAM1_D2+
CAM1_DN2 143 CAM1_D2−
CAM1_DP3 135 CAM1_D3+
CAM1_DN3 137 CAM1_D3−
CAM
J8
CAM1_D0−
CAM1_D0+
CAM1_D1−
CAM1_D1+
CAM1_C−
CAM1_C+
+3V3
4K7 R23
4K7 R25
CAM1_IO0
CAM1_IO1
CD1_SCL
CD1_SDA
+3V3
GND
1-1734248-5
TE_1473005-4
166 TVDAC
VC_TRST_N 172
VC_TDI 174
VC_TMS 176
VC_TDO 178
VC_TCK 180
J11G
+3V3
SDX_VDD 4
SDX_VDD 6
SDX_CLK 10 SDX_CLK
SDX_CMD 12 SDX_CMD
SDX_D0 16 SDX_DAT0
SDX_D1 18 SDX_DAT1
SDX_D2 22 SDX_DAT2
SDX_D3 24 SDX_DAT3
MISC
J10
1040310811
SDX_DAT2 1 DAT2
SDX_DAT3 2 DAT3/CD
SDX_CMD 3 CMD
+3V3 4 VDD
SDX_CLK 5 CLK
6 VSS
SDX_DAT0 7 DAT0
SDX_DAT1 8 DAT1
SHIELD 9
4.7u
C10
GND
GND
Sheet: /Ports/
File: Ports.sch
Title: TankBot Main Control: Ports
Size: USLetter
Date:
Rev: 1.1
KiCad E.D.A. kicad (5.0.1)-rc2
Id: 4/5

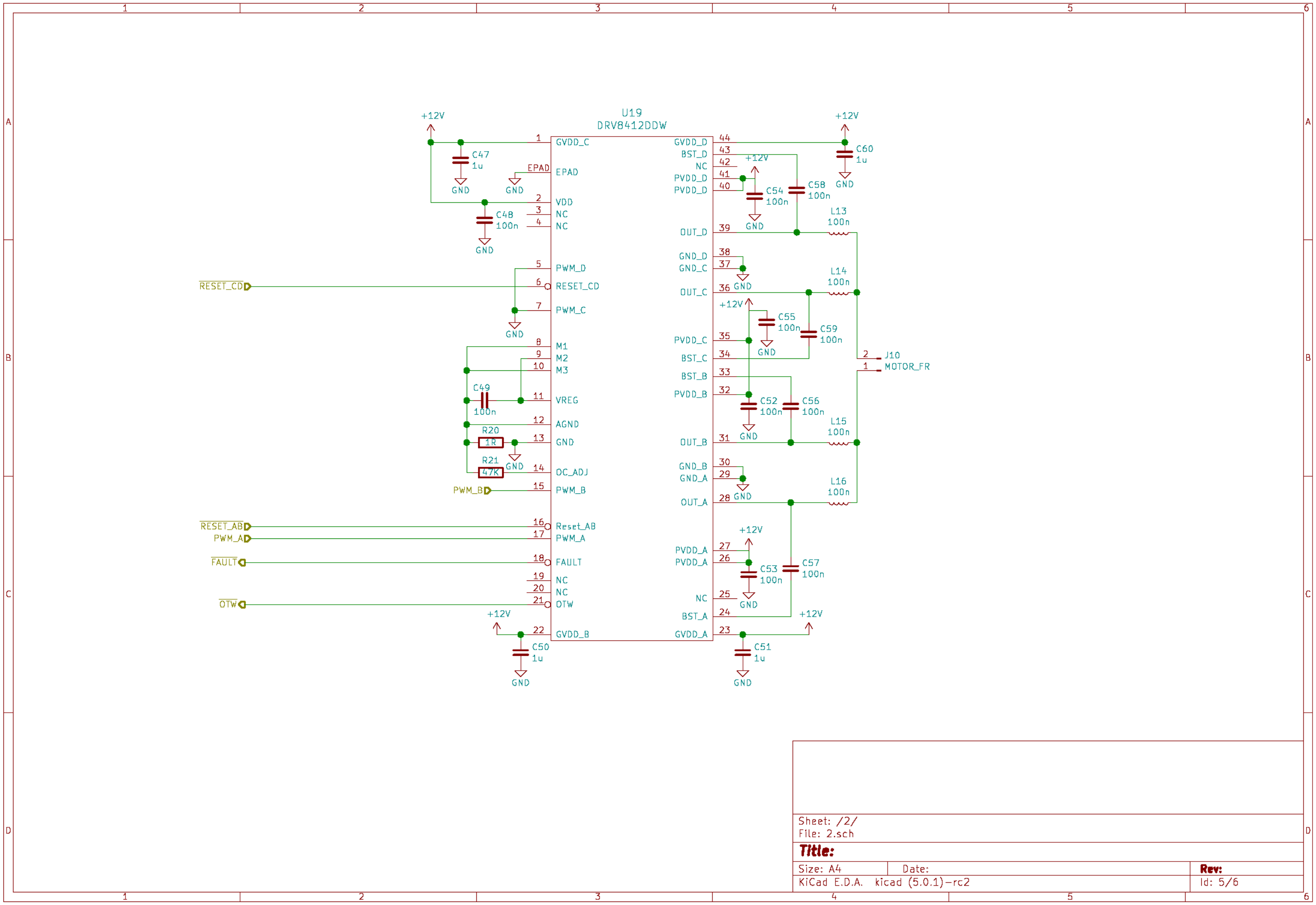

TE_1473005-4
J11F
WILC3000
U4
SDA0
SCL0
CAM_LED
CAM_SHUTDOWN
SPI0_CE1
SPI0_CE0
SPI0_MISO
SPI0_MOSI
SPI0_SCLK
SYS_SHUTDOWN
SYS_POWEROFF
CONSOLE_TX
CONSOLE_RX
SD1_CMD
SD1_DAT0
SD1_DAT1
SD1_DAT2
SD1_DAT3
SD1_CLK
WILC_RST
WILC_CHP_EN
WILC_IRQ
LIDAR_RX
LIDAR_TX
TIVA_RST
MCU_PWR_EN
PWM1
I2C0
292175-8
J1
J2
U5
U6
U7
U8
U9
U10
74LVC1G17
U11
U12
U13
U14
U15
U16
DB3S308F0L
SG-3030CM
X1
C18
100n
+3V3
GND
Sheet: /I/O/
File: IO.sch
Title: TankBot Main Control: I/O
Size: USLedger
Date:
Rev: 1.1
KiCad E.D.A. kicad (5.0.1)-rc2
Id: 5/5

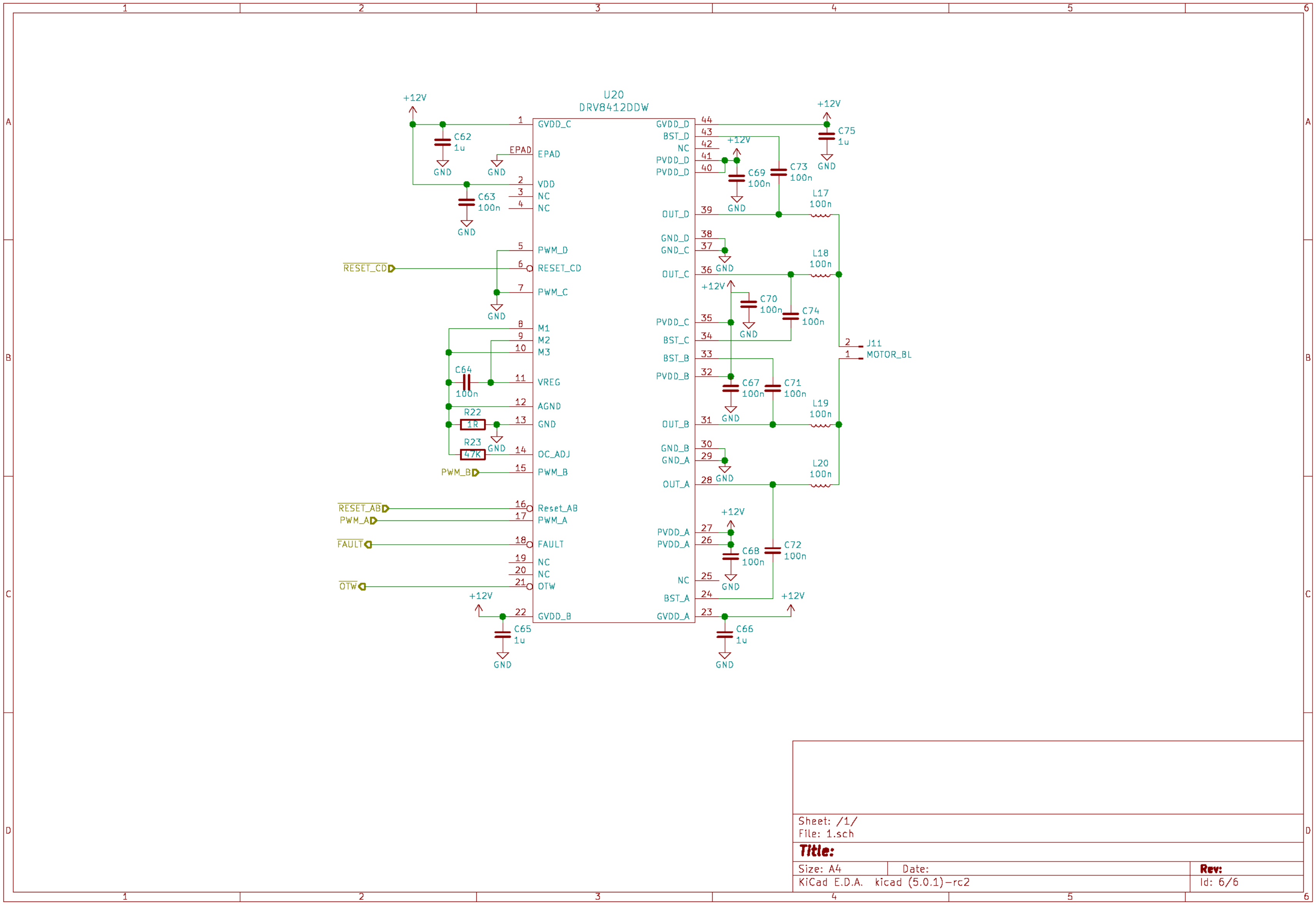

TM4C123GH6PM
DMC2700UDM-7
MIC261201
Sheet: /
File: motor_driver.sch
Title:
Size: USLedger
Date:
KiCad E.D.A. kicad (5.0.1)-rc2
Rev:
Id: 1/6


