

Introduction to Engineering Design with Professional Development 1

Final Report for

Real-Time Tracking and Control Via GPS Technology

Team 4

Section 2

Instructors: Bill Mielke, Linda McCloskey



Version 1.0

May, 10, 2010

Prepared by:

Lucas Doyle (Computer Systems Engineering)

Matthew Williamson (Computer Systems Engineering)

Peter Willette (Mechanical Engineering)

Mike Vo (Mechanical Engineering)

Mike Kessler (Mechanical Engineering)

Chris O'Neil (Mechanical Engineering)

Chris Vena (Aeronautical Engineering)

1 Abstract

Matthew Williamson

The men and women who protect the borders of the United States face many challenges and dangers every day, from the vast swaths of land they are charged with monitoring, to the threat of armed and dangerous illegal immigrants. To this end, new technologies can help to increase the effectiveness, as well as the safety of those who patrol the borders of the United States. This project aimed to create a mobile platform to monitor areas of the border. This product would allow a single border patrol agent to monitor large areas of the border, while remaining out of harm's way.

To achieve this goal, a robotic platform was developed that integrated remote control and video monitoring with Global Positioning System technology. This system provides a live video feed of the environment around the robot, as well as motion control and position tracking, without the need for a border patrol agent in the field.

These efforts resulted in a functional video streaming and location monitoring system. The locomotive system faced many setbacks, however, and was not completely operative at the close of this project. Overall, this system demonstrated the potential of this idea, if more time was spent on completing the design, and following the design process more closely.

2 Table of Contents

Real-Time Tracking and Control Via GPS Technology	1
1 Abstract	2
2 Table of Contents	3
3 Tables of Figures and Tables	4
4 Introduction:	6
5 Project Objectives & Scope	7
5.1 Mission Statement	7
5.2 Customer Requirements	7
Lucas Doyle and Peter Willette	7
5.3 Technical Specifications	9
6 Assessment of Relevant Existing Technologies	10
7 Professional and Societal Considerations	12
8 System Concept Development and Selection	13
9 Subsystems	20
9.1 Frame and Weather Proofing	21
9.2 Track System	22
9.3 Drive Train Analysis and Design	25
9.4 Power Distribution Circuit	32
9.5 Low Level Control	36
9.6 Computer and Wireless Web Server	40
9.7 Global Positioning System	44
9.8 Maps	46
9.9 Camera	48
9.10 High Level Web Control	49
10 Results and Discussion	50
10.1 Significant Accomplishments	50
10.2 Results and Conclusions	51
11 References	53
Appendix A: Selection of Team Project	55
Appendix B: Customer Requirements and Technical Specifications	56
Appendix C: Gantt Chart	59
Appendix D: Expense Report	60
Appendix E: Statement of Work	62
Appendix F: Lessons Learned	63
Appendix G: User Manual	65
Appendix H: Configuring Ubuntu to login and connect to Wi-Fi at boot	67
Appendix I: Intro to HTML	67
Appendix J: Selection of Video Streaming Technique	69
Appendix M: Motor Documentation	72
Appendix N: GPS Documentation	74
Appendix O: Battery Documentation	82
Appendix P: DC-DC Converter Documentation	83
Appendix Q: IED H-Bridge Documentation	99
Appendix R: Servo Documentation	115
Appendix S: Low Level Code	116
Appendix T: High Level Code	118

3 Tables of Figures and Tables

Figure 1: WowWee Rovio (WowWee Group Limited).....	10
Figure 2: PackBot (Kod*Lab)	11
Figure 3: Initial Concept of Ice Thickness Robot.....	14
Figure 4: Initial Concept of Hazmat UGV	15
Figure 5: Initial Concept of Border Patrol UGV	16
Figure 6: Final CAD Design.....	19
Figure 7: Hierarchical Overview of Subsystems	20
Figure 8: Cross Section of frame and weatherproofing system.....	21
Figure 9: Overview of Entire Frame and Weatherproofing Subsystem.....	22
Figure 10: Pros and Cons Comparison of Tracks and Wheels (Hornback, 1998).....	23
Figure 11: CAD of a drive wheel, consisting of two sprockets and a hub connecting them.	24
Figure 12: CAD Drawing of Tread Piece.....	24
Figure 13: Tracks made from rubber hose segments riveted to two bike chains.	25
Figure 14: Final Rear Hub Assembly	28
Figure 15: Concept Sketches for Tensing Mechanism	29
Figure 16: Sectional View of Front Axle Assembly.....	31
Figure 17: 9V DC-DC Converter Circuit Diagram	34
Figure 18: DC-DC Converter as Implemented in Final Design	34
Figure 19: Overall Circuit and Control Diagram. Power and ground signals are	35
Figure 20: PWM Signal Examples(Hirzel, 2008).....	36
Figure 21: RC Filter Circuit	37
Figure 22: RC Filter Circuit as Implemented in Final Design	37
Figure 23: The Arduino Microcontroller (Arduino, 2010)	39
Figure 24: HS-422 Servo.....	39
Figure 25: View of Computer as Installed on Robot	40
Figure 26: General Appearance of Website	44
Figure 27: The Real-time GPS Control.....	47
Figure 28: GPS Search Form (Before and After Search).....	47
Figure 29: View of the Camera on Website	48
Figure 30: Flowchart Displaying Flow of Control Signals through the Robot	50
Table 1: Customer Requirements and Relevant Technical Specifications	9
Table 2: Related Products Used for Benchmarking.....	11
Table 3: Initial Design Ideas Eliminated from Contention	13
Table 4: Final Concept Decision Matrix	17
Table 5: Potential Propulsion Methods	22
Table 6: Comparison of Buying and Making Tracks.....	23
Table 7: Voltages Required by Devices on Robot(CompuLab, 2009),	32
Table 8: Current Drawn by Devices on Robot Under Various Loads (CompuLab, 2009), (Documentation found in appendices K, N and O)	32
Table 9: Comparison of 30Ah Equivalent Batteries(All-Battery.com, 2009),(BatteryJunction.com, 2006), (BatteryMart.com, 2010)	33
Table 10: Pros and Cons of Design Project Choices	55

Table 11: Key Customer Requirements and Relevant Technical Specifications	57
Table 12: Total Project Costs	60
Table 13: Time to Pay Off the Purchase of Six Robots	60
Table 14: Potential Approaches for Streaming Remote Video from Robot to Customer	70

4 Introduction:

Peter Willette

The first thing that the team decided on was what project to choose from. In order to make the selection process easier, projects that there was no interest in or didn't fit the team's skills were immediately eliminated. After these eliminations, the remaining choices were global positioning system robot, home automation, water purification, campus recycling and a robotic transporter. After this list was made there was a list of pros and cons constructed for each of these possible project choices (See Appendix A). Based on this list, and the fact that the skills present in the design group were heavily geared towards mechanical and computer engineering the final choice on the overall project was made. With the project decided upon, there were still many directions that the project could go in. Ten possible implementations of GPS tracking technology were developed. A list of cons was used to narrow the ten solutions down to three (See Section 8). Once the three choices of a Border Patrol, Hazmat Detection and Ice Thickness robots were decided upon, a decision matrix (See Section 8) was constructed in order to choose a final project.

Once this project was decided upon, appropriate benchmarking was conducted (See Section 6) to determine what needed to be implemented in terms of the entire system and also the various subsystems. With this benchmarking complete, process of actually designing the robot began. The Solid Works CAD package was the basis for the design of all of the mechanical subsystems. For the electronic subsystems the knowledge of the team along with what the IED shop had to offer was sufficient for this project as far as implementation of a prototype. Once all of the subsystems were designed the building and assembly process began. The team split into electrical and mechanical sections to focus more heavily on each of their specialties. This entire process was completed with several weekly meetings that all members of the group attended.

In today's world, the topic of security is an issue of high priority. More recently the security of the Mexico and United States has become an issue of concern. There are major bouts of violence occurring due to the ongoing drug war. In an attempt to prevent this drug violence and the illegal immigration, the United States Border Patrol monitors the border. This has become a dangerous and expensive operation. The goal of this project was to allow the human element to be taken out and the amount of manpower required to patrol the border to be reduced. With the human element taken out the safety of the officers would be greatly improved. Also, with less man power required to patrol more of the border, less monetary resources would have to be devoted to this portion of border patrolling. This topic can be seen in greater detail in sections 5.2 and section 7.

5 Project Objectives & Scope

Peter Willette

- The project objective was to complete a robot that synthesized real time GPS tracking technology to meet the border patrol's needs as a customer.
 - Inside Scope:
 - Useable camera during the day and night
 - Moveable from remote location
 - Easy to use
 - Weather resistant
 - Able to negotiate various terrains
 - Easily transportable via motor vehicle
 - Out of the Scope:
 - Fully operational in an inverted position
 - Resistant and defensible to human attacks
 - Very long range camera capabilities
 - Triangulation of exact humans position and not the robots position
 - Unrestricted wireless operation

5.1 Mission Statement

Matthew Williamson

This project will produce an intuitive, user-friendly robot that will allow border patrol agents and other security professionals to increase their effectiveness while also increasing their safety by moving them out of harm's way.

5.2 Customer Requirements

Lucas Doyle and Peter Willette

The customer base for this project consists of US border patrol personnel. Repeated attempts made to contact a field agent of the US Border Patrol as well as relevant personnel in the Department of Homeland Security were unsuccessful. As a result, the best option available to the team was to use online resources. The requirements were therefore compiled based on assessments made from the US Border Patrol website. These assessments were made with regard to the demands that the robot would be subject to given what the average border patrol agent does in a given day.

These Requirements are listed in order of priority, with the most important requirement listed first. For more detail on these requirements, see appendix B.

Customer Requirements Included:

- 1. Intuitive Control** – This was defined as having an average user able to operate the robot with no more than ten minutes of training. Make it as easy as possible to leverage advanced, high level function of the robot through its controls.
- 2. Weather Resistant** – This is defined as being able to withstand direct sunlight, two inches of rain and winds up to ten mph.
- 3. Transversal of Varied Conditions** – This was defined as being able to negotiate a three inch high obstacle, and different surfaces such as sand, gravel, grass and up to three inch stones.
- 4. Ability to determine location** – Defined as the ability to send GPS coordinates to a base computer.

- 5. Long Range Cameras** – This was defined as the user being able to distinguish a person from 100 yards away, in the daylight.
- 6. Able to be transported in a pickup truck** – This was defined as the entire robot being no more than six feet long by four feet wide.
- 7. Night vision** – This was defined as the robot being fully able to function at night.

Customer Requirements Excluded:

Fully operational in an inverted position

The components that the team had were not suitable for an extra servo which would have made it possible to rotate the camera and allow the robot to see upside down. Another possibility for this would have been to have the software invert the image; however this was out of the knowledge of the members on this team.

Resistant to and able to defend against human attacks

Resistance to attack would have meant covering the robot with more expensive materials with a greater resistance to impact. As far as the robot defending itself, both lethal and non lethal force would not be looked highly upon as this is a school project. If implemented in a real situation, tasers and other more lethal weaponry could be incorporated into the design.

Optical zoom for very long range camera capabilities

This could be achieved by simply upgrading to a better camera. For the scope of this project it was not necessary or relevant to do so.

Triangulation of actual human's position

This could have been achieved by adding a laser range finder, however these are very expensive. Alternatively, more advanced software could have possibly solved this issue, however this was out of the range of knowledge of this group.

5.3 Technical Specifications

Group

Customer needs are not specific enough to develop a prototype with measurable performance characteristics. Without these measurable characteristics, it is impossible to determine whether or not a prototype was successful. To this end, all customer requirements were mapped to technical specifications shown in the following table.

	4 hour min. run time	10 mile min range	3'x2'x2' max	Withstand 100psi	2 inches of rain	Withstand drop 5ft	120 temp resists	Maximum 30 lbs	Under 50 decibels	15x optical/digital zoom	Minimum 2 handles	1 external charging port	less than 15 inputs in program	2 inches of ground clearance	45 degree approach and departure angles	GPS onboard	16 infra red leds on board
Long Range	o	o															
Fits in back of a car			o					o									
Tough				o	o												
Weather resistant					o		o	o									
Quiet								o									
Easy to move in and out of vehicle			o								o						
Wide angle of view																	
Easy to charge												o					
Intuitive control													o				
Traverse varied conditions		o			o		o							o	o		
Long range cameras										o							
Ability to determine location															o		
See at night																o	

Table 1: Customer Requirements and Relevant Technical Specifications

6 Assessment of Relevant Existing Technologies

Mike Vo

Patent Research

Using Google Patents to look for any possible robots that are similar to this robot yielded very little information. Though there were patents for robot design and robot control systems, those patents were of little use for they do not fit the purpose of what the intention of this robot.

Benchmarking

Looking at some of the available products from other companies, two products were benchmarked because of how well each robot implements their system. Those two products were the Rovio from the Hong Kong based company, WowWee and the military robot PackBot by iRobot. The WowWee Rovio was benchmarked for its ability to be controlled from any device with an internet connection. This robot provided an example of how to implement the webcam as well as the control system into this surveillance robot. The PackBot from iRobot provided an example of chassis design as well as a global positioning system (GPS) device integration.

The WowWee Rovio is a robot that, among many other things, implements a webcam. According to the WowWee website, the user will see and hear everything the Rovio sees and hears. The user can control the Rovio from anywhere in the world, using a personal computer, a Mac, a smartphone, a cell phone, or any other device, just as long as the device has an internet connection. The user can maneuver the Rovio to any area only as long as there is no obstacle objecting its path and the robot's batteries are not depleted.



Figure 1: WowWee Rovio (**WowWee Group Limited**)

The PackBot by iRobot is part of a series of military robots and is currently being used in Iraq and Afghanistan. The PackBot is a durable robot and has a GPS integrated into it. In the video demonstration of the robot found on YouTube, the robot is seen being thrown into a room, hitting the hard (concrete) floor, and driving off. As the video plays on, the robot is shown driving off a high ledge (about 20 foot high), landing on grass, tumbling down a hill and was still able to drive away.



Figure 2: PackBot (**Kod*Lab**)

The PackBot implements a GPS into its system. Using the GPS, any soldiers or other military personal is able to know the location of the robot as well as the location of any enemy combatants.

Using both the Rovio and the PackBot as an example, the information gathered helped provide how the surveillance robot should operate and what this robot should look like. The Rovio provided an example of how the surveillance robot should incorporate the webcam and how the control system for the robot should be. The PackBot provided an example of how to implement the GPS and how to make the robot durable.

Though both are provide an example of how to make this robot, both are expensive. The WowWee Rovio is \$299(WowWee Group Limited) and the PackBot is \$115,000 (Robot Stock News, 2006). While the cost of the WowWee Rovio is lower, it is designed for indoor applications, and is thus unsuitable for border patrol.

Website	Relation to this project
http://www.battlekits.com/	Drive Train
www.mnfirstregional.org/docs/splash08/Splash08_Drive train.ppt	Drive Train
engineering.unlv.edu/first/teams/pdf/first_robatics.ppt	Drive Train and Chassis
http://www.rctankcombat.com/articles/track-systems/	Drive Train
http://www.mcmaster.com/#table-top-conveyor-belts/-6wz3rk	Drive Train
http://www.kadtronix.com/robotframe.htm	Chassis
http://www.trossenrobotics.com/store/p/3181-Traxter-Robot-Chassis.aspx	Chassis
http://www.active-robots.com/products/platforms/	Chassis
http://www.omninerd.com/articles/Waterproofing_Electronics	Waterproofing
http://videosift.com/video/Golden-Shellback-Electronics-Waterproofing	Waterproofing
http://www.submarineboat.com/waterproofing.htm	Waterproofing
Somewhere on the internet – thing about the xbee radios / Arduino	Wireless Control

Table 2: Related Products Used for Benchmarking

7 Professional and Societal Considerations

Mike Vo

The implementation of this surveillance robot will have significant impact on the people who use it, the environment that it is used in, and the safety of the people that use it, as well as the safety of those that are tracked by it. As of now, the needs to protect the borders from those who enter the country illegally require the need of placing people on the border. Placing people on the border puts the lives of these patrolmen in danger, because those that enter this country illegally may be criminals.

The number of crimes associated with drugs has risen in recent years. The border between The United States and Mexico has grown more and more dangerous. On March 15, 2010, Mexican drug dealers killed a US consulate employee and two family members in Ciudad Juarez (Hall, 2010), which is across the border from El Paso Texas and both are roughly about 1.4 miles away from each other. Again in Ciudad Juarez, between May 1 and May 2, 2010, 25 people were killed (My Fox Atlanta, 2010). The use of this robot may help save lives by replacing the need to have humans monitor the border. This robot will replace the human aspect of watching the borders. The need to physically place border patrolmen on the border will no longer be needed and the risk that having patrolmen on the border will be eliminated.

The robot will have an impact on the safety of illegal immigrants as well. Illegal immigrants crossing the border face dangerous conditions. Those coming to The United States illegally through the Mexico-US border will be exposed to the extreme weather that the desert offers. Temperatures in this region during the summer months are generally in the 90s and can surpass triple digits, while temperatures in the cooler months are more comfortable during the day (City Rating, 2002). At night, no matter what day it is in the year, temperatures drop dramatically. Temperatures can drop to freezing even during the summer months. These rough conditions will pose a risk to anyone, especially the illegal aliens. The night vision and the camera view of this robot can help find illegal aliens in the desert. Finding them may save their lives, because they will not be exposed to the extreme weather conditions that this region has to offer.

If the robot performs its task correctly, it will help crack down on illegal aliens and help stop them from coming into this country. Limiting the flow of illegal aliens into this country can allow money to be spent elsewhere. Billions of dollars are spent on programs such as food stamps and welfare for illegal aliens. Illegal aliens also tend to take the unskilled jobs (and therefore money) from Americans who would otherwise have them. With fewer illegal aliens in this country, less money will be spent on supporting them and more jobs will be available for American workers.

8 System Concept Development and Selection

Chris O'Neil and Chris Vena

The skill sets present in the design group are heavily geared towards mechanical and computer engineering and programming. Therefore, the group has chosen to pursue the Lockheed sponsored project entitled 'Real-Time Tracking and Control Via GPS Technology' (see appendix A to see details about this decision). Upon reviewing the problem statement and requirements, a list of possible solutions were briefly outlined. However, many of these design ideas were quickly dismissed for the reasons listed below:

Concept	Reason it was not Pursued
Bomb detection robot	Robot needs to withstand explosion
Fire rescue	Robot needs to be fireproof, GPS not useful indoors
Forestry mapping of a trail	Misapplication of robotics (hiker is much better suited), target customer (forester) highly unlikely to use this
Pesticide sprayer	Robot would need to be very large to be effective, difficulties testing
Corn maze cutter	Would be very difficult to implement without gasoline engine for roto-tiller
Fish Detector	Electronics in the water, boat based solution
Flying forest fire detector-helicopter/plane	Airplane based, weak customer base
Avalanche detection/causation	Requires use of explosives, difficulties testing
Fish detector	Boat based, fish move around and are difficult to track
Mine detector / remover	Would need to be able to withstand an explosion

Table 3: Initial Design Ideas Eliminated from Contention

Concept One: Robot Designed to Measure Ice Thickness

Problem:

Many winter activities occur on the ice of frozen lakes. In order for these activities to be safe, it is crucial that the thickness of the ice is accurately known. The most common current method for measuring ice thickness involves having a person walk out on the ice and drill a hole until they reach the water. This puts people at risk of falling into the frigid water. A robotic device could provide accurate readings of the ice thickness without risking a human life.

Customers:

This device would be useful for lakeside towns and villages where ice fishing is a common winter activity. Additionally, ice rescue teams from fire and police departments could use the data that it would provide to find the safest route to a person that has fallen through thin ice.

Customer Requirements:

- Intuitive controls
- Can traverse varied conditions such as ice, snow and slush
- Buoyant in case it falls through the ice
- Provides an accurate measure of ice thickness
- Displays ice thicknesses on digital user interface
- Able to be lifted and transported in the bed of a truck

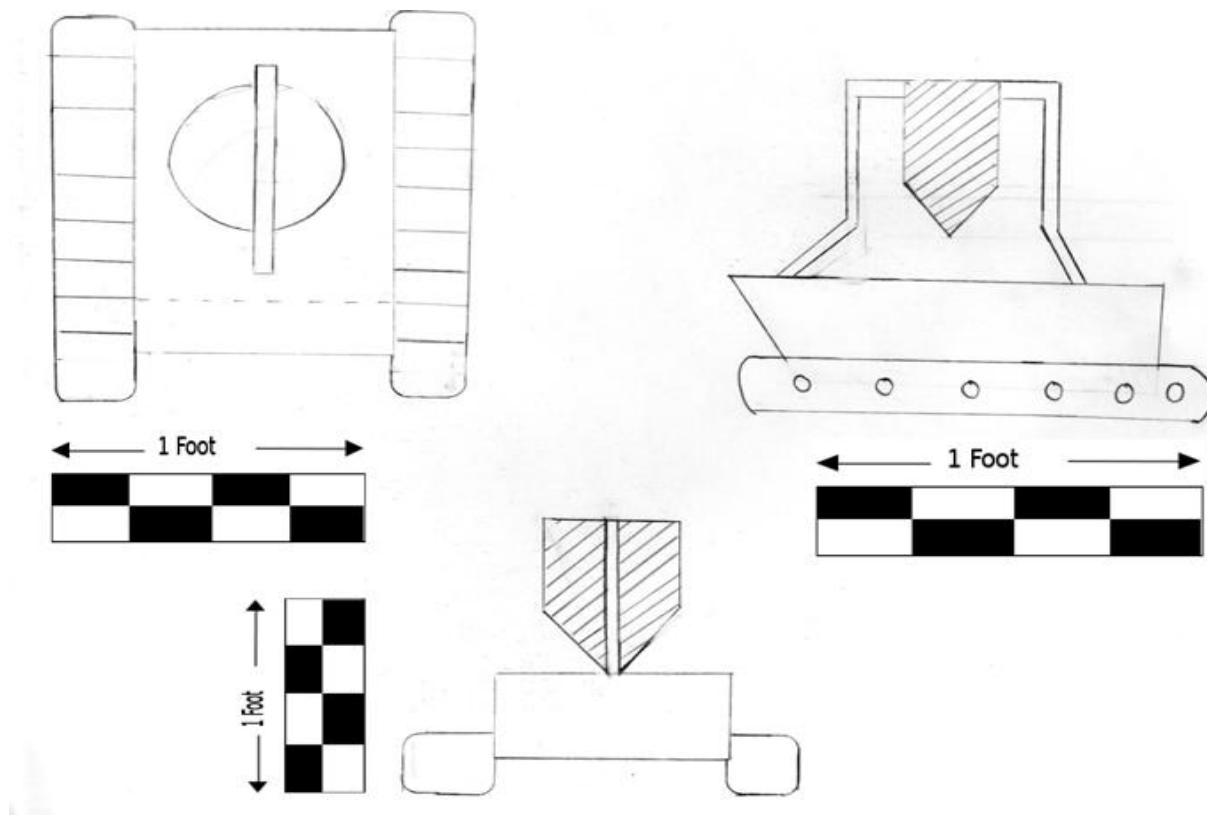


Figure 3: Initial Concept of Ice Thickness Robot

Concept Two: Unmanned Ground Vehicle (UGV)

Problem:

Many environments around the world are unsafe or potentially unsafe for human exposure. Examples of such environments include disaster zones, nuclear power plants, and other areas with high concentrations of chemical or radiological substances. This unmanned ground vehicle would take a human operator out of harm's way to monitor the environment. Furthermore, it would also determine and display on a map which areas are safe or unsafe for humans.

Customers:

Government hazmat squads, nuclear power plants, missile silos and other chemical or nuclear facilities would find the most use out of this product. It could also be used to monitor any potential spillage, radiation leak and the safety of workers.

Customer Requirements:

- Land based robot equipped with multiple sensors such as a geiger counter and other chemical sensors
- GPS coordinates track robots position and the position of radiological and chemical levels
- A map displaying a log of past positions and sensor data will be able to be browsed by the user
- Able to be controlled by average person after at most ten minutes of training
- Able to be transported in the bed of a truck

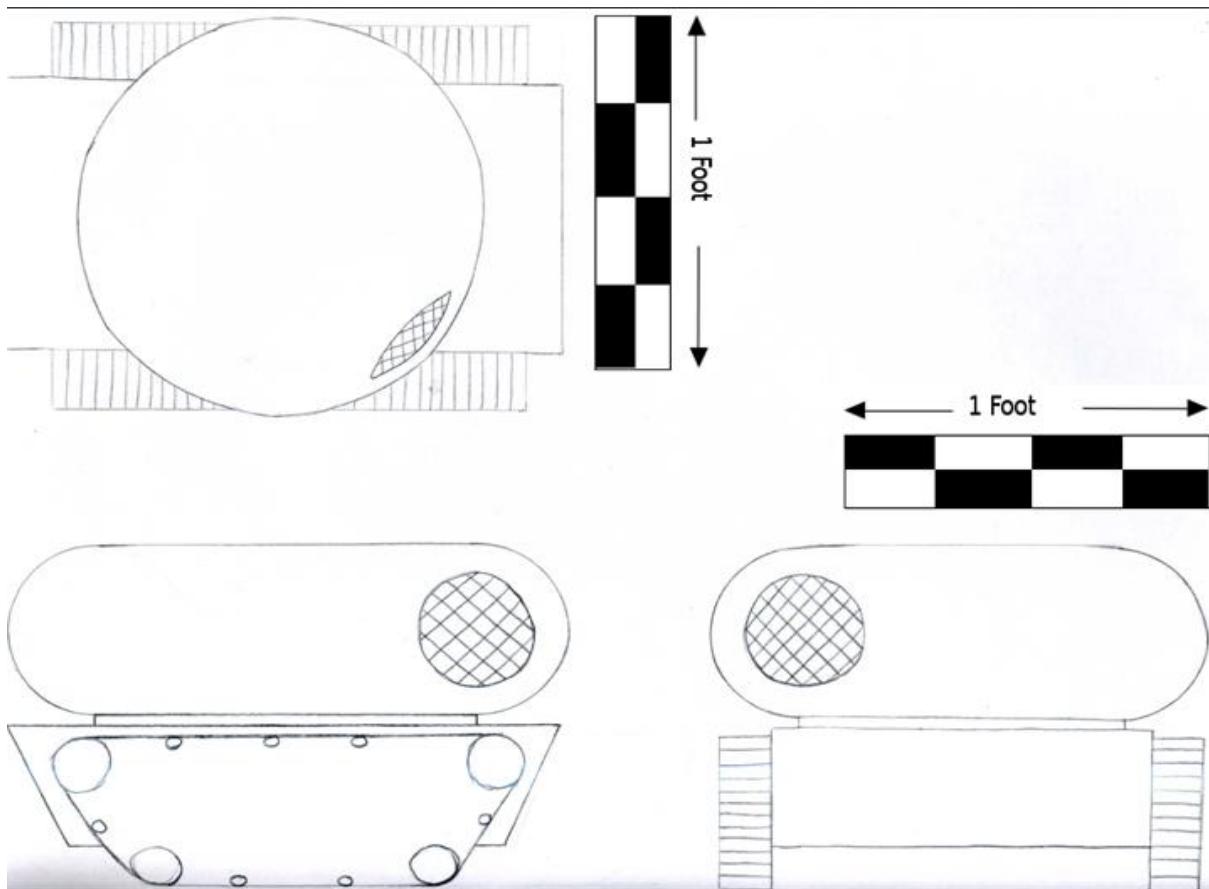


Figure 4: Initial Concept of Hazmat UGV

Concept Three: Border Patrol UGV

Problem:

Illegal immigrants oftentimes cross the border into the United States. A land based robot, equipped with a camera and GPS would allow border patrol and other law enforcement agencies to tag and track the location of these illegal immigrants as they enter the country.

Customers:

By purchasing and making use of this product, a country's federal government would be able to identify and track trespassers at the country's border. Additionally, documenting the generated data over time would allow law enforcement agencies to identify where the immigrants are primarily entering the country.