U16
DRV8412DDW
+12V
GVDD_C
EPAD
VDD
NC
PWM_D
RESET_CD
PWM_C
M1
M2
M3
VREG
AGND
GND
OC_ADJ
PWM_B
Reset_AB
PWM_A
FAULT
OTW
GVDD_B
GVDD_D
BST_D
PVDD_D
OUT_D
GND_D
GND_C
OUT_C
PVDD_C
BST_C
BST_B
PVDD_B
OUT_B
GND_B
GND_A
OUT_A
PVDD_A
BST_A
GVDD_A
C2 1u
C3 100n
C4 100n
C5 1u
C6 1u
C7 100n
C8 100n
C9 100n
C10 100n
C11 100n
C12 100n
C13 100n
C14 100n
C15 1u
R14 1R
R15 47K
L1 100n
L2 100n
L3 100n
L4 100n
J6 TURRET
GND
RESET_CD
PWM_B
RESET_AB
PWM_A
FAULT
OTW
Sheet: /turret/
File: turret.sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. kicad (5.0.1)-rc2
Id: 2/6


U17
DRV8412DDW
GVDD_C
EPAD
VDD
NC
PWM_D
RESET_CD
PWM_C
M1
M2
M3
VREG
AGND
GND
OC_ADJ
PWM_B
Reset_AB
PWM_A
FAULT
OTW
GVDD_B
GVDD_D
BST_D
PVDD_D
OUT_D
GND_D
GND_C
OUT_C
PVDD_C
BST_C
BST_B
PVDD_B
OUT_B
GND_B
GND_A
OUT_A
PVDD_A
BST_A
GVDD_A
+12V
GND
C17 1u
C18 100n
C19 100n
C20 1u
C21 1u
C22 100n
C23 100n
C24 100n
C25 100n
C26 100n
C27 100n
C28 100n
C29 100n
C30 1u
R16 1R
R17 47K
L5 100n
L6 100n
L7 100n
L8 100n
J8
MOTOR_FL
RESET_CD
PWM_B
RESET_AB
PWM_A
FAULT
OTW
Sheet: /4/
File: 4.sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. kicad (5.0.1)-rc2
Id: 3/6


U18
DRV8412DDW
GVDD_C
EPAD
VDD
NC
PWM_D
RESET_CD
PWM_C
M1
M2
M3
VREG
AGND
GND
OC_ADJ
PWM_B
Reset_AB
PWM_A
FAULT
OTW
GVDD_B
GVDD_D
BST_D
PVDD_D
OUT_D
GND_D
GND_C
OUT_C
PVDD_C
BST_C
BST_B
PVDD_B
OUT_B
GND_B
GND_A
OUT_A
PVDD_A
BST_A
GVDD_A
J9
MOTOR_BR
RESET_CD
RESET_AB
PWM_B
PWM_A
FAULT
OTW
+12V
GND
C32 1u
C33 100n
C34 100n
C35 1u
C36 1u
C37 100n
C38 100n
C39 100n
C40 100n
C41 100n
C42 100n
C43 100n
C44 100n
C45 1u
L9 100n
L10 100n
L11 100n
L12 100n
R18 1R
R19 47K
Sheet: /3/
File: 3.sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. kicad (5.0.1)-rc2
Id: 4/6


U19
DRV8412DDW
GVDD_C
EPAD
VDD
NC
PWM_D
RESET_CD
PWM_C
M1
M2
M3
VREG
AGND
GND
OC_ADJ
PWM_B
Reset_AB
PWM_A
FAULT
OTW
GVDD_B
GVDD_D
BST_D
PVDD_D
OUT_D
GND_D
GND_C
OUT_C
PVDD_C
BST_C
BST_B
PVDD_B
OUT_B
GND_B
GND_A
OUT_A
PVDD_A
BST_A
GVDD_A
C47 1u
C48 100n
C49 100n
R20 1R
R21 47K
C50 1u
C51 1u
C60 1u
C54 100n
C58 100n
C55 100n
C59 100n
C52 100n
C56 100n
C53 100n
C57 100n
L13 100n
L14 100n
L15 100n
L16 100n
J10
MOTOR_FR
+12V
GND
RESET_CD
PWM_B
RESET_AB
PWM_A
FAULT
OTW
Sheet: /2/
File: 2.sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. kicad (5.0.1)-rc2
Id: 5/6



U20
DRV8412DDW
GVDD_C
EPAD
VDD
NC
NC
PWM_D
RESET_CD
PWM_C
M1
M2
M3
VREG
AGND
GND
OC_ADJ
PWM_B
Reset_AB
PWM_A
FAULT
NC
NC
OTW
GVDD_B
GVDD_D
BST_D
NC
PVDD_D
PVDD_D
OUT_D
GND_D
GND_C
OUT_C
PVDD_C
BST_C
BST_B
PVDD_B
OUT_B
GND_B
GND_A
OUT_A
PVDD_A
PVDD_A
NC
BST_A
GVDD_A
+12V
GND
C62 1u
C63 100n
C64 100n
C65 1u
C66 1u
C67 100n
C68 100n
C69 100n
C70 100n
C71 100n
C72 100n
C73 100n
C74 100n
C75 1u
R22 1R
R23 47K
L17 100n
L18 100n
L19 100n
L20 100n
J11
MOTOR_BL
RESET_CD
RESET_AB
PWM_B
PWM_A
FAULT
OTW
Sheet: /1/
File: 1.sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. kicad (5.0.1)-rc2
Id: 6/6

7.2. PCB Layout

Figure 56 Shows the preliminary PCB layout with fills not shown for clarity. The largest component is the DDR2 socket which houses the Compute Module. From the socket, at the top right corner of the board is the ATWILC3000 wireless module. The traces connecting the two devices where designed such that the length of each trace is the same length $\pm 0.5mm$, this is key in high speed communication for correct timings. To the left bottom and lower center of the PCB contain the external connections for modules such as the CSI camera and USB. To the lower right, UART for the LiDAR as well as a USB mode select jumper are located. The MIPI differential pairs for the CSI where designed as 100Ω differential pairs while the USB lines where designed as a 90Ω differential pair. For both cases, the lines were routed as differential CPW lines as described in section 5.3.

The bottom left and right of the board houses all the power management circuits including the stepdown regulators and voltage supervisor IC. Throughout the board, there are vias scattered in a grid pattern, these are grounding vias used to stitch the ground planes between the various layers and keep ground at the same potential everywhere, the traces connecting the grounding vias in a grid are just to set the net to GND, these are absorbed by the ground fill. The 3D rendering of the PCB is shown in Figure 57. Models of components were retrieved from the respective manufacturers website when available.