Customer Requirements:

- Able to be controlled by average person after at most ten minutes of training
- Can traverse varied conditions and terrain such as small rocks and sand
- Long range cameras that are able to identify a person
- Ability to track and determine location
- Nightvision equipped
- Weather resistant
- Able to be transported in a pickup truck

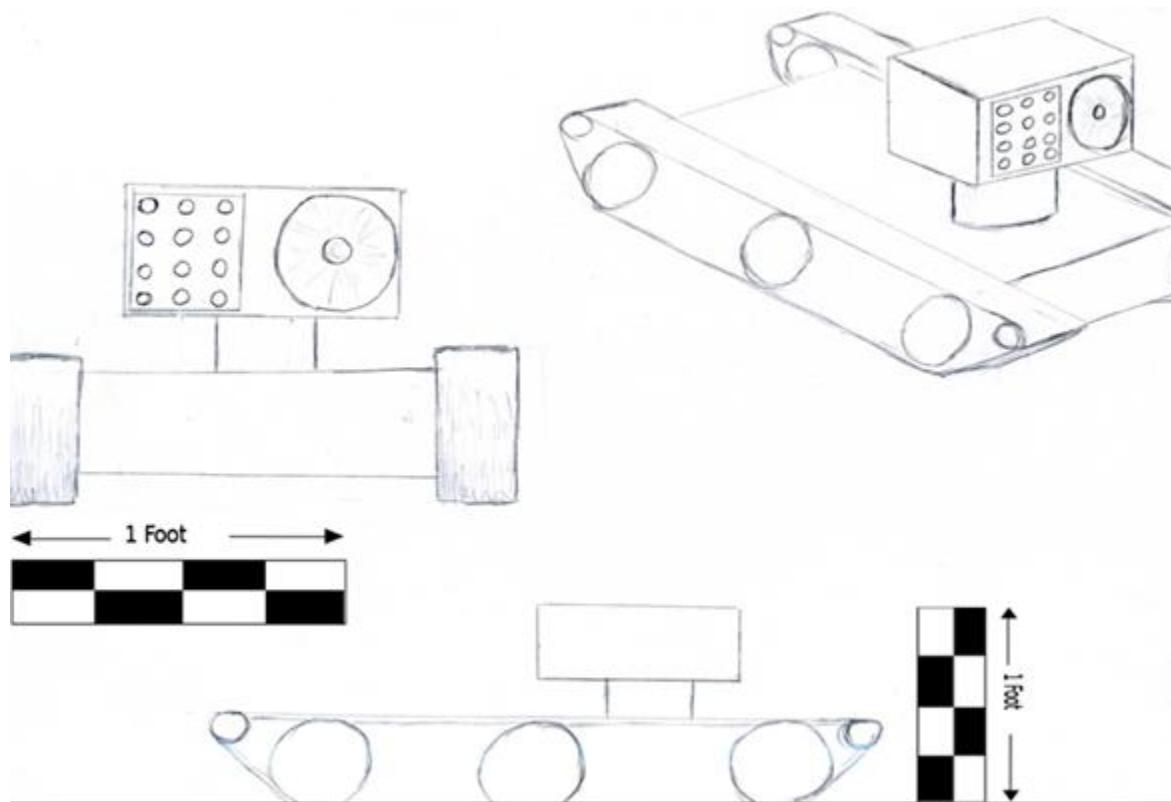


Figure 5: Initial Concept of Border Patrol UGV

Decision Matrix:

		A		B		C	
	Segment	Border Patrol		Hazmat Detection		Ice Thickness	
Selection Criteria	Weight	Rating	Wtd	Rating	Wtd	Rating	Wtd
Cost	25%	3	0.75	2	1.00	1	0.25
Potential Market Size	20%	3	0.60	2	0.40	1	0.20
Complexity of Systems	20%	3	0.60	2	0.40	1	0.20
Necessary Range	10%	1	0.10	3	0.30	2	0.20
Testing Feasibility	5%	4	0.20	1	0.05	1	0.05
Ease of Operating Environment	5%	3	0.15	1	0.05	2	0.10
Effectively Utilizes GPS	15%	3	0.45	3	0.45	3	0.45
Total Score		2.85		2.65		1.45	
	Rank	1		2		3	

Table 4: Final Concept Decision Matrix

Based on the criteria shown in Table 4, the Border Patrol UGV was the best decision. The first element compared between the project ideas was cost. Overall the Border Patrol robot had the lowest combined cost to the group because some group members had access to vital components needed for this specific robot. As for the Hazmat Detection, extra money would have to be allotted to pay for the various detectors necessary. The Ice Thickness robot would also have been fairly expensive because of the cost for the drill. Potential market size was also another very important aspect in deciding which project to pursue. Again the Border Patrol robot seemed best suited for this. The government is a very large customer base that is very interested in this project. The Hazmat Detection robot also has a relatively large customer base in the government, but would be used less because its primary function is in the event of a nuclear meltdown or other hazardous material accident, which are relatively rare occurrences. The Ice Thickness robot received low marks in this section because it had a lack of a large, definable customer base. The final major criterion taken into account for the project selection is the complexity of the systems. The Ice Thickness robot had major problems associated with it because of the drill it uses. It would have been necessary to have a very large power supply that was not readily available for this project. The Hazmat Detection robot would not have been too difficult to use for this project, however they are expensive. For the Border Patrol robot, the only major system that had to be integrated to the robot and GPS is the camera. The group had a webcam ready to use and servos to rotate it. In addition, it and the GPS could be integrated through Google Maps.

There were some other minor aspects that went into the group's decision making process. The first discussed is testing feasibility. The Ice Thickness robot would have been nearly

impossible to test because winter was ending and there was nowhere it could be properly tested. The Hazmat robot would also have been difficult to test because there is no easy way to simulate a nuclear meltdown or hazardous material spill. The Border Patrol robot is by far the most simple to test based on the customer requirements. Each of these robots had a different area that they had to cover. The Border Patrol robot had to cover an extremely large range all along the border, while the Hazmat Detector robot only had to cover a more confined area. As for the Ice Thickness robot, its area of use is a lake, which can be very large, but still manageable. The environment in which these robots would be tested also played a role in the group's decision. As noted earlier, the Hazmat Detection and Ice Thickness robots would have been difficult to test and experiment with because the environments they are designed for would be difficult to simulate. Although it would be challenging to recreate the summer heat that the Border Patrol robot would have experienced, the terrain and other aspects would not have been too hard to create. The final criterion discussed was the effective use of the GPS. The Border Patrol robot would record the location of any illegal alien entering the country. The Hazmat Detection robot would use the GPS to mark out a perimeter where people could safely enter, however it would be limited in the fact that the GPS is not effective inside. The Ice Thickness robot uses the GPS to mark locations where the ice is to thin for any activity. It would then take this data and map out the lake and show which spots were safe.

Final Decision and Justification

The border patrol UGV was the best choice. It had the lowest cost and a large potential customer base, most notably the government. This concept also has the capability to assist more people than the either of the other two concepts. It helps the Border Patrol, allowing them to guard a larger area along the border without wasting manpower. In addition, many people try to cross the barren desert in the Southwest United States and are often unprepared for their journey. With the border patrol UGV, it is possible to find them before they get lost or run out of supplies. Overall, there is a greater potential customer base for this concept than either the ice thickness or hazmat UGV.

The border patrol UGV also faces fewer major technical hurdles. For the hazmat UGV, the electronics must be hardened to resist radiation and other corrosive substances. The ice thickness UGV must be waterproof, and the drilling mechanism adds complexity to the vehicle. The border patrol UGV needs fewer complex systems, as its main purpose is to monitor its surroundings via a camera. Given the time and budget constraints, the border patrol UGV proved to be the most realistic concept to pursue.

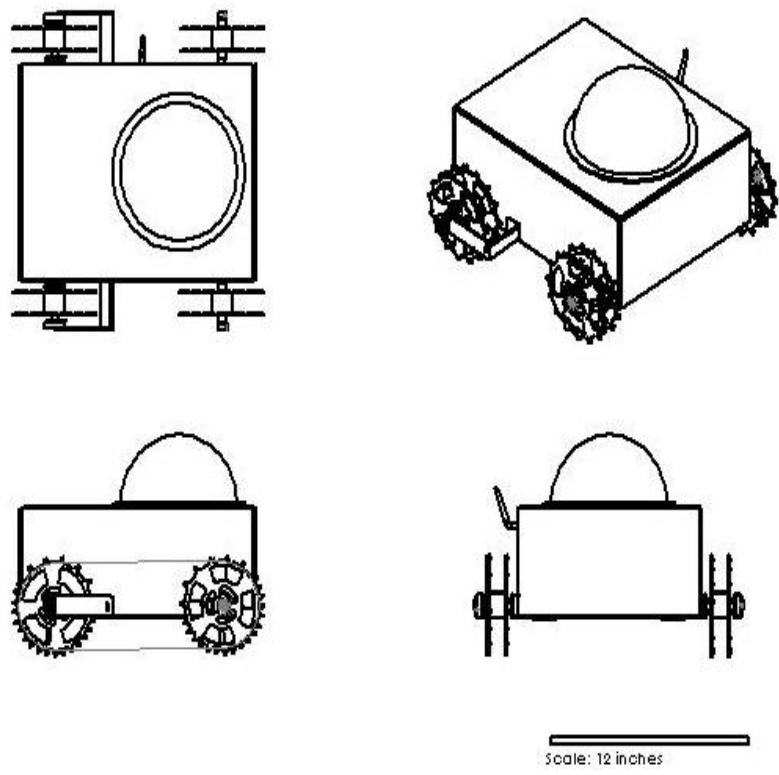


Figure 6: Final CAD Design

9 Subsystems

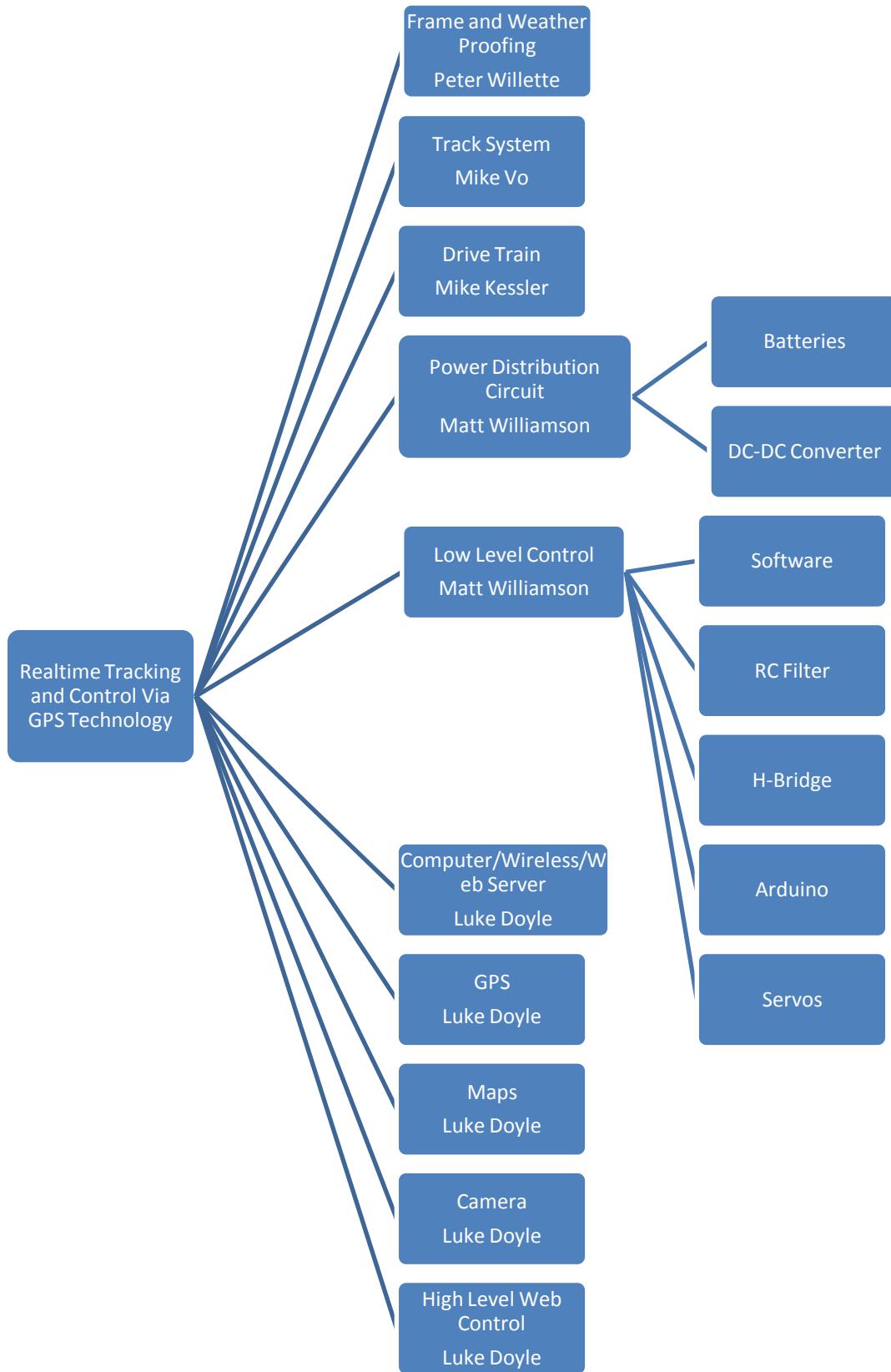


Figure 7: Hierarchical Overview of Subsystems

9.1 Frame and Weather Proofing

Peter Willette

For the design of the entire chassis alternative designs were considered based on references from first robotics teams.

The materials selection process consisted of comparing various materials such as steel, aluminum, plastic and other composites. In the end steel won out due to its resistance to cracking and its high strength properties. Aluminum was ruled out because it is a very difficult material to work with due to the fact that if welded it becomes weak and is prone to fatigue. Plastics and other composites were ruled out due to a lack of strength or higher price than steel (respectively). The only way to truly test this design was to look at what had been done in the past, modify what had been done to fit the needs of this project and then implement the design. One large area that was overlooked was the designing detail of exactly where all brackets, mounts and anchors would go and what they would look like. This was a major downfall when it came to the final assembly of the robot. The components did not fit in as well as they could have.

The materials selection and design process was simple for the selection of weatherproofing materials. Plastic was a low cost material that stood up to all of the requirements necessary to complete a prototype. Benchmarking was undertaken by looking at industry's solutions to weather proofing electronics. Several options were explored such as the "EBOX Weatherproof Electronics Enclosure (Jjenco, 2006)" all other alternatives were of this nature, except for one. A clear untraceable coating was found that can waterproof any electronic while keeping its full capability. This would have been an effective solution to this problem, however it is still in development and not available to the public yet (Covert, 2008). This system can be tested simply by leaving the robot outside on a sunny day, which will fulfill its requirement in order to withstand direct sunlight. As far as the rain resistance requirement, a bucket of water and or a hose can be implemented to douse the robot with water.

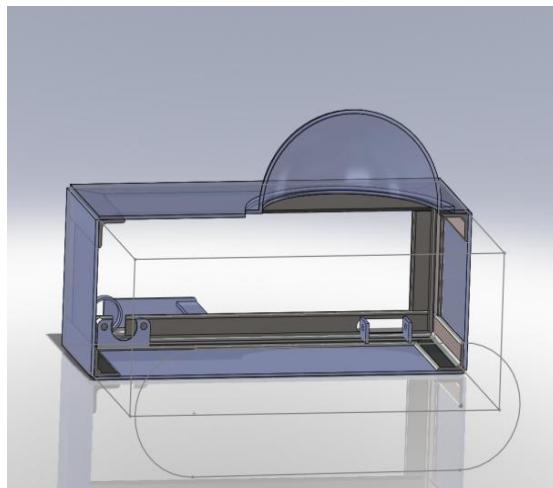


Figure 8: Cross Section of frame and weatherproofing system



Figure 9: Overview of Entire Frame and Weatherproofing Subsystem

9.2 Track System

Mike Vo

To determine the method in which the robot would propel itself, extensive research was performed. This research yielded three possibilities:

Track	 (Wikipedia)
Wheel	 (Wikipedia)
Weg	 (Kod*Lab)

Table 5: Potential Propulsion Methods

Do to the complexity of implementing them with the robot the wegs were quickly eliminated, leaving only the track and the wheel. To find the best option of the two, further research was done and a list of pros and cons was created. Looking at the list in Figure 10, the tracks beat out the wheel.

Study Results	<i>Tracked Vehicles</i>	<i>Wheeled Vehicles</i>
Route Flexibility	✓	
Cross Country Mobility	✓	
Traction on Slopes	✓	
Road Speed		✓
Logistics		✓
O&S Costs		✓
GVW, Volume, & Payload	✓	
Maneuverability/Turning Radius	✓	
Transportability	✓	
Weight Growth Potential	✓	
Gap & Obstacle Crossing	✓	

Figure 10: Pros and Cons Comparison of Tracks and Wheels (Hornback, 1998)

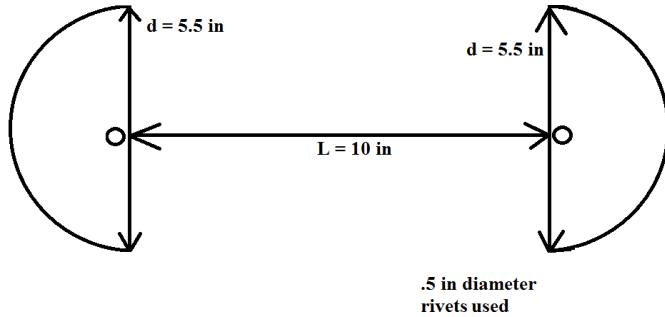
After deciding upon the tracks, the decision to buy or make the tracks was needed. To help decide which option is the best, research was done and a list was made. Below is a chart of how the decision to make the tracks was chosen instead of buying the tracks. This chart was based on anecdotal information found on a number of websites.

	Buying Tracks	Making Tracks
Cost	X	
Performance	X	X
Durability	X	X

Table 6: Comparison of Buying and Making Tracks

Before designing the tracks, more research was needed. The design of the sprocket as well as the chain tracks was accomplished thanks to information and tips found online at rctankscombat.com/articles/track-system. To meet the requirement of clearing a 2" high obstacle, the diameter of the sprocket must be at least 5", so the sprocket was set at the minimum diameter of 5". To find the number of treads that is needed to make the tracks, calculation was done as seen below. Based on the calculation, 160 treads are needed and the dimensions of the treads are .75" wide and 1" long. The design of the sprocket and treads attached to the robot can be seen in figures 11 and 12. The material used for the sprocket was decided to be steel, due to the strength and durability that steel possess.

Track/Tread Calculation



$$\begin{aligned}
 C &= \pi d \\
 &= \pi d + 2(L) = \frac{1}{2}(n) \\
 &= \pi(5.5) + 2(10) = \frac{1}{2}(n) \\
 N &\approx 75 \text{ link} \\
 \text{Total treads} &= 150 \text{ treads}
 \end{aligned}$$



Figure 11: CAD of a drive wheel, consisting of two sprockets and a hub connecting them.



Figure 12: CAD Drawing of Tread Piece.

Initially, the plan was to water jet the treads as well as eight sprockets out of a sheet of steel. However, due to the size and number of the treads, the treads were not able to be water jetted. Due to this setback, alternatives were needed, and three options were proposed. The first was to enlarge the size of the treads. Enlarging the treads would have allowed the treads to be water jetted, but with the enlarged size of the treads, it may have stiffened the chain and not allowed the chain to bend, thus making the track ineffective. The second option was to keep the treads at the same dimension and plasma cut all 160 tread pieces. This option would have been very slow and the pieces may not have been in symmetry with one another. The third option was to cut the treads out of a rubber garden hose. This option was most appealing because the rubber treads could be cut easily with shears and in the same dimensions as the metal treads. This third option was chosen and implemented for the final design.



Figure 13: Tracks made from rubber hose segments riveted to two bike chains.

After the rubber treads were cut and riveted to the bike chains, when testing the chain on the sprocket to see if the sprocket fit, a problem arose. The teeth on the sprocket did not line up with the treads in the chain. Every other tooth of the sprocket would hit every other rivet of the chain. Though the dimensions of the sprocket were calculated correctly, human error was the cause of this setback. The rubber treads were intended to be riveted to the chain two inches apart from one another, but instead were riveted one inch apart. Due to a lack of time, the track system was eventually abandoned for a basic wheel system. The wheels were made out of the sprockets, wrapped with the rubber hose, which mimicked a tire. This allowed for a basic demonstration of the drive control system, although the robot wasn't able to move under its own power.

9.3 ***Drive Train Analysis and Design***

Michael Kessler

The drive train of the robot was designed with several points in mind:

Goals of the drive train take from customer requirements:

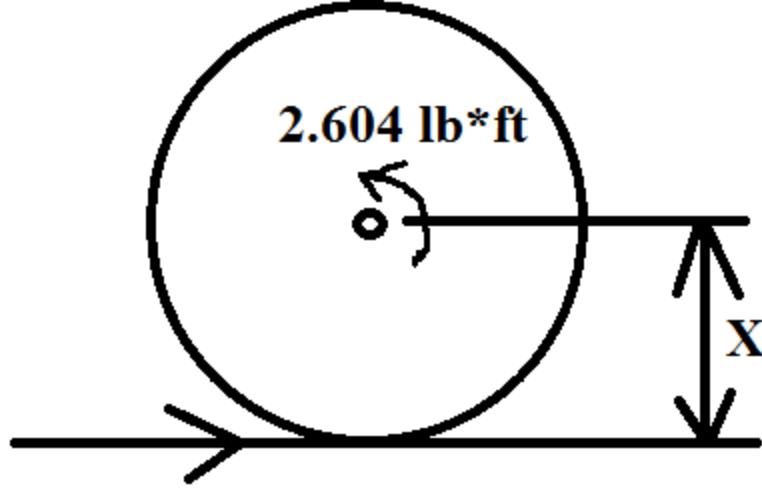
- Robot must be able to navigate a 45 degree incline
- Speed of approximately three mph
- Withstand a five foot drop
- have a minimum of two inches of ground clearance

Other goals of the system that were considered when selecting shaft diameters and hub materials:

- Support the weight of a robot of about thirty pounds
- Withstand outputted torque of motors
- Have functionality built in to tense and align the chain track system

The rear hubs and motor mounts were designed around the Pittman gear reduction motors provided by the IED lab. These motors were chosen for several important reasons; the variety of mounting options, the accurate data sheets available, and the internal gear reduction that eliminates the need for any external gearbox. Additionally, the motors were available at no cost to the group. The motors were mounted in the plane of the lower frame rail in order to maximize ground clearance, which is approximately two inches. These motors provide an adequate amount of torque for this application, theoretically enabling the robot to negotiate a 100% grade with a five inch drive wheel diameter. Calculations for required wheel diameter and predicted top rotational speed are shown below.

Wheel Diameter Calculation:



$$500 \frac{\text{oz}}{\text{in}} * \frac{1 \text{ lb}}{16 \text{ oz}} * \frac{1 \text{ ft}}{12 \text{ in}} \\ = 2.604 \text{ lb} * \text{ft}$$

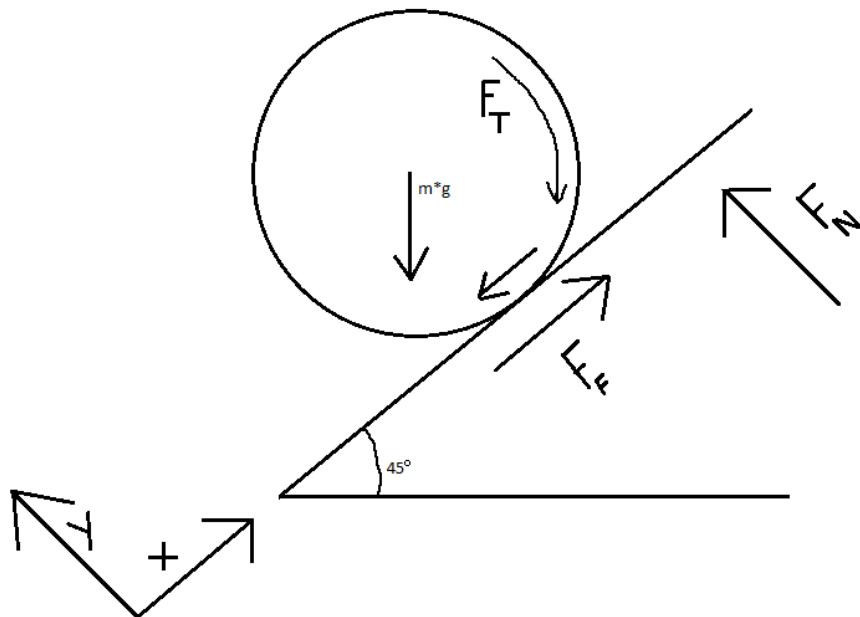
$2.604 = F_{opp} * x$, where x is the radius of the wheel

$$35 \frac{\text{lbfm}}{32.2} = 1.088 \text{ slug} = 24.74 \text{ lbs} = F_{opp}$$

$$X = \frac{(2.604 \text{ lb} * \text{ft})^2}{F_{opp}} \quad (\text{multiply by 2 for there is 2 motors})$$

$$X = .21 \text{ ft} \times 12 \frac{\text{in}}{\text{ft}} = 2.5 \text{ in}$$

Top Rotational Speed Calculation:



$$F_f = \mu F_N \quad F_D = F * D = F \left(\frac{5 \text{ in}}{12 \text{ in}} * 1 \text{ ft} \right)$$

where D is the diameter of the wheel and was found to be 5 inch

$$\Sigma F_X = F_f - m * g * \cos(45) - F_T + F_N * \cos(45)$$

$$\Sigma F_Y = -m * g * \sin(45) + F_N * \cos(45)$$

$$\begin{aligned}
 &= -(30\text{lb})(32.2\text{ft/s}^2)\sin(45) + F_N*\cos(45) = 0 \\
 (.2) F_N - (30\text{lb})(32.2\text{ft/s}^2)*\cos(45) - F(\frac{5}{12}\text{ft}) + F_N * \cos(45) &= 0 \\
 \mathbf{F \approx 8\text{ft*lb} = 1536 \text{ oz*in}}
 \end{aligned}$$

$$\frac{\mathbf{1536 \text{ oz * in}}}{2} = \mathbf{758 \text{ oz * in per motor}}$$

Max rpm ≈ 40 rpm

To transfer the rotational force from the motors to the sprockets, the rear hubs needed to be designed to withstand the loading from the sustained abuse encountered when negotiating harsh terrain. They are supported at either end with ball bearings to eliminate cantilevered loading and greatly increase the life of the electric motors. Aluminum was chosen for its high specific strength, light weight and ease of machining. Set screws were utilized in transferring rotational force from the motors to the hubs. Calculations for these hubs are shown below, including expected loading from a drop of five feet to fill the appropriate customer need.