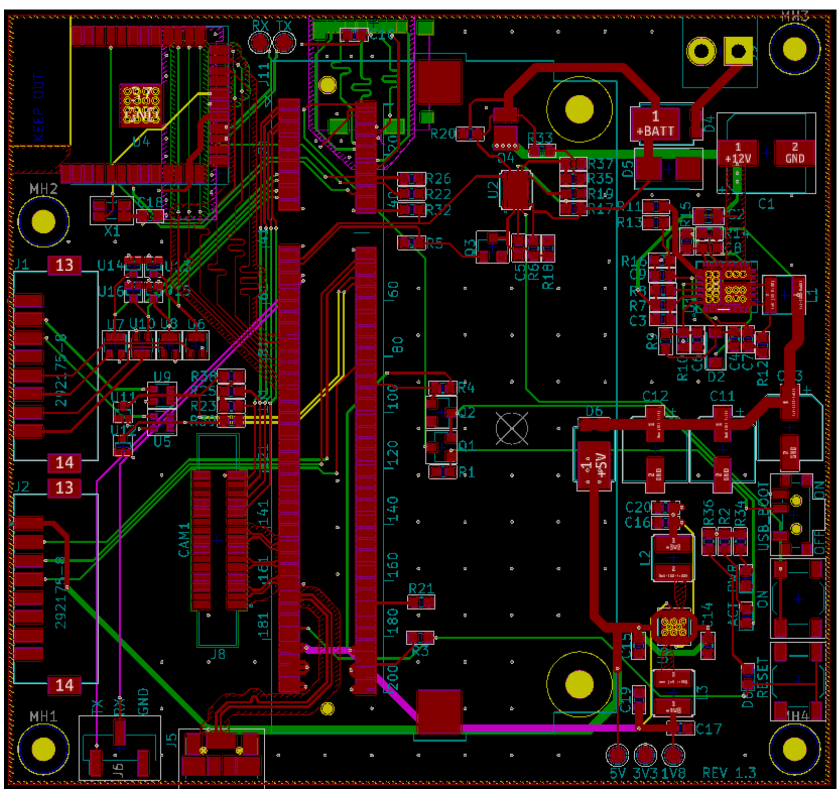


Figure 56 Raspberry Pi PCB Layout

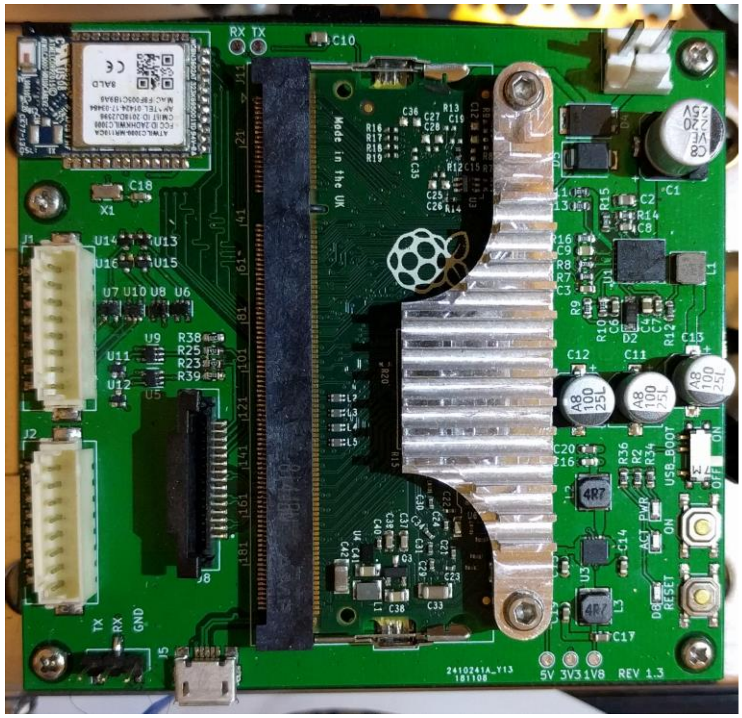


Figure 57 Raspberry Pi Final PCB Layout

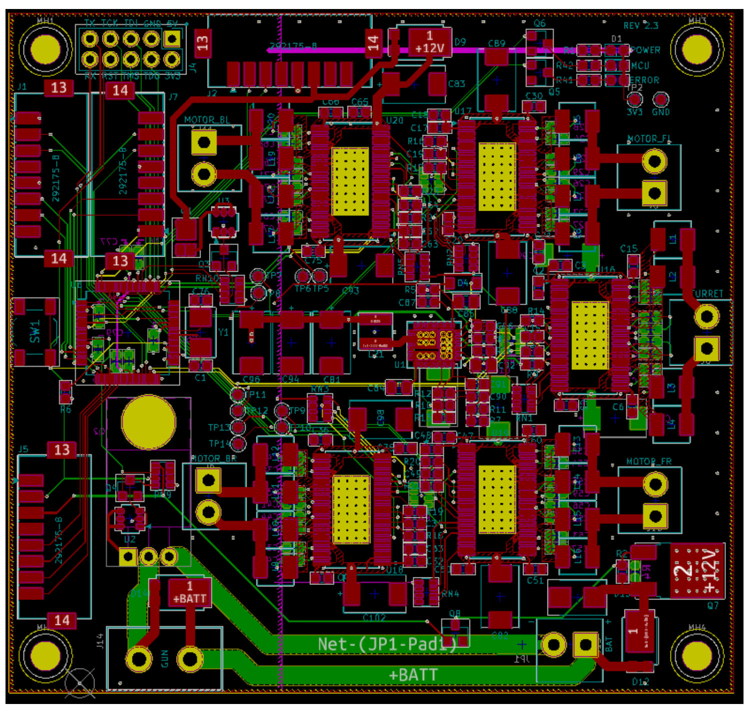


Figure 58 Motor Controller PCB Layout

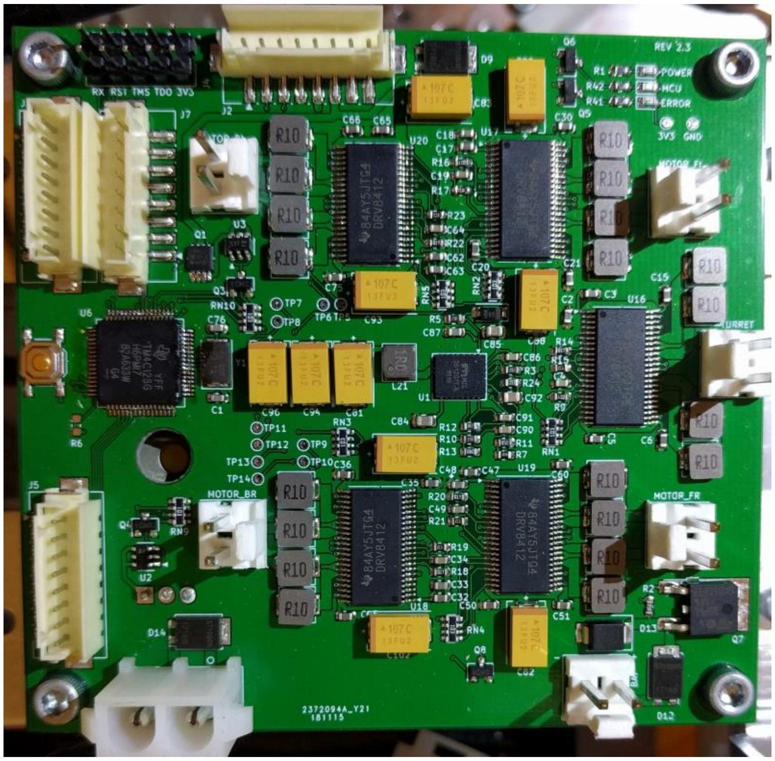


Figure 59 Motor Controller Final PCB Layout

7.3. PCB Vendors

Various manufacturers of PCB's are available to us at our disposal. The PCB vendor is an important choice as the PCB will make or break the project. Things to consider are quality, cost, and lead times. The 4 manufacturers to consider are OSH Park, JLCPCB, PCBWay, and Elecrow. These manufacturers were chosen as they are the cheapest.

The chart shown in Figure 60 lists the 4 manufacturers cost of production of a PCB vs size with and without shipping of a stencil. The chart shows the total cost including shipping for a 2-layer board $\leq 100\ mm^2$ or $3.93\ in.^2$ with and without a stencil. The dashed lines are with a stencil and the solid lines are without a stencil, shipping included.

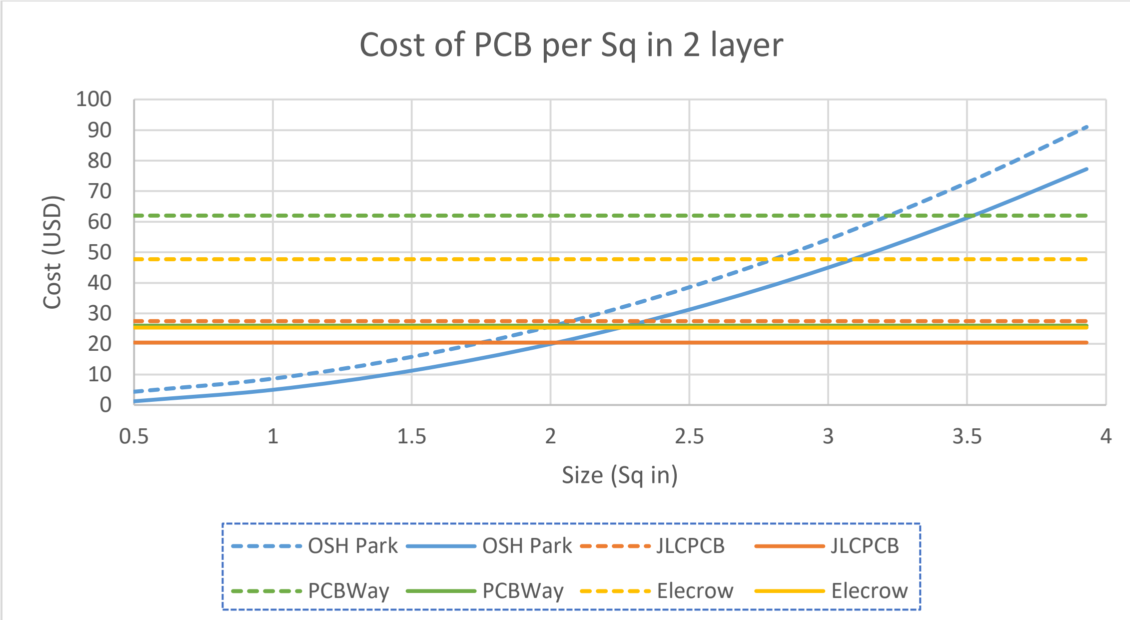


Figure 60 Cost of PCB per Sq. in for 2 Layers

7.3.1. OSH Park

As expected, OSH park is extremely cheap with its pricing per sq. inch model. However, once the board becomes larger than 2 $in.^2$ they quickly become the more expensive option. Moreover, OSH park themselves do not offer stencil manufacturing so a third-party manufacture will need to be used. This service usually takes 12 days from submittal to actual receiving of the PCB, this is a relatively quick turnaround and may be beneficial to the team if we decided to use this vendor.

7.3.2. JLCPCB

The next manufacturer, JLCPCB is both the cheapest in terms of base cost and total cost both with and without stenciling. The other options while initially close to the same price point as JLCPCB without a stencil, with the stencil included their shipping costs rise substantially.

Another option to consider is PCB's over 2 layers. There are various reasons as to why 4 layer and above boards might be used. One being the need for a dedicated ground plane for high speed signals, another being a dedicated power plane to avoid complex routing of power rails around components greatly simplifying overall design, and lastly to route large amounts of signals.

7.3.3. Final Vendor Decision

For this PCB, it was decided to use 4 layers when designing the PCB due to the need for differential signal trace size requirements, therefore a 4-layer board from the same vendors was considered. Figure 61 details the same 4 manufacturers but with 4 layers, both with and without a stencil. For a 4-layer board, all manufacturers except OSH Park reduce their max size by half down to $50mm^2$ or $2in^2$. To get the same board in 4 layers at $100mm^2$ would 250% over the cost of the 2-layer board.