Dynamic loading from drop calculation:

$$\begin{aligned}
 PE &= mgh; KE = \frac{1}{2}mv^2 \\
 PE_1 &= KE_2 \\
 \frac{1}{2}v^2 &= gh; v = \sqrt{2gh} = 17.944 \frac{\text{ft}}{\text{s}} \\
 &\quad \text{impulse - momentum} \\
 mv &= F\Delta t \\
 \Delta t &= .1 \text{ s} \\
 m &= \frac{30 \text{ lbm}}{32.2 \frac{\text{lbm}}{\text{slug}}} = 0.932 \text{ slug} \\
 \frac{mv}{\Delta t} &= F = \mathbf{167.1 lbf}
 \end{aligned}$$

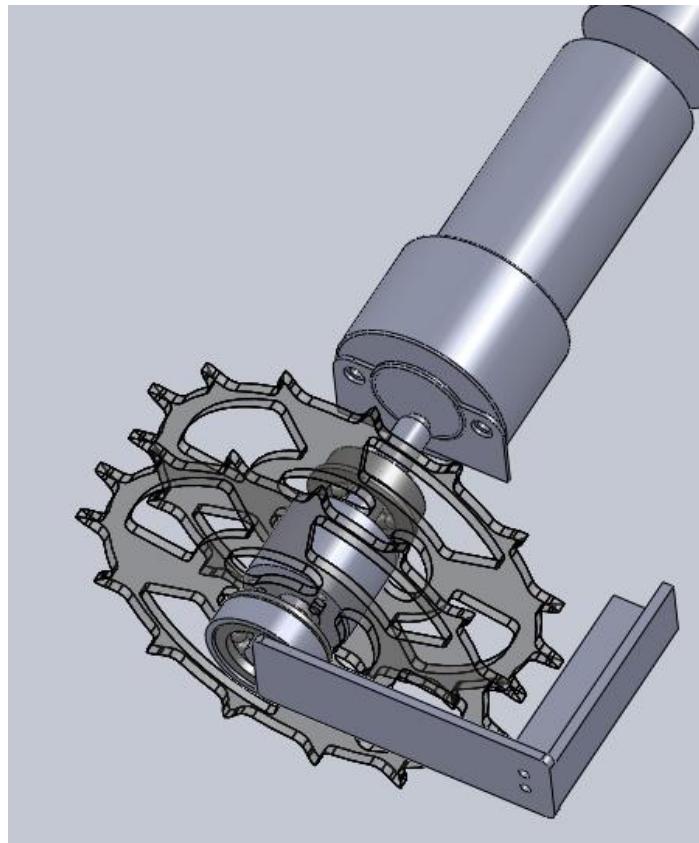
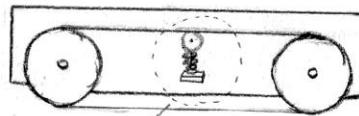
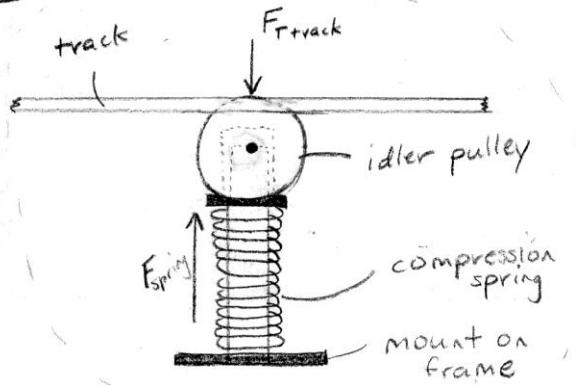


Figure 14: Final Rear Hub Assembly

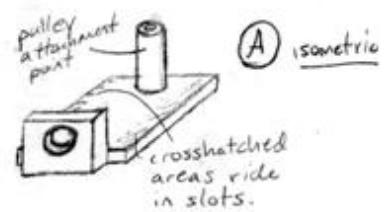
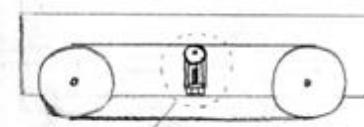
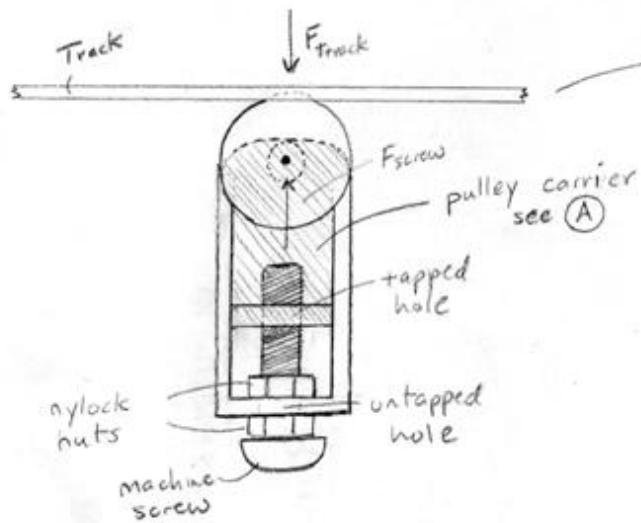
In order for the tracks to function as intended, a tensing device had to be considered. This would ensure the tracks are aligned and taut to prevent them from slipping off of the sprockets. Research was conducted to explore various options for accomplishing this. Concept sketches for these are shown in figure 15.

Concept 1



Full robot side view.

Concept 2



Concept 3

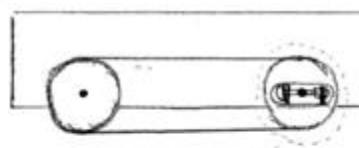
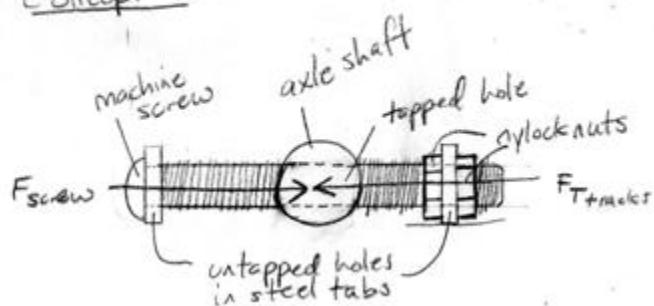


Figure 15: Concept Sketches for Tensing Mechanism

One design in particular stood out as the best solution. Instead of using a separate idler wheel and adding another level of complexity, the front axle could be adjusted to provide tension in the tracks. A fixed axle with threaded holes would ride on two screws. The screws would be able to turn freely on the plates, but when the bolts turn it moves the axle. Turning the screws force the axle forward, thereby forcing the front sprockets to move forward and tightening the tracks. Bushings were used between the fixed axle and the front hubs to minimize power losses from friction. Shaft size calculations and final CAD images are shown below.

Combined loading of front axle calculation:

Equations:

$$\begin{aligned}\sigma_y &= -\frac{M_x y}{I} \\ \sigma_x &= -\frac{M_y x}{I} \\ \tau &= \frac{VQ}{It} \\ \sigma_{avg} &= \left[\frac{\sigma_x + \sigma_y}{2} \right] \\ R &= \sqrt{\left[\frac{\sigma_x - \sigma_y}{2} \right]^2 + \tau^2} \\ \sigma_{max} &= R - \sigma_{avg}\end{aligned}$$

Constants:

$$\begin{aligned}F_N + F_{drop} &= V = \frac{167.1 \text{ lbf}}{4 \text{ shafts}} = 49.25 \text{ lbf} \\ M_x &= 49.25 \text{ lbf} * .5 \text{ in} = 24.625 \text{ lbf} \\ M_y &= 10 \text{ lbf} (\text{force of tension in tracks}) \\ x = y = r &= .125 \text{ in} \\ \sigma_{yield \ steel} &= 58,015 \text{ psi} \text{ (Anzor Stainless Steel Fasteners, 2007)} \\ I &= \frac{\pi}{4} r^4 \\ Q &= \frac{2}{3} (r^3) \\ t &= 2r \\ \sigma_y &= -\frac{6.365}{r^3} \\ R &= 38620.16 \text{ psi} \\ \sigma_x &= -\frac{31.35}{r^3} \\ \tau &= \frac{10.45}{r^2} \\ \sigma_{avg} &= -6430.16 \text{ psi}\end{aligned}$$

$$\sigma_{max} = 45051.11 \text{ psi} < 58015 \text{ psi}$$

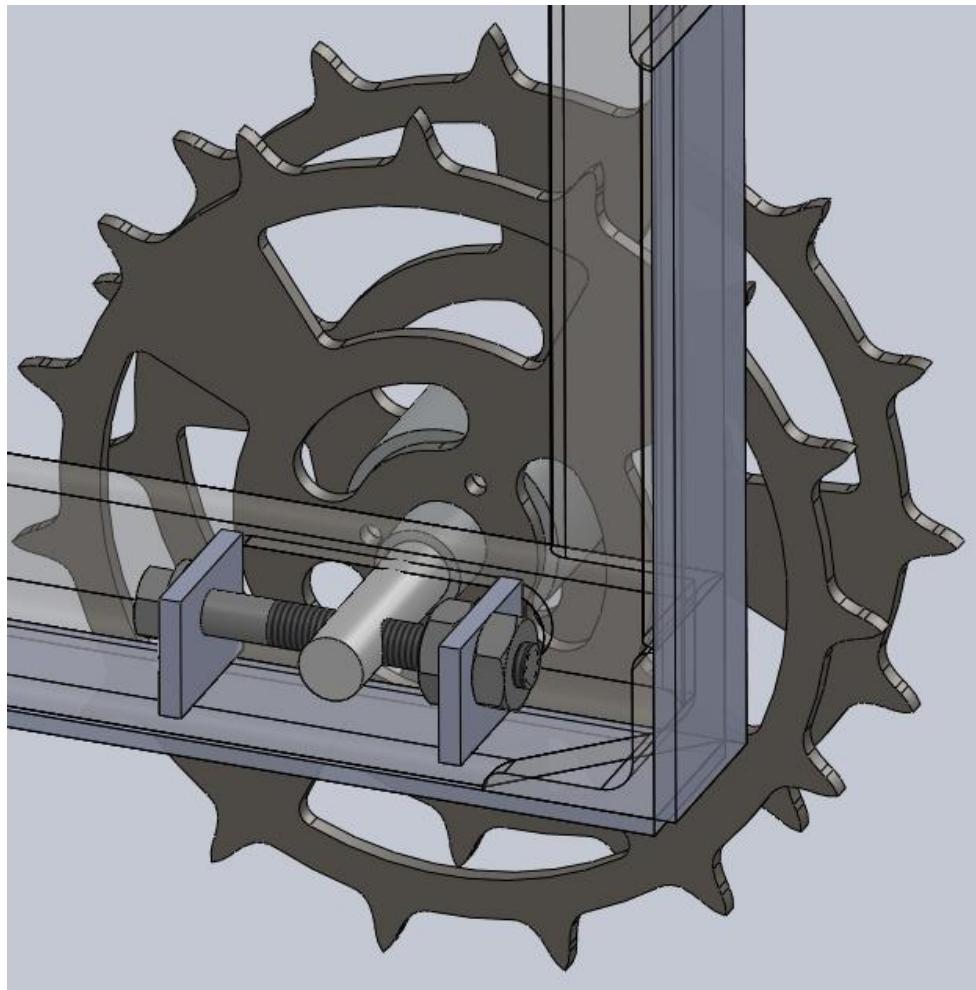


Figure 16: Sectional View of Front Axle Assembly

To make the robot safer for the user and keep the tracks cleaner, covers were designed to shroud the interior of the tracks. These covers were designed to be made out of Lexan and follow the profile of the tracks. They were designed to attach to the rear hub supports and could come off of the vehicle easily with three screws. However, they were never manufactured for the prototype presented.

Results and Significant Analysis

Several things became clear after assembly of the robot; some of which verified the modeling that was done and proved that the components were designed correctly. The shafts were able to support the static robot, though a drop test was never performed. Some problems arose that were not properly accounted for during the design stages. The track system did not function as it was designed to, and the electronics packaging inside the robot was neglected in the design stages. The climbing ability of the robot was not tested because the tracks were not functional and therefore the robot could not move on its own.

One major problem became apparent almost immediately; this was the tracks malfunctioning. The sprockets did not mesh with the chain once the treads were riveted to the chain. Although benchmarking was performed on this solution and it was found to be a functional one, it should have been tested before the design was selected as the final one. However due to tight time

constraints, testing of alternative designs was not feasible, and the group was committed to the chosen track system.

In the future it may be a wise decision to dissuade students from taking on such an intricate problem for their first design project. This was one large error in judgment on the team's part that stopped the robot from functioning the way it was designed to. The decision to go ahead was made based upon a lack of time for the water jet machine, and parts had to be made in order to continue making progress. If the sprockets had not all been made all at once, the robot would not have been finished before the end of the semester. This major design issue inhibited testing the robot's ability to climb a forty-five degree incline.

Another issue that needed to be addressed was the lack of any mounting system for the electronics inside of the robot. All of the electronics made it into CAD, but there was never any solid bracket system for them; so something was built at the last minute. The final build was not ideal, and although it did partially function as intended it was not optimal. Similarly, the camera mount was not rigid because of a lack of any design preparation.

9.4 Power Distribution Circuit

Matthew Williamson

The power distribution subsystem has one major purpose. This system distributes power from the batteries to all the devices onboard the robot that require power. These devices include two H-Bridges (Motor speed controllers) and the drive motors connected to them, as well as the onboard computer and Arduino microcontroller (and the two servo motors attached to the Arduino), webcam and Global Positioning System device that is attached to the computer. For the initial design of the circuit, the voltage and current required by each device to be powered was determined. These are listed in Tables 7 and 8, shown below

Device	Count	Required Voltage (V)
H-Bridge (High Power Side)	2	5-24
H-Bridge (Logic Side)	2	8-24
Computer	1	8-15

Table 7: Voltages Required by Devices on Robot(CompuLab, 2009),

Component	Quantity	Individual Load (A)			Total Load(A)		
		No Load	High Load	Typical Load	None	Peak	Typical
IED drive motors	2	0.16	9.64	3	0.32	19.28	6
IED H-Bridge	2	0.1	0.1	0.1	0.2	0.2	0.2
Arduino	1	0.05	0.05	0.05	0.05	0.05	0.05
HS 422 Servo	2	0.05	0.05	0.05	0.1	0.1	0.1
FitPC2 computer	1	0.5	0.67	0.6	0.5	0.67	0.6
GPS antenna	1	0.065	0.08	0.08	0.065	0.08	0.08
Camera	1	0.8	0.8	0.8	0.8	0.8	0.8
				Total Amps:	2.035	21.18	7.83

Table 8: Current Drawn by Devices on Robot Under Various Loads (CompuLab, 2009),
(Documentation found in appendices K, N and O)

Based on a typical load of 7.83 Amps, and the desired run time of 4 hours minimum, it was calculated that the batteries would need to have a minimum capacity of 31.32 Amp-Hours. With this as a basic requirement, several battery options were explored, including sealed lead-acid, lithium-ion, and nickel-metal hydride.

Battery Type	Cost (\$)	Weight (lb)
Lithium-Ion	398.97	3.9
Nickel-Metal Hydride	284.85	10.5
Sealed Lead-Acid	73.90	18.6

Table 9: Comparison of 30Ah Equivalent Batteries(All-Battery.com, 2009),(BatteryJunction.com, 2006), (BatteryMart.com, 2010)

As shown in table 9, there is a significant tradeoff between cost and weight. Given a large amount of funding, lithium-ion batteries would be the ideal choice. However, as a result of the very small budget this project had, the most inexpensive option was chosen. Two 12 Volt, 14Amp hour, sealed lead-acid batteries were made available at no charge, and given budget constraints; they were used to provide power for this project.

The final step necessary to complete the circuit was a method in which to regulate the voltage flowing to the computer and logic side of the H-Bridge. Since the high-power side of the H-Bridges draw power directly from the batteries, it was concluded that the more sensitive electronics such as the on board computer and H-Bridge logic side, would be better protected from unexpected spikes in voltage, if they were not powered directly from the batteries. Additionally, the two batteries were wired in series providing a maximum voltage drop across the high-power side of the H-Bridge. This provided the motors with their maximum capacity of 24 Volts. However this setup provides a voltage that is greater than the maximum voltage for the computer. To provide the proper voltage to the computer and regulated power to the logic side of the H-Bridges a DC-DC “Buck” converter was used. The particular device chosen, a National Semiconductor LM338K can, given an input voltage between 3 and 35 Volts, produce an adjustable output between 1.25 Volts and whatever the input was. This wide range of output options, low price and high availability lead to the selection of this regulator for this project.

With the DC-DC converter chosen, the last step in creating the circuit was to set up the DC-DC converter to produce an output voltage of 9 Volts. As per manufacturers specifications, given the setup shown in figure 6,

$$V_{out} = V_{ref} \left(1 + \frac{R2}{R1} \right) + I_{adj} (R2)$$

Where V_{ref} is 1.25 Volts, the internal reference voltage of the DC-DC converter, and I_{adj} , the current coming out of the ADJ pin, as shown in figure 6, is insignificant (less than $100\mu A$). Therefore, in order to determine the resistors needed to produce an output voltage of 9V, R2 was assumed to be 1K, I_{adj} was left out due to its relative insignificance, and the equation was solved for R1:

$$9 = 1.25 \left(1 + \frac{1000}{R1} \right)$$

$$R1 = \frac{1250}{7.75} = 161\Omega$$

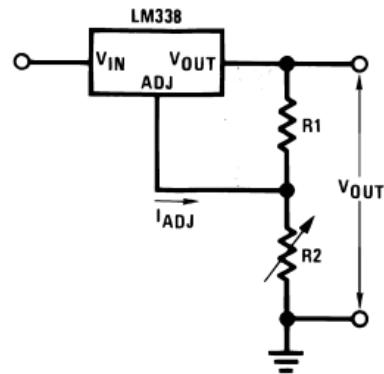


Figure 17: 9V DC-DC Converter Circuit Diagram



Figure 18: DC-DC Converter as Implemented in Final Design

With all of the components to the circuit completed, it was wired together. This resulted in an operational circuit capable of powering each device on the robot. An overall view of this circuit can be seen in figure 19, where the red lines indicate the positive power lines, and the black lines indicate grounds.

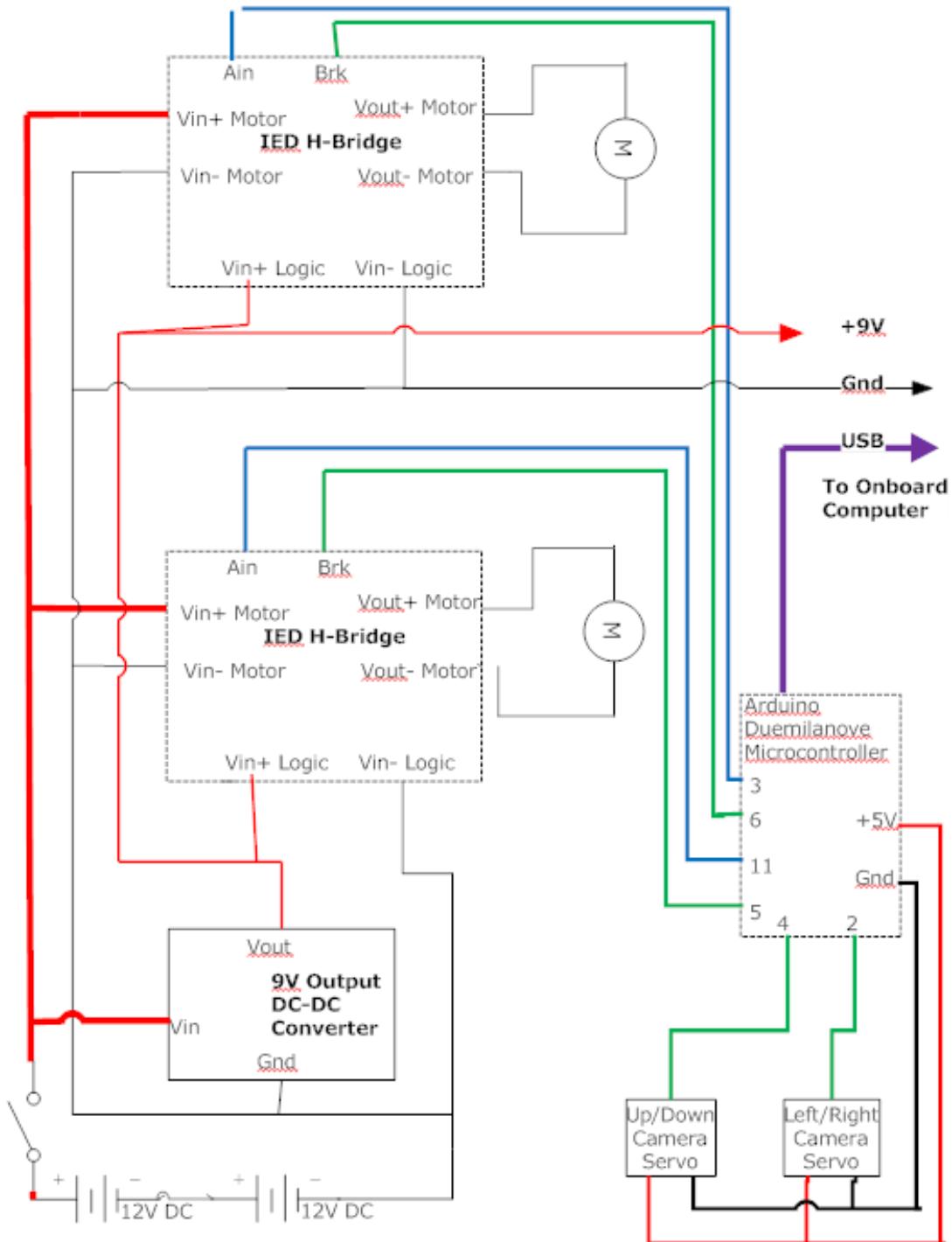


Figure 19: Overall Circuit and Control Diagram. Power and ground signals are red and black, respectively, and control signals are blue, green or purple

9.5 Low Level Control

Matthew Williamson

The low level control subsystem of this robot consists of both the software implementation of control and the devices on board the robot that turn this code into the physical motions of driving the robot and moving the camera. The devices used by this subsystem include the Arduino Microcontroller, the two H-Bridges used for speed control, the RC filter circuit between the Arduino and H-Bridges, and the Servo Motors used for panning and tilting the camera.

Software

The software aspect of the low level control consists of a program running on the Arduino microcontroller, which waits for input from the user before performing any actions. This input is received via the Arduino's USB link to the onboard computer. At its most basic level, the control software waits for a keyboard character (such as "a" or "w") to be sent to the Arduino. Once a character is received, the software determines which action to perform based on the character it received. These actions include driving the robot forward, backward, left or right, or turning the camera up, down, left or right.

To signal the H-Bridges to perform these physical motions, the Arduino produces a Pulsewidth-Modulated, or PWM control signal. This signal is a square wave, with a peak to peak voltage of 5Volts in other words the distance between the peaks on the wave is measured as 5 Volts. When this signal is run through an RC filter, the output becomes a constant voltage representing the average voltage of the square wave. By varying the time period in which the signal is at 5 Volts (known as the duty cycle and can be seen in Figure 20) the average voltage will differ.

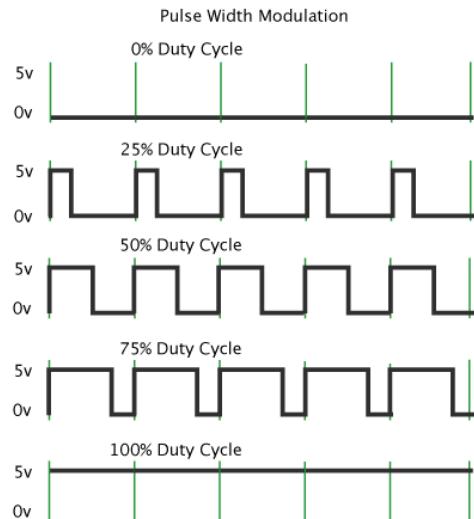


Figure 20: PWM Signal Examples(Hirzel, 2008)

For the H-Bridges used, a signal of 0 Volts (corresponding to a duty cycle of 0%) will power the attached motor at full speed in one direction. A signal of 5V (corresponding to a duty cycle of 100%) will power the attached motor at full speed in the opposite direction, and a signal of 2.5V (corresponding to a duty cycle of 50%) will put the motors in the neutral state. The servo motors used for panning and tilting the camera operate in a similar fashion, but instead of motor speeds, the various duty cycles correspond to servo positions.

RC Filter

The RC filter, as described above, converts the square wave output from the Arduino into a constant voltage to be read by the H-Bridge. The design of this circuit is very straightforward, and can be seen in Figure 21.

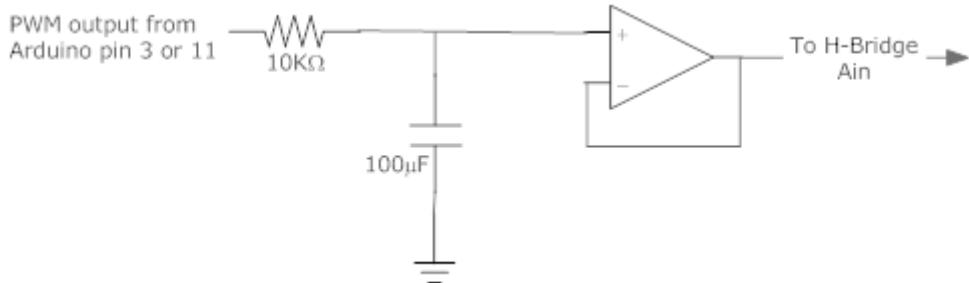


Figure 21: RC Filter Circuit

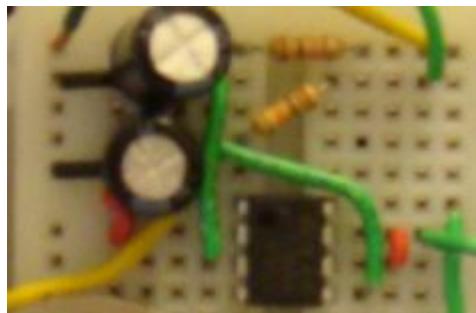


Figure 22: RC Filter Circuit as Implemented in Final Design

This circuit was designed initially to provide a quick response to changing PWM outputs from the Arduino, while still providing an accurate average of the PWM output. To do this, the RC time constant, a value that represents the time it takes the capacitor to charge to approximately 63% of its full capacity (RC Time Constant, 2010), was calculated. This value was initially required to be at least 10 times larger than the period of the PWM output (the time it takes for the square waveform that can be seen in figure 21 to repeat itself), but no larger than 3 seconds long, to ensure fast response of the motors to changing inputs.

Period Calculation:
Arduino PWM Frequency = 480Hz

$$Period = \frac{1}{Frequency}$$
$$Period = \frac{1}{480} = 0.00208 \text{ seconds}$$

Initially, it was determined that a 0.1 second time constant would provide a steady voltage, but also a smooth response. This was determined by multiplying the period by 50, which results in the average voltage being taken over 50 PWM cycles, giving a very accurate reading of the average.

RC Time Constant (τ) Calculation:
 $\tau = R * C$
Assume a 100μF Capacitor

$$\frac{\tau}{C} = R = \frac{0.1}{.0001} = 1000\Omega$$

Using the values calculated, the control circuit was tested to verify the response of the motors to changing input signals. The testing was successful, and the circuit performed as expected. The motors responded to changes in input signals and reversed direction in under one second. During this testing, however, it was determined that a response time of under 1 second would be detrimental to the sprockets and other mechanical aspects of the drive train. With the motors reversing direction very quickly, large torque forces would have been placed on the shafts of the motors, and the screws attaching the motor shafts to the sprockets. In order to slow down the response of the motors, the RC time constant needed to be increased. This was done by increasing the resistance until the desired performance was achieved, which was determined to be that the motors took approximately three seconds to slow down and stop, or six seconds to slow down, stop, and change direction fully. This performance was achieved with a resistance of 10KΩ.

The final component to the RC Filter circuit was an operational amplifier (op-amp). This component acts as a buffer between the filter circuit and the circuit receiving the filtered signal. Without this component, the resistance and other characteristics of the secondary circuit would affect the RC filter circuit, complicating the RC time constant calculations, and introducing additional variability to the output of the filter.

H-Bridges and Motors

The two H-Bridges used in this robot are used to control the speed and direction of the two drive motors. The main purpose of using this device is to allow a low-power control signal, such as the one output by the Arduino to control a considerably higher power motor, such as the IED motors used in this project. This task can also be performed by a device known as a speed controller. For this project, the H-Bridge was chosen for its low cost (it was free to borrow for this project) and the fact that it was recommended and designed for use with the IED motors. The IED motors were likewise chosen for their low cost, as they were also free for use with this project.

The function of the H-Bridges is very straightforward. The H-Bridge receives a signal from zero to five volts on its “Ain” port. Using a small controller on the H-Bridge, this signal is then interpreted to control the motor attached to the H-Bridge. As previously stated, a signal of 0V turns the motor at full speed in one direction, 5V turns the motor at full speed in the other direction, and 2.5V stops the motor. Signals in between these values turn the motor at intermediate speeds. For more information on the H-Bridge, see appendix O.