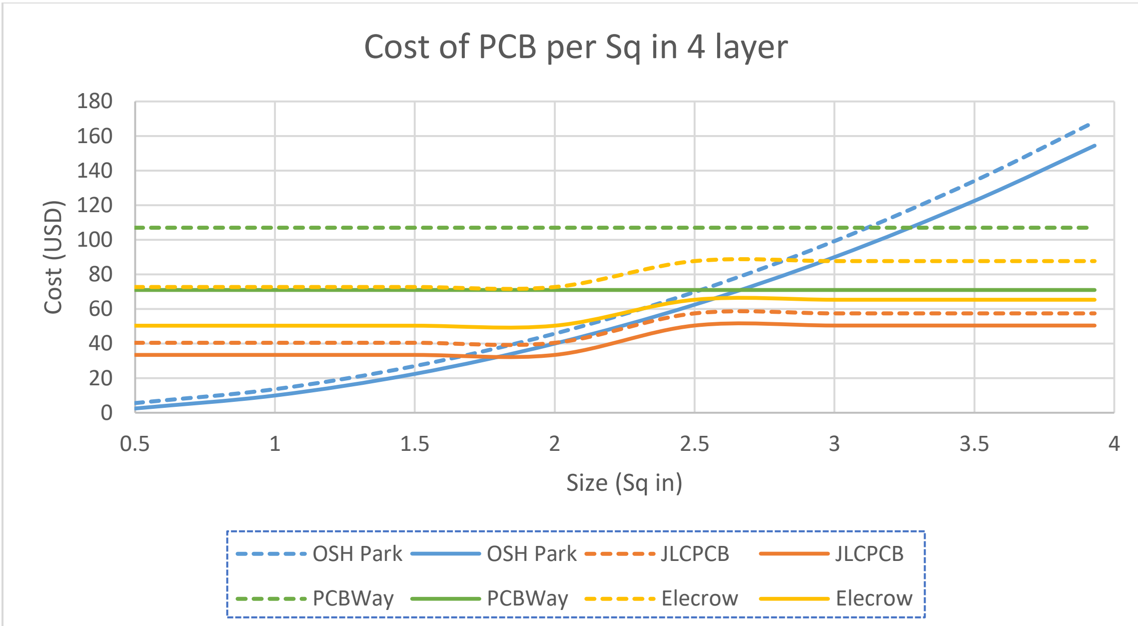


Figure 61 Cost of PCB per Sq. in for 4 Layers

7.4. PCB Assembly

Once the PCB is received from the manufacturer, it needed to be populated with components. Normally large-scale manufacturing of PCB's with surface mount components are done by using a solder paste printer to apply paste to the exposed pads. Then the components are placed onto a PCB via an automatic pick-and-place machine fed by reels of components. Lastly the board is fed into a reflow oven to be heated. The opposite side is repeated until the entire board is populated.

For smaller runs, components are placed manually either by hand or by a manual pick-and-place machine. The board is still placed into a reflow oven and the whole process is repeated for the opposite side. While overall cheaper than large runs, there is still a cost incurred by the service which places the parts, often they charge on a per part basis to place components manually.

Since most of the components on the PCB were surface mount parts, and to save costs, a stencil was used in conjunction with solder paste to apply an even coat of paste to all the surface mount pads. The challenge arises when soldering surface mount parts in this method which is soldering on both sides of the board. One method to combat this is to use higher melting point solder on one side vs the other, so the components do not fall off when the board is flipped, and the opposite side is reflowed. Since a reflow oven is not available, a toaster oven can be used as a replacement. Since the paste contains toxic chemicals, a toaster oven specifically for this task would need to be purchased, incurring additional costs.

Another method which completely avoids the purchase of a toaster oven is to design the board such that all or the most complex QFN type packages are on the same side. Then the board can be heated using a simple pan with the board achieving a uniform temperature and melting the solder onto the exposed pads. This also has an additional benefit of melting the solder from the bottom up, avoiding damaging the components by overheating them with a heat gun

With that in mind, to save on costs, the main PCB was designed with all its components except the SD card slot on the top layer. The board was then placed on top of a hot plate and heated from the bottom up.

7.5. Original Bill of Materials

Table 21 details the original bill of materials for the project. It served as a preliminary assessment of the cost of the project for consumers, neglecting the labor and design costs that would normally be implemented for the project. It also offers a price for the possible builder of the same design. Though, it provides a comparison to the requirement specification involving total cost of the bot. The success of meeting the requirement can be assessed from comparison with the results of this table. The table is laid out to display precisely which materials entered the final design as well as how many of each material was finally used. The price column represents the final cost for the quantity of that material required for the final project. A total is provided at the bottom of the table for convenience in comparison with the requirement specification related to total cost. Matching the total cost to the requirement or undershooting it is greatly desirable and remarks on the success of meeting the requirement. Figure 62 shows all parts used for the development of the TankBot during the research and development stage as well as some parts that were used in the final design.

Material	Quantity	Price
M4 Airsoft AEG w/ Metal Gearbox – Bone Yard	1	$45.00
VEX EDR Rotational Motor	2	$29.98
84T Aluminum Spur Gear	2	$6.49
42T Aluminum Spur Gear	1	$1.62
18T Aluminum Spur Gear	1	$1.67
ES2J Diode	1	$0.12
TIP110 Darlington Pair Power BJT	1	$1.36
12V 7Ah battery	2	$37.73
LiDAR	1	$39.99
Wilc3000 Wi-Fi module	1	$9.70

Camera	1	$13.49
Camera cable adapter	1	$8.99
Inductor shielded 4.7uH 652-SRN4018-4R7M	2	$0.49
Inductor PA4332.102NLT 1uH	1	$0.64
P-Channel MOSFET 78-SIS413DN-T1-GE3	1	$0.60
Regulator Dual 3.3,1.8 621-PAM2306AYPKE	1	$0.71
Regulator switching MIC261201	1	$0.00
Schottky diode 600mA BAT54GWJ	1	$0.12
Raspberry Pi Compute Module	1	$39.55
DC Geared motor - Mopar 05067591Ae	4	$160.00
Motor Control Unit L293D	4	$8.00
DRV8412DDWR – Full Bridge Driver	2	$0.00
Qauick Increment Encoder	4	$76.10
Micro USB Port	1	$0.00
MOLEX_5055680471	3	$1.98
TE_1-1734248-5 Connector	1	$1.03
TE_1473005-4 DDR2 socket	1	$0.00
N-Channel MOSFET NX7002AK	3	$0.10
LTC2953CDD-2-PBF	1	$0.00
Aluminum Platform*	1	$280.00

(*) The Aluminum Platform consists of all the nuts, bolts and the aluminum frame. But excludes the motors, Mecanum wheels and encoders used in the price.

Table 21 Bill of Materials

Figure 62 All parts for TankBot

The choices made behind the materials chosen will be summarized here for the sake of highlighting why the component was the necessary choice within the budget. The M4 Airsoft was chosen because of its relative durability to other choices and the ability to ascertain it at a massive discount. The VEX EDR rotational motors were chosen due to their reliability (as experienced by the designer) despite possibly cheaper options. All 3 spur gears were chosen due to their relative durability against other choices despite cheaper options for less durable gears.

The ES2J and TIP110 were relatively insignificant portions of the budget, as such cheaper options were not sought after due to their reliability. The Qauick decoders were originally chosen because of the reliability to cleanly output the two-phase signals of the graph. The choice to use these came at a higher cost but the outputting signals would have less noise damage. The original choice of using the Mopar 05067591AE was selected to have the performance of being able to move a quantity load over the speed it can go. This has given a rise in the cost purchasing the motors, but performance of using these motors outweigh its cost. Should an encoder not work, the simplest solution is to get another.

The motor driver DRV8432 was selected to use because it gave greater current output supply along with the voltage range to use. The current draw limit that a single motor requires for our project is only at two-thirds the limit of the rated potential of the DRV8432. Therefore, the purchase of this is the better option. Should the DRV8432 driver should not work. The second option is the DRV8842, which can meet the requirements for one motor and would need to use 4 chips to meet all four motors. The purchase of Mecanum wheels were selected so that it gives the robot more degrees of freedom of movement. Should the Mecanum wheels not work for the project, then next best is to use conventional wheels, since the frame and motor infrastructure is there. The use of Mecanum wheels will not flounder since it has been proven to work.

As for the robot’s base, the use of a custom built became the more economical and the more versatile. But should the manufacturing of this base not meet the levels of requirements, the prebuilt base of 4WD Mecanum Wheel Platform was obtained as a substitute.

8. Project Prototype Testing Plan

Although testing of the prototype has been previously discussed, the precise plans in place for testing the various systems has not yet been elaborated. That is the purpose of this section. The testing has been divided into two major categories: hardware testing and software testing. The motivation behind this division is that much the testing can be done in absence of the fully complete prototype. As such, intermediate successes can be determined along the way to verify which parts are functional in the final design should troubleshooting be necessary. This serves as a critical part of the troubleshooting that will necessarily be done to finalize the project.

8.1. Hardware Specific Testing

This section lays out the existing hardware testing that has already been done with regards to component selection and future test plans for the robot system. Much of the hardware can be tested independently of the fully constructed robot, so preliminary and prerequisite testing will often be in absence of the bot. Final testing, however, will always be performed with the final prototype bot. At that point, each of the requirement specifications, constraints, and standards were analyzed for adherence. Success will be indicated by adherence to these various items as well as many other lower-level goals.

8.1.1. Power Circuits Subsystem

For the power circuit design, not only do we need to ensure that the voltage values are within tolerances but also the ability for the circuits to perform under constant load, both electrically and thermally. Therefore, for these circuits, an electronic load was employed along with the standard DMM and input power source to test the thermal and regulation performance of the circuits.

8.1.1.1. Test Circuit 1

The power circuit for this robot consists of various small surface mount parts and due to the thermal requirements of these components, they require thermal pads under the IC's. Because of this, a simple breadboard test environment was not sufficient.

Therefore, the reference designs where placed onto a PCB and test boards where manufactured and circuits where assembled. The first power system circuits that where tested were the low voltage cutoff using a LM311 comparator, ACT4060 variable switching regulator configured with 5V and 9V outputs and the PAM2306 dual output fixed switching regulator. The test circuit is shown in Figure 63.

Of the circuits tested, the original low voltage cutoff circuit worked as intended, however because there was no graceful power down implemented the circuit would not be a good contender for a final design. The next two voltage circuits using the ACT4060 both failed under load, not even reaching 50% of the rated current on the datasheet so other regulators needed to be investigated.

The last circuit was the PAM2306 dual output circuit for the 3.3V and 1.8V rails, this is the only circuit that performed within the datasheet ratings with the voltage only dipping

6.25% from the unloaded voltage under full load. The thermal rise for the PAM2306 with thermal bias and pad underneath the IC was approximately 30°C when running at 1A.

8.1.1.2. Test Circuit 2

The next voltage regulator that was tested was for the AP65450 switching voltage regulator. This regulator has a switching frequency of 650 kHz and a continuous rated current limit of 4A. The test circuit was assembled according to the reference design provided in the datasheet.