Arduino

The Arduino microcontroller, shown in figure 23 is used to produce control signals for the servo motors and H-Bridges. This device runs the software described above, and delivers the output specified by the software to the devices via the output pins. Two options were considered for the microcontroller, the Arduino and the SiLabs C8051. Again, cost became the leading factor in the decision to choose the Arduino. An Arduino was available for use at no charge, because a group member already owned one. Additionally, the Arduino was determined to be much easier to program. It is very well-documented online, and there were many functions available in the programming environment to perform actions such as moving servos, that are not available on the C8051. Overall, the Arduino was chosen due to its simplicity and cost.

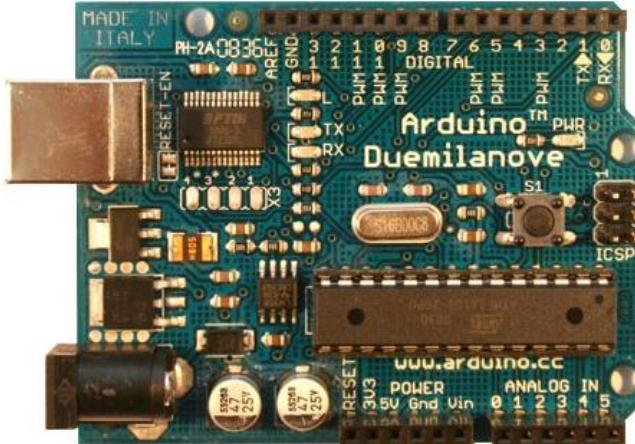


Figure 23: The Arduino Microcontroller (Arduino, 2010)

Servo Motors

The HS-422 servo motors shown in figure 24 and used in this project are used to pan and tilt the onboard web camera, so the user can gain a greater awareness of the environment around the robot. In similar fashion to the batteries and Arduino, the servos were chosen for their low cost and availability. A group member also owned the servos, which eliminated the cost of purchasing the servos. Additionally, a group member had already used the servos in the exact fashion in which they were used in this project (to pan and tilt a web camera), proving that they would effectively pan and tilt the camera.



Figure 24: HS-422 Servo

Overall Control System:

The control system overall performed well. The Arduino proved to be an excellent device to send control signals, and the servos used to pan and tilt the camera performed flawlessly. The H-Bridges used to control the motors proved to be problematic. They were prone to breaking, for reasons not understood given the limited knowledge of electric circuits possessed by the group. A speed controller, such as those used in RC cars would have been a better choice. Despite issues with the H-Bridge, the control system was successfully demonstrated, with all functions working as specified.

9.6 Computer and Wireless Web Server

Lucas Doyle

Why use a Computer?

One of the goals of the Border Patrol Robot is to supply intuitive controls to the customer when using the robot. More specifically, the goal is to have the customer be able to effectively operate the robot with less than 10 minutes of training. In order to achieve this, from the moment the robot turns on many devices need to “just work” without the customer having to do anything technical or specialized. This means that a software platform has to be put in place that automatically takes care of these technical tasks so the customer doesn’t have to.

To this end, it was determined that the optimal way to solve this design problem was to include a computer on board the robot to act as the centralized controller for the robot. This decision was made for several reasons:

- They are familiar and flexible pieces of technology
- Computers have a high degree of connectivity and compatibility:
 - Many important peripheral devices (e.g. the camera) can be directly controlled through USB
 - Wireless internet (also known as Wi-Fi) simplifies communication between user and robot
- In combination with Wi-Fi, a web server running on the robot's computer will allow for control from any web browser
- Running a computer allows full control over all aspects of the robot's operating software
- Small, power efficient industrial PC's are available and ideal for this application

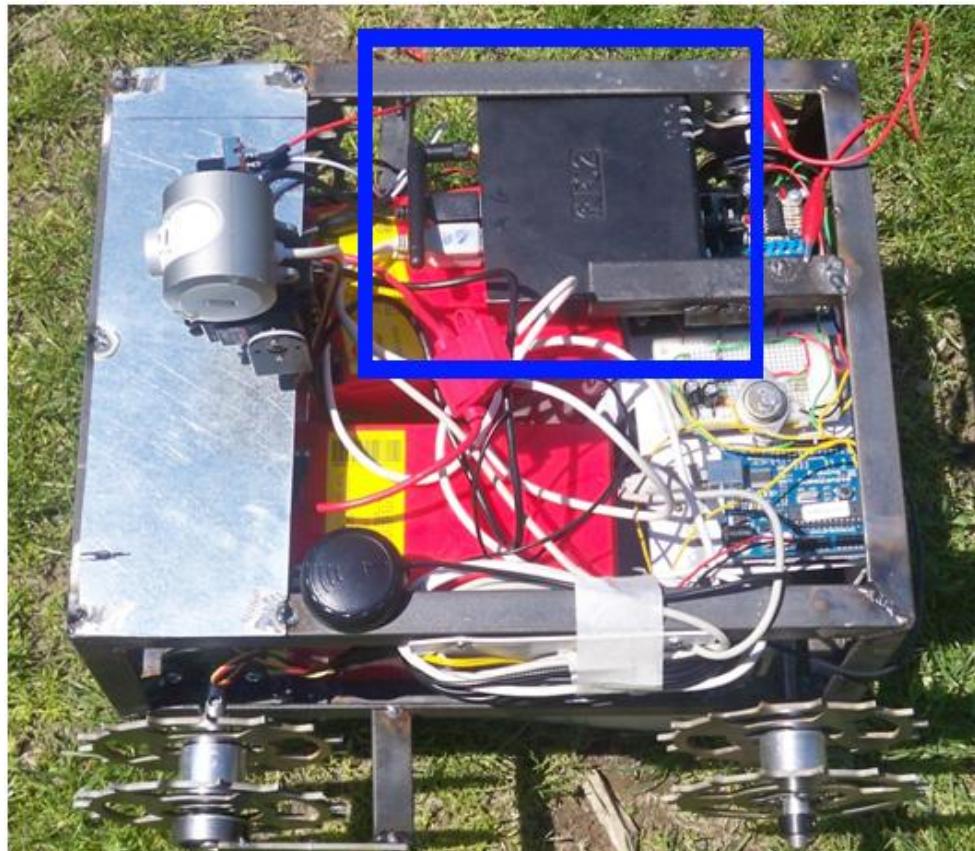


Figure 25: View of Computer as Installed on Robot

Choosing a Computer

A CompuLab fit-PC2i industrial PC was chosen to be the robot's computer. It was chosen because of its physical and electrical properties, processing power, connectivity and availability. First, the physical size of the computer is 4" x 4.5" x 1.05" and its weight is 13 ounces, making it about half the size (but slightly heavier) than an average paperback. This is well within what the frame can handle as far as physically fitting the computer on the robot.

The electrical properties – namely the power efficiency, voltage and current the computer runs at – are all within the ideal design constraints of the power distribution circuit. The computer consumes 8 watts of power at high loading of the Central Processing Unit (CPU) and 6 watts when load is low. To put this in perspective, the Lenovo t61p (the standard issue RPI laptop for the class of 2012) consumes 90 watts of power on average. This minimal power requirement is due in part to the minimal current draw of the fit-PC2i. It never draws more than 1.5 amps of current, meaning that in combination with the 24 amp-hour batteries on the robot, the computer could operate for a theoretical maximum of 16 hours. Last, the input voltage to power the computer can be anywhere from 8-15 V, ensuring that the 9 V power circuit on the robot will power the computer, as well as leaving a factor of safety in the system if the voltage unexpectedly drops or surges.

The actual computing power of the fit-PC, namely the processor, Random Access Memory (RAM) and hard disk space, also fit the demands of the border patrol robot. The computer's running capability in fact is identical to the components found in many modern day netbooks. See the manufacturer's website (<http://www.fit-pc.com/web/fit-pc2/fit-pc2i-specifications/>) for exact component documentation. Additionally, the graphics adapter allowed the computer to be plugged into an external monitor when it was being initially developed and debugged, which was immensely convenient from a developer's point of view.

The connectivity of the computer played an important role in choosing it as well. For one, the computer features four (Universal Serial Bus) USB ports, which are conveniently used to interface with the Arduino microcontroller, GPS board and webcam. These subsystems are detailed in their own sections, but suffice to say that they all interface through USB. Additionally, the wireless connectivity of the computer is arguably the most important feature of the machine. Without this, the robot would not be able to operate wirelessly and be controlled through the internet.

Finally, the availability of the computer was too good to pass up. One of the team members already owned one of the fit PC's, so there would be zero expense on behalf of the group. The computer was so ideally suited to the application of the border patrol robot that the decision was made to go ahead and use it.

The Linux Operating System

The computer is running Linux as its operating system. Just like Windows 7 or Mac OS X, Linux is in charge of providing an interface between applications and the computer's various low level hardware (processor, RAM, hard drive, etc). This piece of software, the core of the operating system (or any operating system) is known as the kernel. The Linux kernel is unique in that it is known for being extremely stable, efficient, and free.

Surprisingly, not many people outside of programmers, scientists and engineers have heard of or used Linux; however it is used by billions of people every day. For instance, most modern smartphone operating systems are built on top of Linux. The Apple iphone OS, Google's Android and Palm's webOS are all built on top of Linux. Additionally, it is not uncommon for various modern appliances, including set cable boxes, car computers, and wireless routers to be running some form of Linux. The most widespread application of Linux is the internet itself. An overwhelming portion of the internet's web servers are running on top of Linux (as is the robot's

web server). Indeed, the kernel is ideally suited to run a web server, which this project crucially needed to operate.

While Linux has traditionally been difficult to use and configure due to its complexity, over the years it has grown more consumer friendly. Linux has done this in the form of hugely popular Linux distributions used on many PC's all over the world. The most popular of such distributions, known as Ubuntu (specifically Ubuntu 9.10), was chosen to be installed on the computer due to its inherent user friendliness.

To sum up, Linux is a good choice of an operating system for the robot because it is free, stable, efficient, widespread, easy to install and configure, and ideally suited to run a web server.

Wireless Communication

As mentioned earlier, the wireless connectivity of the computer in combination with a web server running on top of Linux makes for a very robust platform to for the user to operate the robot. Wireless networks are ubiquitous nowadays, and is most likely something that the customer already has prior experience setting up and using in their home or workplace. Since the robot was prototyped on the RPI campus, it currently makes use of the on campus Wi-Fi network known as 'rpi_802.1x'.

The robot's computer has been configured to automatically connect to this wireless network and establish a domain name. The name "stonebot.ath.cx" was arbitrarily chosen, but this name could be anything that the customer requests. For detailed instructions on how to configure the computer to automatically connect to a Wi-Fi network when powered on, see appendix ???.

Using wireless internet as a communication medium has many advantages. First, it enables the customer to operate the robot from a computer anywhere in the world so long as they have an internet connection. This effectively gives the customer the freedom to be anywhere they want when using the robot, whereas if some type of radio transmitter / receiver had been chosen, the operator would have to be within a certain radius to control the robot. Second, since all that is required to operate the robot is a modern web browser, the customer does not have to install any specialized software to use the product. Additionally, by utilizing web based user interface (UI), functionality can easily be increased to include any mobile internet device (iPods, blackberries, etc) instead of just computers connected to the web.

There are disadvantages to using web based controls. First, when in operation, the robot uses a considerable amount of bandwidth (about 600kbps). This means that a customer with a slow or unreliable connection will be unable to properly operate the robot. For instance, if a customer was using dial up for some reason, they would only be able to receive information at 56kbps, making the robot wholly unusable. Additionally, operation of the robot is limited to where there is Wi-Fi. If for some reason the robot loses connection to the internet (e.g. drives out of Wi-Fi range) it is effectively stranded and must be manually retrieved. Fortunately, with the advent of 3 G networks for smart phones all over the country, this problem could easily be solved with the inclusion of a USB 3 G adapter on the computer. This would give the robot a permanent and fast wireless internet connection.

The Web server and Website

The web server provides a foundation that the website is built upon. As hinted at by the customer requirements, the website is the gateway for providing intuitive control of the robot to the user. In Ubuntu, installing and configuring a web server and associated programs is a one line command from the terminal. This one command actually installs several important programs. See appendix K for details about this installation these programs.

Normally, these programs require a significant amount of configuration in order to work properly with one another. Fortunately, by installing them all at once in Ubuntu these settings and configuration files are automatically set, which saved the programmer an immense amount of time.

PHP

An overwhelming majority of the robot's operating code is written in PHP. As mentioned in the bullet point above, PHP is widely used by many websites on the internet to generate dynamic content. Over the years, many libraries (called classes) have been written for PHP to expand its functionality to other areas as well. For example, the robot is using a library called PHPserial to communicate with serial ports (e.g. the Arduino and GPS, detailed in other sections of subsystem development).

Briefly, the PHP works by embedding PHP code into the HTML source (HTML is beyond the scope of this section. For more information on HTML, see appendix???) of a web page. This then gets passed through an interpreter every time the page is loaded by a user. This interpreter looks for the PHP code and evaluates it before sending the result to the user through the Apache web server. When PHP is evaluated, it can trigger actions and potentially change the look or content of pages based on user input, other files or even coordinates received from a GPS. It is in this way that PHP enables the creation of dynamic websites with the level of functionality required to make the robot usable by the customer.

Website Architecture

The websites architecture is set up in a specific way to allow the dynamic loading of specific modules that pertain to different aspects of the robot's functionality. It is also written in such a manner that all common elements of the websites structure and appearance are contained in single files. Using PHP it is possible to link these files together such that when one file is changed, the effect on the website is global. This makes modifying, maintaining and adding content easy for future programmers and upgrades made to the robot.

The different modules are written such that they can easily be included into any web page on the robot. For instance, the majority of the website was written by establishing a page template and, in most cases, changing one line of code. This modularity and separation of style from content makes the construction of the web pages easy to understand and easy for future programmers to expand the functionality.

Website appearance and JavaScript

While not strictly a requirement, it was important to the group that the website be as aesthetically pleasing as possible. When customers are potentially going to use the robot for hours at a time, it does not hurt for the interface to be easy on the eyes. Therefore, the decision to use the free jQuery JavaScript User Interface library was made. jQuery is a free and open set of JavaScript tools written specifically for writing user interfaces. These allowed a quick, easy and free way of making many aspects of the website animated, colorful and effective as well as provide a global and consistent icon set throughout.