However, when any amount of current was drawn the voltage would fall to less than 50% of the open circuit voltage. Oscilloscope readings of the SW output showed the switching frequency was approximately 1.13 MHz, almost double of the advertised switching frequency.

8.1.1.3. Final Test Circuit

The final test circuit uses the MIC261201 who's schematic is shown in Figure 41 in section 5.2.2. The circuit was tested with an electronic load and a 3-amp load was drawn from it without dipping below 5V. While the regulator chip is more expensive than the previous two, it has been proven to work in the intended application circuit. Moreover, since we were able to acquire free samples of the chip by our team.

Figure 63 Power System IC Test Circuit 1

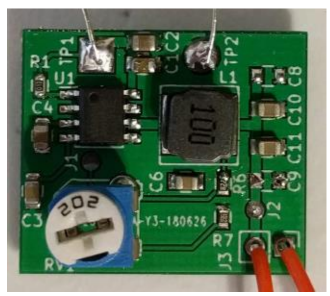


Figure 64 Power System IC Test Circuit 2

Figure 65 Power System IC Final Test Circuit

8.1.1.4. Future Power Circuit Testing

Even though the circuit by itself has been verified, once the completed board is created the power will need to be verified again. To do this the electronic load that was used in the individual circuit testing were again used to test the overall circuit. Individual power rails were tested for voltage ripple in the presence of various load conditions.

8.1.2. Turret System

Prerequisite tests were done to test only the firing of the turret under minimal control of an Arduino MCU. Figure 49 was tested for gun switching first. The pins of the MCU can only draw so much cumulative current across all the pins, so it remains an important aspect of consideration across all fields of the project, as does power consumption. The turret should be able to start and stop firing under manual control from a pushbutton activated and deactivated by a user. These prerequisite tests will also involve hooking only the gears to the yaw and pitch motor system to confirm whether the gears alone can be driven properly by the motor system.

Four pushbuttons were used to ensure proper operation in all directions of the motor system. Two pushbuttons were dedicated to the yaw system and two to the pitch system. Of the two, one will force the system in one direction and the other in the opposite direction. Successful prerequisite testing infers that basic operation is understood by the designer and further implementation is possible. Failure to succeed at these prerequisite tests suggests that the parts implemented aren't suitable at all for the project at hand and should be carefully reconsidered after determining precisely which parts are failing.

The turret system initial testing phase will consist of the turret independent of the bot, hooking it into an Arduino to act as the MCU. The turret was fully mounted within the frame assembly under full control of the yaw and pitch rotation systems. The four-pushbutton assembly described above will continued to be used to test the yaw and pitch systems under the full load of the gun. A pushbutton trigger will then test if the firing activation still works properly. Successful initial testing remarks upon the success of the completion of the minimum requirements of the yaw and pitch system for the project, as well as concurrent firing of the gun. The gun should still fire during this testing phase, and failure to do so might suggest some interconnection issues between the rotation system and the firing system. If the gun fails to rotate at this phase, it's likely that the motors aren't strong enough to rotate the system. If this is the case, different motors will need to be considered. At this point, it could also be the case that the accelerometer sensors aren't functioning as desired. In that case, re-design of the accelerometer sensors or the MCU's internal workings on the results of the sensor may be necessary.

When the initial testing phase proves successful, the turret will be ready to be installed on the bot for a secondary testing phase undergoing breadboard operation. In cooperation with the system, the Arduino MCU will be exchanged for the system's central control MCU. The turret should be able to aim quickly and accurately under the control of the central MCU guided by the CV and LiDAR systems. The turret should fire as expected within the entire system, even within a breadboard system. Successful secondary testing will remark upon the success of the interconnection of the systems, as well as the systems. This will mark complete success of the project, leading to the final implementation. Failure at this stage suggests that the port to the central control MCU was a failure. It might also suggest that the MCU is not properly guiding the system at all, which might be an error within the code itself.

This can easily be troubleshooted by using a different MCU. If the system works, the code porting was a problem, or the MCU itself was a problem. If the system continues not to work, then the code itself is faulty and will need to be scrutinized for errors. Another problem that could be encountered is mistargeting, which will likely be caused by the CV, LiDAR, or angle acquisition systems. In this event, the angle acquisition system will be questioned first to ensure the jostling of the robot and motion over rough terrain that causes vibrations in the robot aren't causing any problems. This can be tested in an independent, simplified system. Should the independent, simplified system not reveal any problems, the CV and LiDAR systems will need to be investigated for error.

With success of the secondary testing phase, the tertiary testing phase of the turret system will be under full PCB implementation. Within this phase, the bot should be fully locomotive, and as such the turret should perform as expected under moving conditions controlled by the central MCU, guided by the CV and LiDAR. At this phase, the design requirements will be scrutinized deeply to ensure that demonstration results in a success and overall operation is coherent with the design requirements laid out at the project's onset. At this final mark, the subsystem will be completely confirmed as a success. However, upon failure, something must be wrong on the PCB, which would call for an immediate questioning of the PCB. Careful investigation will need to be done to ascertain precisely what the problem is on the PCB. Once the problem is found, the secondary testing phase will restart. This loop in methodology will be continued until eventual success of the system.

In conclusion, the general testing methodology for the turret system will begin with basic, simplified testing to make sure that the ideas generated within the design are valid. Installing the gun within the frame and testing will ensure that the motors are strong enough to handle the load of the gun. Installing the frame setup with the gun on the breadboard-enabled robot will ensure further success in a model of the final implementation. Success within the final implementation itself confirms overall success of the system.

8.2. Motor Subsystem Testing

Initial testing of the motors was done with a L293D motor. The following circuit is the driver for a set of two motors to have bidirectional movement. The driver consists of 2 power inputs, VCC1 is used to power the chip while VCC2 is used to power the load. The D variant of the L293 has integrated diodes to reduce system complexity and size therefore external clamp diodes are not needed. The input signals for each of the inputs will be a PWM signal from 0 to 5 volts, or a peak amplitude at the VCC1 level. The output is the same PWM signal but having a peak amplitude at the VCC2 level. The following circuit is the initial test circuit. A LM7805CV voltage regulator is used to supply 5 volts to the integrated circuit. The orange and green signal come from the MSP430G2553 microcontroller.

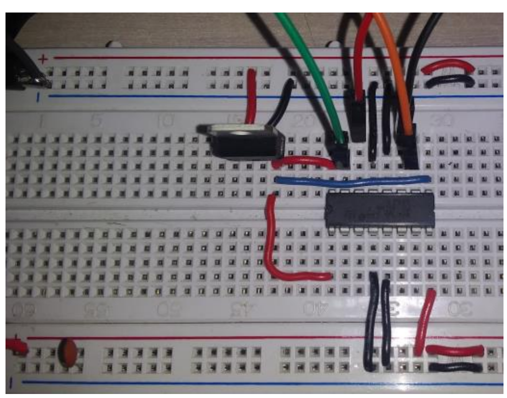


Figure 66 Motor Driver Circuit

When initial testing of this circuit was undertaken, the motors under examination were also tested. When the chosen motor was selected, the L293 motor could drive the Mopar

05067591AE without any load. But when the Mopar motor was tested for its holding torque, the motor drew about 8 Amps of current. This significantly increased the heat of the integrated circuit and was deemed unusable. The next chip for its replacement is the DRV8412. The DRV8412 circuit is shown in Figure ‘’. The further testing of this driver will be when a PCB and its corresponding components are acquired. Testing on the breadboard was not possible sing the chip comes only as a surface mount device. According to data sheet reference, the new driver should be able up to 6 amps of current at 12 volts.

8.2.1. Mecanum Wheel Subsystem

Prerequisite testing of the Mecanum wheel subsystem should involve moving just the empty encapsulation of the robot. The robot should be able to make smooth turns and generally move smoothly in a planar manner through various terrain: blacktop, smooth floor, grass, across small hills, and sand. Failure at this point would remark on general failure of the wheel system at large, demanding an immediate, all-spanning redesign of the wheel system with, perhaps, better parts to handle the situation at hand.

The initial testing phase of the Mecanum wheel subsystem will test the robot’s mobility on concrete and blacktop. From this point on, the turret system and frame should be installed to the robot, emulating the full final implementation of the robot. Under these simple conditions, the robot should be able to start and stop motion quickly. It should also be able to initiate and follow through on turns about the surface. Planar motion should be smooth despite small disruptions due to the minute imperfections of the surface. Should the robot fail to perform under these conditions, the wheel system or motors should be scrutinized to ensure they can both deal with the load and deal with the environment. Some performing maneuvers the robot should be able to do is differential driving of all four motors. This will allow the motor to shift from side to side. Troubleshooting will be done as needed.

The secondary testing phase of the Mecanum wheel subsystem will test the robot’s mobility on tougher terrains: grass, sand, dirt, etc. Like the simple conditions, the robot should be able to traverse the terrain effectively so that the patrol, follow target, and go to location modes of the robot perform appropriately. Again, planar motion will be observed and imperfections on the rough surface should not provide an insurmountable challenge for the robot. Should the robot face insurmountable challenges the wheel system or the motors should be troubleshooted for suitable corrections.

The tertiary testing phase will challenge the robot to scope small hills and elevated terrain. As in the previous phases, the robot should be able to overcome these obstacles effectively to ensure that the motive modes of the robot perform as expected. Obstacles that prove too challenging can lead to re-design of the wheel system or, more directly, the motors that were implemented within the system.

The final testing phase concerns placing obstacles within the robot’s way to test its ability to maneuver around them. This is important in all the motive modes of the robot because enemies ducking behind objects to avoid the robot should not impede the robot’s ability to follow through on its objective operation. This will require the CV and LiDAR to appropriately guide the MCU to make the right decisions to avoid crashing into terrain and objects when executing objective modes. By this point in testing, failure would suggest errors in the CV or LiDAR systems, or within the MCU’s ability to effectively guide the motion of the robot under CV and LiDAR determinations. A multi-system examination will need to be done to correct errors at this point.

In conclusion, prerequisite testing with ensure that the ideas behind the design are valid. The initial testing will ensure the wheel system can handle the load of the entire robot. The secondary testing will ensure that rougher terrain does not impede the robot. The tertiary testing will challenge the robot to handle the hardest terrain it can. The final testing confirms the robot's ability to be completely motive without direct human intervention guiding it. At this point, the wheel system will be completely confirmed.

8.2.2. Incremental Encoders Testing

To test the Encoders used for the project, the circuit will be set up as in Figure 34. With the power voltage set to 5V and the two signals fed into an oscilloscope. The incremental encoder will be tested to see what if the microcontroller can read the two directions of the encoder. The incremental encoder has two tracks that have the slots for the light source and the photodetector. To evaluate the incremental encoder, the microcontroller will wait for both signals to be at Low. The controller will then wait for which ever signal triggers first. The signal that triggers first, whether A or B, will determine if the robot is moving in the forward or backward direction. When applied to the development board, the microcontroller was not trigger. When observed in the oscilloscope the output signal was less than one volt. The following images are the outputted signals. The yellow signal is signal A and purple is signal B.

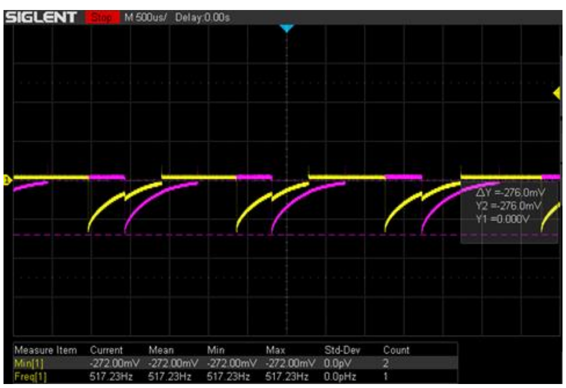


Figure 67 Encoder Output Signal (Forward)

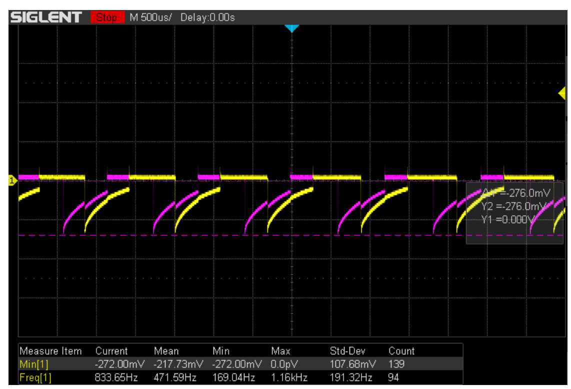


Figure 68 Encoder Output Signal (Backward)

The minimum voltage for the microcontroller to read is 3.3 volts. To solve this an amplifier would need to be used to amplify the signal. Also, the signal is outputted as impulses rather than a square wave. To fix this a Schmitt trigger will need to be used to get the impulse and change it to a square, therefore the microcontroller can read it without problems.

8.2.3. Motor Testing

To make the selected motor as a viable candidate, the motor has been tested through several criteria. The first is to test the maximum rotations per minute. Therefore, a marker, was placed to the end of the rotor and several trials were performed to count the amount of rotations performed by the motor in one minute. Five trials were performed to get an average of 90 rotations per minute. The next criteria were to find the maximum holding torque of the motor. To perform this a force will need to be applied on the rotor to create a counter rotation. To create this force, a measured block of mass was connected to the motor by a very thin string. The mass of the string is minute and added insignificant variations to the blocks of mass. Some commercial motor has the holding toque of the motor as kilogram-centimeter. This is referred as an amount of mass one centimeter from the center of the rotor. Therefore, the torque is the product of the weight of the mass by one centimeter. When testing the motor, this principle was used, and mass was added at one centimeter from the rotor. But this means that a 40 kg mass must be able to be supported from a thin string, which cannot be feasible. Therefore, the radius was extended to 10 centimeters. This means a mass of 4 kg must be suspended to create a holding torque. The average measured force created for each of the four motors is 40.41 N. Therefore, the average holding torque was at 4.04 Nm, or 41.2 Kg-cm. When the motor was at the holding torque, the current supplied by the power supply was measured. The maximum current of the motor is 8.32 A. This test confirms that the motor should be able to produce the amount of force to move this project.

8.2.4. Frame Testing

To test the frame of the robot, and see that is can handle its payload, the turret, we can simulate this by adding a force, the same as the rough weight of the collective objects and apply it for a given period. Should the frame buckle or bend extravagantly, then the frame cannot be a suitable used to house the project. In this case we will first attempted to rectify the issue with reinforcements for the parts which were struggling the most under load. If this is still not sufficient a complete redesign will have to be considered.

Another method testing of the frame is to measure the power loss across the aluminum frame when using it to distribute ground. This can determine if having the frame as ground distributor is feasible or not. The expected power loss would be at about a couple milliwatts, but with the measured data the efficiency of the power distribution across the robot will determine if the aluminum frame can be used as the distributive ground supplier. If the frame does not act as ground well enough for our purposes then we will have to explore one of two options, the first being the thickening of the ground wires, this is used if the load coming from the circuits is too large for the ground wire to safely move to ground. The second of these options is to hardwire the ground wire of each circuit into the next circuit and so on until the series is directly connected to the battery of the robot such that the circuits are dumping into the battery instead of into the frame as is conventional.

8.2.5. Camera Sensor

For TankBot to properly respond to visual inputs, we must make sure the camera systems are functional and as efficient as possible. Slow frame rates or refresh rates could be the difference between the system successfully firing and hitting a moving target versus missing behind the moving target. Testing the camera system is relatively easy as it can be connected to any desktop with a USB connection to make sure that it is functional. To make sure that the camera can be effectively communicated with we will have to be able to both send and receive information from the camera.

The process for testing a camera's resolutions is very simple, by simply taking a picture of a resolution test chart and blowing it up to examine just how many pixels can be observed in the image.

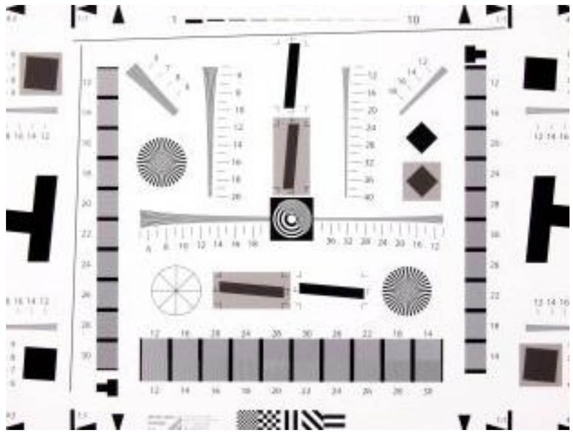


Figure 69 Resolution Test Chart

Testing the frame rates and refresh rates can be a much more involved process. A couple of options were found during the research stage of development. One method of testing frame rates involves building an array of LED's driven by a microprocessor that is counting at upwards of 10kHz. By recording the LED's over a short period of time and examining subsequent frames from the video we could see how far apart each of the pictures that make up the video are in microseconds.

8.3. Software Test Environment

The software test environment provides a way for the software to be tested in absence of the total final product. As such, it is pivotal to prerequisite and preliminary testing. The test plans related below will remark on the success of individual components to aid later troubleshooting. In fact, it is necessary to avoid heftier workloads involved in troubleshooting later. Investing time in independent testing methods allows the engineer to analyze precisely what systems are at fault in the final assembled product in an easier manner that simply doing the full analysis when a problem is initially encountered.

8.3.1. OpenCV Testing

The OpenCV software is currently being developed in house using a HP Webcam. This is not the final application of the software but will be in use until the parts get in to port the software over to our final robot design. Testing will be conducted both indoors and outdoors to make sure that ambient light from the sun or other inputs does not dramatically reduce the effectiveness of the software in determining targets are identifying motion in any way.

Once we have the actual camera we can begin laboratory testing including making sure that the software can identify the shape of a person, distinguish it from the surrounding environment using blob detection, and outline it properly using edge detection. This process will consist of many iterations of inputting a previously unseen real time image from our camera of the test site and reading the output to determine if the software is working correctly, if it is, it should correctly identify all faces and all humans.

Also, if the system is truly outperforming initial expectations, one of our stretch goals is to be able to identify friendly vs hostile targets in the software, some of the ways we might achieve this is be indicating that a gun, or the shape of a gun, or any specific shape or symbol is representative of a threat, so we would need to test various gun shapes along with various symbols from all angles to make sure that the software would identify, track, and categorize this properly.

In terms of computing environment, our software is currently being tested on a HP Laptop running Windows 10 and being programmed and developed in Microsoft's Visual Studio application.

8.3.2. Locomotion Testing

To test the locomotive software of the machine we will have a couple of subsequent tests that will start at the very most basic and move on to harder and harder commands.

The first of these commands in just getting any motion from the software at all, this requires sending a signal to the machine, having it convert that signal into a command, the command into code, then when the circuit has received the command then it is time for the software to output a signal to the motors. At this point the software must convert that command from the user into a command for the motors so that it will move the robot, finally, the encoder will need to record and report back exactly how much the wheels have moved thus how much the robot has moved. If any of these processes fails, the software is most likely to blame, and the code must be debugged to find out what is going wrong.

The second command is path finding, requiring the software to find a path to a desired location through terrain that has been analysis by the LiDAR system using the SLAM method. The software should avoid areas of extreme terrain, including walls, stationary objects, and places where the ground is too steep to safely traverse.

To test the locomotion of the robot, we will give the robot coordinates for it to move. First an unobstructed path and measure the displacement between the apparent location and the real location. As we analyze this examination, we will determine if the controller can be fine-tuned to give as little error displacement to the actual target as possible. The next examination is for the robot to go through rough terrain and see the displacement of its real location to its apparent location. The final examination is for the robot to follow a set of coordinates and see the displacement of its apparent location to its real location. One method to reduce the displacement of error is to uses vector fields, as discussed in section 6.3.3.

8.3.3. Kernel Mode Driver Test Environment

The various modules for the main system require hardware driver modules to create standard interfaces for the higher-level programs to interface with. As such, these drivers will either need to be written or modified to work properly with the target system. The test plan for kernel drivers is building debug builds and dumping stack traces when segmentation faults occur. Alternatively, when the driver is stable enough not to crash the system, log file printing will be used to debug drivers.

Drivers that interface with external hardware will also be subject to probing via an oscilloscope to verify bus signal integrity as well as GPIO control.

8.3.4. Debuggers

Debuggers are tools to break down a program's logic to find bugs in the code. Understanding of debuggers and how to interpret their output speeds up troubleshooting tremendously. There is a whole host of debuggers available at our disposal. This section describes the hardware and software debuggers that will be used in this project.

8.3.4.1. JTAG

JTAG is named after the Joint Test Action Group, which is an industry standard for verifying designs on printed circuit boards after manufacturing as well as verifying code correctness. JTAG implements standards for on board debugging using a special dedicated debugging port which uses a low overhead serial interface. This interface allows for access without requiring direct external access to the system address and data busses. The JTAG interface connects to an on-chip access port that implements a state protocol to give access to the registers inside the MCU.

Both the BCM2837 and the MSP series of MCU's all implement this interface which can be used to debug the control code run on the device.

8.3.4.2. GDB The GNU Debugger

GDB is the GNU Project Debugger, a free GPL debugging software suite for various languages including ASM, C, C++ and various others. GDB allows the programmer to debug code by specifying conditional breakpoints, examine program variables during execution, and edit program during execution. Since most of software development will be done on GNU/Linux this debugger will be an integral part in debugging the control code to verify correctness in execution.

8.3.5. C IDE

Since the software on the Compute module will be written entirely in the C language, an IDE is the preferred way of writing code as opposed to a text editor.

8.3.6. NetBeans

One of the IDE's that we will use is called NetBeans. This is primarily a Java IDE but has plugins for C/C++ support. The C/C++ editor is also well integrated with GNU GDB debugger. we can view and set variables, trace exceptions, perform system calls, add line and function breakpoints then view them in the Breakpoints window. NetBeans can inspect the function call stack and local function variables, create global variable watches, and view active threads. It can also evaluate a selected expression by hovering the cursor over it and viewing the tooltip in real-time. The Disassembler window displays the assembly instructions for the current source file.

8.4. Software Specific Testing

Each of the following sections relating to a different subsection of the project will lay out the testing process used to evaluate how each of the different parts of the robot are preforming. Included in each section will be the step by step testing procedure, the expected outcome, the actual outcome, and a discussion on how the actual outcome affects the project. Furthermore, should the actual outcome become too undesirable, the troubleshooting method for aiding the issue will be laid out.

8.4.1. Computer Vision

Testing the computer vision software is a relatively simple trial and error process. For our computer vision system to be considered a success, we must be able to consistently identify a human outline from any background, simple or complex. The testing process will consist of multiple steps starting with inputting simple images to a desktop and making sure that the software can accurately identify basic objects including people. Next, we will move on to inputting complicated images, such as scenes with multiple focuses and relatively little contrast but still just images from a database of pictures. The last testing phase will be conducted live on the robot, requiring the camera system to be able to provide the software with images that the software can then attempt to break down and analyze. There are two main options for how a machine can learn to identify shapes and patterns, called Supervised, and Unsupervised Learning.

Supervised learning is done by providing the machine learning software with a set of instances that have been correctly identified and providing what the correct output should be, this is called a "training set". The software then attempts to create a model that works for the training set and using those same images, get the exact same outputs. The machine is then put to the test by attempting to generalize the model created from the training set so that is can be applied to any case. The outcomes of new, unlabeled images are analyzed such that the programmer can provide feedback to the machine and this process is repeated until an acceptable fail rate is achieved, for us this acceptable fail rate is %10, corresponding to the 90% success rate stated in our expected outcomes for the project.

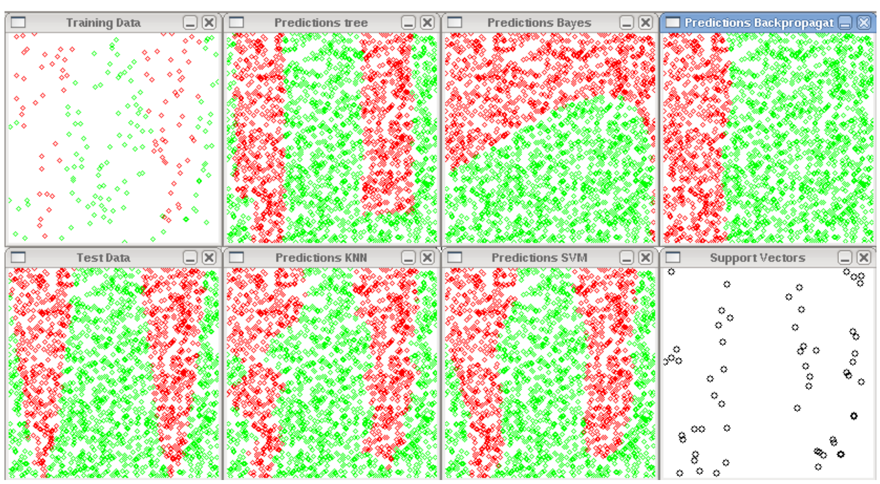


Figure 70 Supervise Learning in OpenCV

Unsupervised learning is different in that instead of giving the machine a training set, thus the images do not come pre-labeled so that the machine does not know what the outcome should be. Instead, the machine is given an unlabeled set of images and it intrinsically tries to find its own connections between the images. After connecting what makes these images positive or negative, a model is developed, tested, and improved in much the same manner as in the supervised learning method. A negative aspect of this kind of learning is that it takes so much longer for the machine to find an acceptable model than the Supervised learning method, the tradeoff comes from the fact that while the machine is working much harder, the user does not have to do any work labeling images initially.

Recently, researchers have been able to combine these two methods into what is being called semi-supervised learning. This uses a small of labeled images in a training set, plus many unlabeled images that the machine can attempt to make its model from. This has the benefit of being much faster than the unsupervised learning method while only utilizing a minimal amount of the user's time.

We have chosen to do the Supervised learning method in our project for a couple reasons, these include the fact that supervised learning is more hands on, less nebulous and thus we have more control over the input and outcomes of the machine. In addition to this the computing power needed for the machine to comes to an acceptable failure rate is far less for supervised learning than for unsupervised learning. While Semi-Supervised learning may be ideal moving forward, the research on this method is more advanced and the ability to implement and be able to properly utilize this method to achieve the outcome that we are looking for maybe to advance.

8.4.2. Wireless Communication

Wireless communication will be done though Wi-Fi on a local private network. Testing will comprise of bandwidth benchmarks as well as simultaneous streaming with command control to ensure proper function of the TankBot. The Wi-Fi signal will be transmitted using the ATWILC3000. If the wireless module fails to meet specifications, a USB Wi-Fi dongle will be used as a temporary replacement to verify the rest of the software functionality while a redesign is performed.

8.4.3. Range Finding

The range finding system we will be using is based on LiDAR technology. As such, we will need to test the accuracy of the sensor in measuring distance and sending that distance to the host. To do this the sensor is setup at a fixed distance and then sending a measure command to the sensor. The reported distance coming from the LiDAR is the maximum distance unless there are obstacles within the maximum distance. To use the LiDAR to effectively range an area, the 360-degree LiDAR will scan the point at a small degree angle, about 0.01-degree and get the range. By the spinning and getting a point at every increment of the angle, and area around the robot will be detected with all the vertical silhouette of every object. The only consideration for this is to keep objects from the robot out of the view of the lidar. Therefore, misreads cannot occur.

Another consideration of Lidar is to filter out noise that can come from the environment or from the insides of the circuit. To resolve this, the average of three to five points can be used to filter the noise out. The following is the typical LiDAR range finding output.

Figure 71 2D LiDAR Range Output

8.4.4. Main Processor Software Testing

The main program on the Compute module will control and manage all information from all sensors and secondary control systems. To test the control as a GUI will not be present, the Linux serial console will be used to manage the program on board. Remotely SSH can be used to the same effect allowing for wireless access to program debugging. Due to the high cost of a full JTAG debugger, free software debuggers will need to be used instead such as GDB. GDB allows for debugging any executable in Linux by attaching to its process and optionally inserting breakpoints to pause execution and view variable states.

For the kernel Wi-Fi driver, printk() statements are used to write to the kernel log as a simple cost-effective debugging method. While this method is simple, it is much more economical rather than the alternative of using a full hardware JTAG debugger since kernels are the second piece of software to run directly on the chip and cannot have software debuggers attached such as GDB.

8.4.5. On-board In-Circuit Debug Interface (ICDI)

The Tiva series of ARM microcontrollers supports a JTAG debugging interface called In-Circuit Debug Interface (ICDI) which is present on C Series evaluation and reference design kits. ICDI allows for programming and debugging of the on-board ARM microcontroller. The ICDI can be used with the Stellaris Flash Programmer as well as any of the Tiva C Series supported toolchains such as Texas Instruments' Code Composer Studio.

9. Administrative Content

This section details all the administrative content related to this project. It details milestones, budget and finances, and the tentative schedule for the project. Organizing these things allow for better management of the team and project, which eventually results in a better product and better documentation related to the project. This is important because the goal is to create the highest quality product that fits the requirement specifications, constraints, and standards while keeping the project relatable to those who might peer upon its contents in the future.

9.1. Milestone Discussion

The schedule for the project is shown in Section 9.3 and includes all milestones and goals set out to be accomplished within this semester and the timeframe for completion of each task. The timeframe for senior design 1 is based off the due dates for various milestones of the report. Although not required by Senior Design 1, our team will try to complete as much of the PCB design as possible. Currently we are performing software integration with various hardware devices as well as building test circuits to verify component performance.

This is key as getting the testing done as soon as possible gives us the most time to design a proper system instead of fumbling during Senior Design 2 to find parts that work. This can be seen in section 8.1 where part testing has already begun. By dividing the workload of robot base movement, turret gun control, CV target identification and Power design between the members, we can expedite the tasks which would normally be done in senior design 2.

The extra integration tasks that we are not able to finish in this semester will be done in the next. The timeframe for senior design 2 is an estimation on the time needed to design, fabricate, and assemble the finished product from scratch based on previous experiences. The time frame is tentative and subject to change at the start of senior design 2.

9.2. Budget and Finance Discussion

This project is self-funded by the team members and will not be sponsored by any company or by UCF. The following initial budget table shown in lays out the expected budget for the project. The goal was to keep the total design and production cost under $2000.00 USD. Initial testing will be done individually by each member of the team and any parts they will need will be funded themselves. Once the project is out of the design and initial component identification phase the team will split the cost of the production between the 4 of us.

Allocation	Cost
ARM MCU Dev Kit	$130
Various IC's	$200
Sensors	$200
Various Parts	$300
Vehicle Kit and Parts	$300
R & D	$600
Buffer	$270
Total	$2000

Table 22 Initial Budget Allocation

Since it is expected to go through multiple revisions of a PCB, as well as the likely possibility that more than 2 layers will be needed on a 2oz top plated board, the cost for a single member of the team to fund will be too much of a burden.

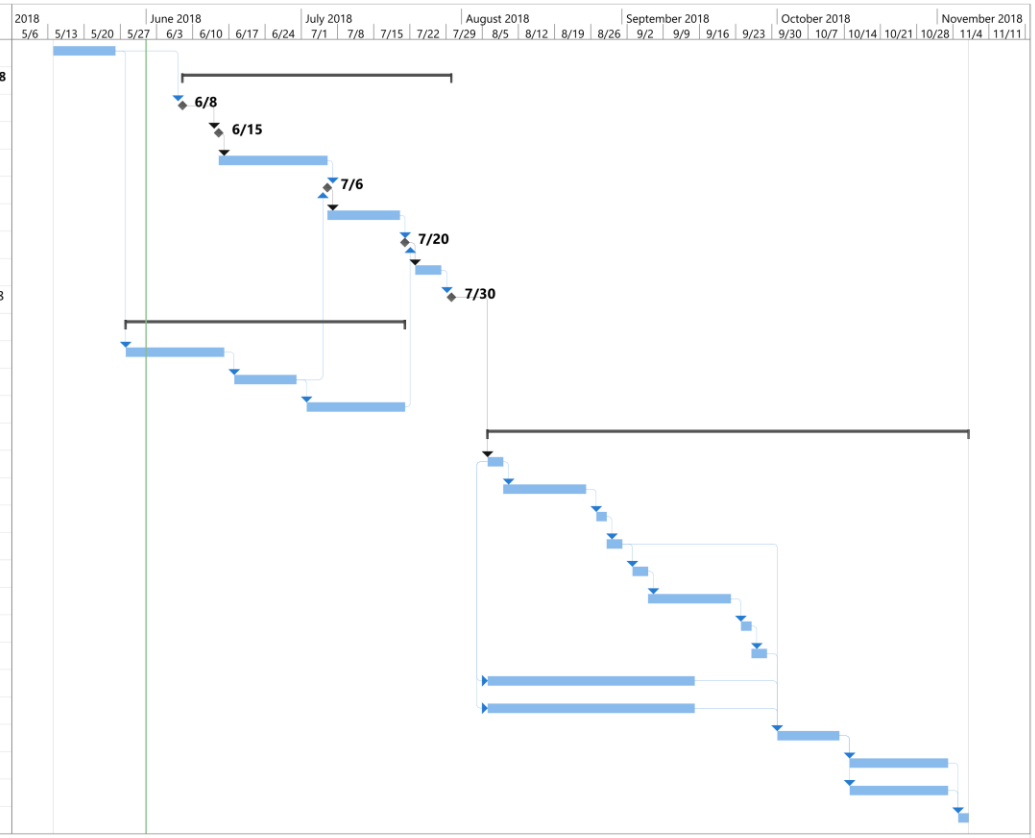
To better manage the budget and keep costs down, our team will also maximize the acquisition of free samples. Each member of the team will be able to request between 3-5 parts each.

Prototype		**Prototype**	
Part	Price	Part	Price
Pi Module x 5	160.00	Weapon	45.00
DC Geared Motor with encoder (x2)	70.00	Turret	8.00
Encoder(x4)	19.00	Self Made Body	330.00
Camera	26.99	Battery(x2)	37.73
BOM Parts per PCB revision	496.65	Motor(x2)	29.98
PCB manufacturing	378.52	Gears - Various	30.00
R&D	1000.00	Lidar	39.99
		Total To Date:	3460.87

Table 23 To Date Budget Usage

9.3. Schedule

ID	Task Name	Duration	Start	Finish
1	Choose Project Topic	10 days	Mon 5/14/18	Fri 5/25/18
2	**Report Files**	**36 days**	**Fri 6/8/18**	**Mon 7/30/18**
3	Divide and Conquer Deadline	0 days	Fri 6/8/18	Fri 6/8/18
4	Update Divide and Conquer files Deadline	0 days	Fri 6/15/18	Fri 6/15/18
5	Add 60 total Pages to Document	15 days	Fri 6/15/18	Thu 7/5/18
6	60 Page Draft Senior Design 1 Doc	0 days	Fri 7/6/18	Fri 7/6/18
7	Update and Add 40 Pages	10 days	Fri 7/6/18	Thu 7/19/18
8	100 Page Submission	0 days	Fri 7/20/18	Fri 7/20/18
9	Finalize Report	5 days	Mon 7/23/18	Fri 7/27/18
10	Final Hard Copy Due HEC 418 12pm	0 days	Mon 7/30/18	Mon 7/30/18
11	**Design**	**40 days**	**Mon 5/28/18**	**Fri 7/20/18**
12	Order Components	15 days	Mon 5/28/18	Fri 6/15/18
13	Test Components	10 days	Mon 6/18/18	Fri 6/29/18
14	Develop Software For Components	15 days	Mon 7/2/18	Fri 7/20/18
15	**Production Tentative (SD2)**	**67 days**	**Mon 8/6/18**	**Tue 11/6/18**
16	Design Prelimiary PCB	3 days	Mon 8/6/18	Wed 8/8/18
17	Fabricate Design	12 days	Thu 8/9/18	Fri 8/24/18
18	Assemble PCB	2 days	Mon 8/27/18	Tue 8/28/18
19	Test PCB	3 days	Wed 8/29/18	Fri 8/31/18
20	Revise PCB Layout (if needed)	3 days	Mon 9/3/18	Wed 9/5/18
21	Fabricate Design	12 days	Thu 9/6/18	Fri 9/21/18
22	Assemble PCB	2 days	Mon 9/24/18	Tue 9/25/18
23	Test PCB	3 days	Wed 9/26/18	Fri 9/28/18
24	Build Chassis for Robot	30 days	Mon 8/6/18	Fri 9/14/18
25	Build Turret	30 days	Mon 8/6/18	Fri 9/14/18
26	Assemble Robot	10 days	Mon 10/1/18	Fri 10/12/18
27	Test Robot	15 days	Mon 10/15/18	Fri 11/2/18
28	Implement and Update Control Software	15 days	Mon 10/15/18	Fri 11/2/18
29	Finalize Robot	2 days	Mon 11/5/18	Tue 11/6/18



Project: SD1Project Schedule
Date: Fri 6/1/18

Task | Split | Milestone | Summary | Project Summary | Inactive Task | Inactive Milestone | Inactive Summary | Manual Task | Duration-only | Manual Summary Rollup | Manual Summary | Start-only | Finish-only | External Tasks | External Milestone | Deadline | Progress | Manual Progress

Page 1

10. References

[1] R. Pérez-Peña, F. Robles and E. Lichtblau, "Orlando Police Defend Actions as Clock Ticked in Massacre," New York Times, 20 June 2016. [Online]. Available: https://www.nytimes.com/2016/06/21/us/fbi-transcripts-orlando-shooting-omar-mateen.html. [Accessed 7 July 2018].

[2] L. Blain, "South Korea's autonomous robot gun turrets: deadly from kilometers away," New Atlas, 7 December 2010. [Online]. Available: https://newatlas.com/korea-dodamm-super-aegis-autonomos-robot-gun-turret/17198/. [Accessed 17 July 2018].

[3] M. Rosenberg and J. Markoff, "The Pentagon's 'Terminator Conundrum': Robots That Could Kill on Their Own," New York Times, 25 October 2016. [Online]. Available: https://www.nytimes.com/2016/10/26/us/pentagon-artificial-intelligence-terminator.html. [Accessed 17 July 2018].

[4] J. Pike, "Global Security," Global Security, 7 November 2011. [Online]. Available: https://www.globalsecurity.org/military/world/rok/sgr-a1.htm. [Accessed 30 June 2018].

[5] C. Nelson, K. Nelson, A. Perez and C. Cuteri, "Autonomous Target Recognition System," University of Central Florida, 2016. [Online]. Available: http://www.eecs.ucf.edu/seniordesign/fa2016sp2017/g02/. [Accessed 30 June 2018].

[6] M. Kelemen, I. Virgala, T. Kelemenová, Ľ. Miková, P. Frankovský, T. Lipták and M. Lörinc, "Distance Measurement via Using of Ultrasonic Sensor," *Journal of Automation and Control. Vol. 3, No 3,* pp. 71-74, 2015.

[7] J. Li and S. Piatek, "hamamatsu," Hamamatsu Photonics K.K., [Online]. Available: http://www.hamamatsu.com/jp/en/community/optical_sensors/articles/LiDAR_guide_to_autonomous_vehicle_market/index.html. [Accessed 26 June 2018].

[8] H. Shaftel, R. Jackson and S. Callery, "How climate is changing," NASA, 18 July 2018. [Online]. Available: https://climate.nasa.gov/effects/. [Accessed 20 July 2018].

[9] D. Sharkov, "KILLER ROBOT ARMY: GOOGLE, TESLA WARN AI KILLING MACHINES COULD DEVASTATE MANKIND," Newsweek, 21 August 2017. [Online]. Available: https://www.newsweek.com/google-tesla-and-over-100-experts-rally-against-killer-robots-652566. [Accessed 20 July 2018].

APPENDIX A
Copyright Permissions

APPENDIX B
Source Code

Source code for the project is available at the public project git repo.

https://bitbucket.org/account/user/sd2018_group1/projects/SEN

APPENDIX C
Datasheets

Datasheets available at the following project git repo

https://bitbucket.org/sd2018_group1/datasheets/src