JavaScript is a client-side language (meaning that processing is done on the customer's computer). Contrast this to PHP, which is a server-side language -- meaning all the computation is done on the server before the page is sent to the client. Because of JavaScript's client side nature, JavaScript code can be processed at intervals according to the user's actions after the web page has loaded. Function calls to the jQuery library had to be written in JavaScript to define the behavior of buttons, pop-up windows, menus etc according to what the user does once the page has finished loading. For instance, the generic button on the robot's website needs to be black by default, orange when the customer's mouse is hovering over and blue when the button is clicked.

All the jQuery UI behaviors written in JavaScript were inserted into a file that could easily be included at will on any web page by PHP. Furthermore, PHP is also used to dynamically generate JavaScript to create pages with content that continuously changes. This interaction of PHP and JavaScript form the basis of what is called AJAX. This stands for asynchronous JavaScript and XML (XML being the extensible markup language); however it generally refers to the technique of mixing client-side and server-side languages to form dynamic and interactive web experience. While not used in any part of the jQuery UI, this technique is crucial to the maps subsystem of the robot and is therefore worth mentioning here.

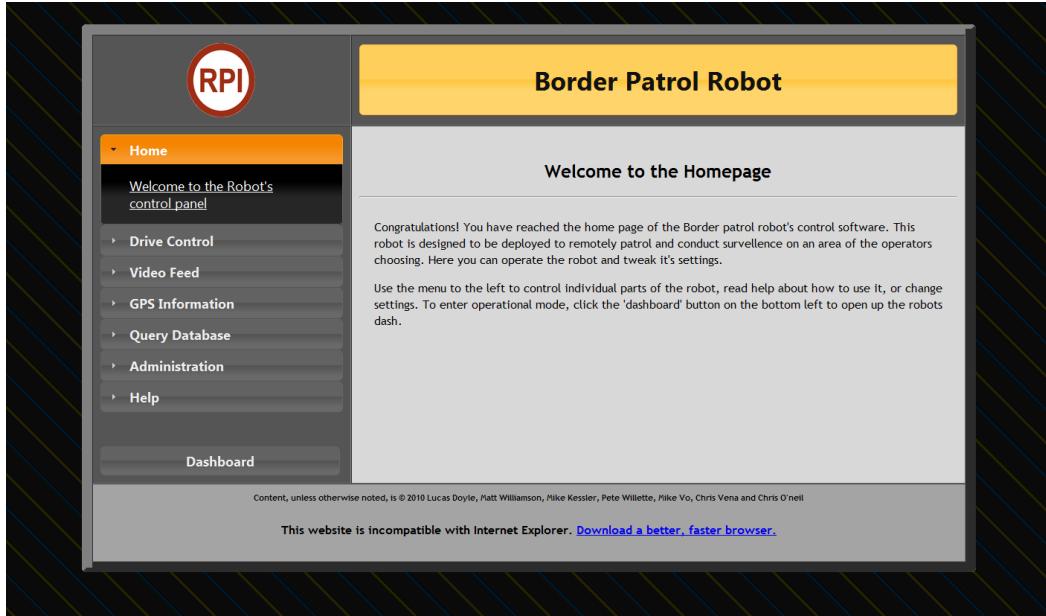


Figure 26: General Appearance of Website

9.7 Global Positioning System

Lucas Doyle

Why use a GPS?

The Global Positioning System (GPS) is what makes the robot aware of its current location. This subsystem is critical part of fulfilling the basic requirements first outlined by the original project description. Those objectives are:

- Record position data as gathered via the GPS, optionally with time stamps
- Transfer / transmit the position data to a separate “remote” location
- Replay & analyze position data at the “remote” location
- Utilize & integrate the GPS / computer system with a remote controlled mobile platform

The border patrol robot accomplishes all of these tasks with the aid of the on board GPS. The GPS itself is a circuit board which attaches to an antenna for satellite reception as well as a USB port for serial communication with the computer. It draws all power from USB and is a self contained microprocessor that automatically connects to satellites in outer space, triangulates its position and outputs its position to the serial port. The transmission of coordinates themselves is formatted in what are called NMEA sentences.

NMEA Sentences and Parser

NMEA sentences or more specifically NMEA 0183 is a specification for a protocol developed by the National Marine Electronics Association. It is a protocol that is used for communication with various marine electronics such as GPS, sonar, electronic compass, gyro etc. While the particular GPS board on the robot is able to output all kinds of information, such as time, velocity, number of connected satellites, etc, the only thing that was required were the latitude and longitude coordinates of the GPS. To this end, several programs, including a NMEA parser, were written to get these coordinates and translate them into something able to be used by the mapping software (see the maps subsystem) or saved in the database for later retrieval.

The core of the GPS subsystem is the NMEA parser. In the context of computer science, a parser is a program that divides text into functional, individual components. This piece of code, written in PHP actually extracts coordinates from the raw output of the GPS in NMEA format. For example, here is a sample NMEA sentence containing the information required from the robot:

\$GPRMC,040302.663,A,3939.7,N,10506.6,W,0.27,358.86,200804,*1A

Commas separate information in the sentences. The parser looks for information between these commas. To see a step by step breakdown of the contents of the sentence that are extracted by the parser that are important to the robot see appendix L.

Together, coordinates represent a point on the map that is able to be readily used by the map portion of the robot's operating code. Most information in the sentence is not used by the parser, although its functionality could be easily increased to use this to include information such as the robot's heading and speed.

Some general remarks about the performance of the parser: NMEA parsers (most any parser for that matter) are usually written in system level languages like C or C++ as opposed to PHP which is a high level language. The reason for this is because system level languages are compiled into programs that are directly executable by the computer. Because of this, they are able to efficiently and synchronously communicate to the serial port of the GPS (itself a system level device).

In contrast, PHP is a high level language that instead of being compiled is run through an interpreter. Because of this PHP is much more inefficient at communicating with system level devices. Additionally, communication needs to be done asynchronously. For this reason, a parser implemented in PHP was not as efficient as it could have been had it been written in C/C++. The techniques used to write the parser also required advanced knowledge of PHP, potentially making it harder for other programmers to modify or troubleshoot.

Despite the aforementioned downsides to a PHP parser, it was implemented and the payoff was worth it. Writing an external NMEA parser in a system level language such as C or C++ would have meant handling complex communication between the external parser and the rest of the robot's software written in PHP. By keeping everything written in PHP, the parser can be used in line with the rest of the robot's operating code.

Choosing a GPS

The decision over which particular GPS board to be used by the robot was trivial. All GPS evaluation boards on the market are targeted at the hobby electronics enthusiasts do the same basic function. While subtly different (some had better reception, some connected to satellites faster, some were different sizes, etc), they all did what was required by the robot (to output NMEA sentences). This is all that was required for the GPS. Additionally, an evaluation board was generously supplied to the group by Lockheed Martin.

9.8 Maps

Lucas Doyle

Why are maps necessary?

The GPS can effectively acquire coordinates; however these are useless to the customer unless they can be somehow visualized. The decision was made to go with Google Maps to visualize the data because the service is free, fast, well documented and easy to use from a programming perspective. The way the service is interacted with is through JavaScript (for more information on JavaScript, see website appearance) calls to the Google Maps Application Programming Interface (API). An API is a gateway provided to the public that allows other programmers to use parts of another application or service in their own application. In this case, Google is providing an API that the robot is using to generate maps. The Google Maps API allows the robot to effectively deliver maps to the user. In order to be able to use the API, a unique key in the form of a long string of pseudo-random text was issued to the robot in order to properly authenticate itself to Google's systems before maps could be generated. Maps are used in two portions of the project: to display the location of the robot on a map in real-time, as well as in the search feature.

Real time positioning

In combination with the GPS NMEA parser (detailed in the GPS subsystem) and a special set of indicators and controls, a map with a single point in the center representing the robots current location was created. This is important because without the real-time map, the customer would be unable to determine the current location of the robot. This page first takes coordinates from the GPS parser, and then using JavaScript calls to the Google Maps API which creates a map of a certain size and puts a marker in the center of the map corresponding to the robots location. The page is set to reload every three seconds, making the marker change position as the robot moves. If the GPS does not have a fix on any satellites, it outputs a message to the user and hides the controls to the GPS. The controls consist of several indicators displaying the current latitude, longitude, time and date as well as a button that the customer clicks when an illegal immigrant is spotted. When the button is clicked, the information in the indicators is added to the end of the database to be searched through later.

This technique works effectively for the application, however since the page reloads every three seconds. It needs to be embedded in what is called an iframe in order to prevent the entire website from refreshing every three seconds. An iframe is an HTML element that functions as a “window” to another web page. This allows web pages to be embedded in web pages. In most scenarios, iframes are bad because they force the web browser to load two web pages at once, which consumes unnecessary bandwidth. This is bad because it puts more load on the robot's web server while requiring the customer to have a slightly stronger internet connection. A better approach would have been to go pure AJAX, as this would have used less bandwidth however it would have been more complex programming.

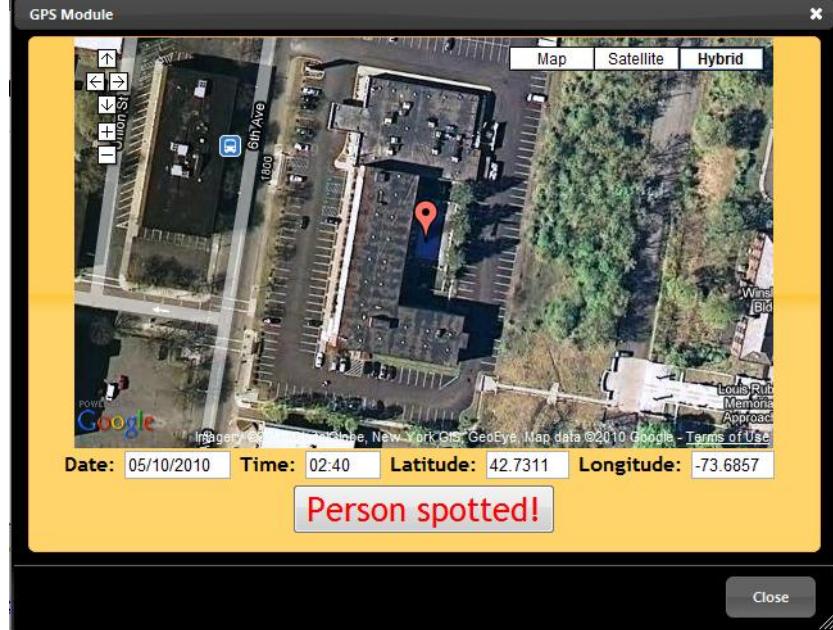


Figure 27: The Real-time GPS Control

Search feature

The robot needed the capability to search through all the data aggregated by the GPS and display this to the customer. This allows the customer to visualize where immigrants have previously been spotted by the robot, which can enable the customer to track trends and possibly predict future breaks in the border's security.

This is done through the use of a database, PHP, JavaScript and managing calls to the Google Maps API. The user starts out by navigating to the search page on the robot and selecting a start and an end date. After the user fills out a form with the desired start and end dates, they click the submit button. PHP then keeps track of the search parameters as the results page is loaded. The end result is a map with all the points that fall within the dates indicated by the user. The user can additionally click on the map and see at what date and time the illegal immigrant was spotted.

Performance of this could have been better. The search form and the resultant map need to be two different pages. As a result of this, both are used inside of iframes from the main search page so that the user does not have to reload the whole page when making a search. See the last paragraph in the section of real-time control for more information about why iframes are generally frowned upon.

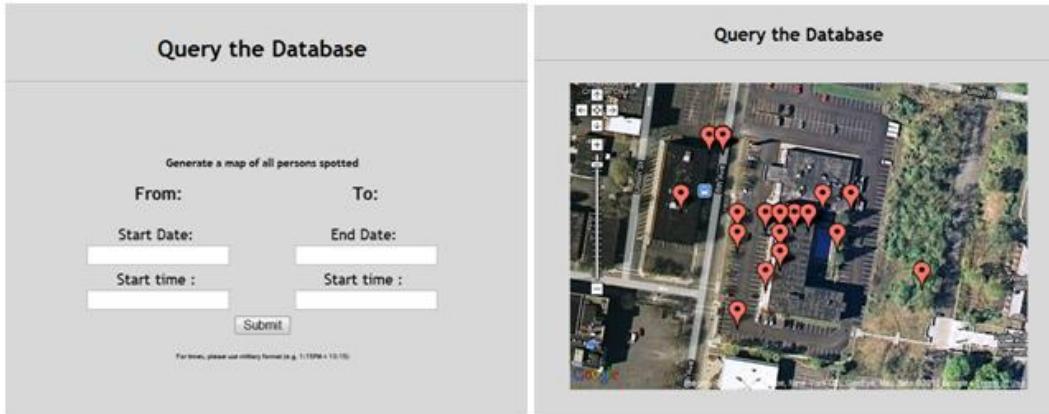


Figure 28: GPS Search Form (Before and After Search)

9.9 Camera

Lucas Doyle

Why a Camera?

For the majority of its operation, the robot is being controlled out of line of sight. In order for the robot to be able to navigate its environment and furthermore be able to identify illegal immigrants, it needs a camera to survey its surroundings. Since a computer was already within the robot, it made sense to go with a computer based solution as opposed to a TV or analog based solution. Therefore, a standard off the shelf webcam was chosen as the actual apparatus to take video from. To expand its field of vision, the webcam was placed on top of a pan / tilt module that allowed the camera to move around at the users request from the website.

Streaming an image

The major challenge faced then was not the camera's hardware, but its software. The camera needed to be able to be viewable on the website, have decent frame rate (less than 15 frames per second), sufficient picture quality that would enable the identification of illegal immigrants from 100 yards, not consume so much CPU that the computer is overloaded and be generally "usable" by the customer. As the team considered more and more solutions to these requirements, it seemed as if they were not going to be met. However, a solution that met all these requirements was attained with no downsides.

A free program called MJPG-streamer for Linux was installed that essentially streams snapshots of the webcam taken over and over to an image tag on a web page that specifies the streamer as its source. This solution, though not well advertised, is used by many standalone commercial internet enabled webcams. A table with all of the technical details of the options tried by the team can be found in appendix J.

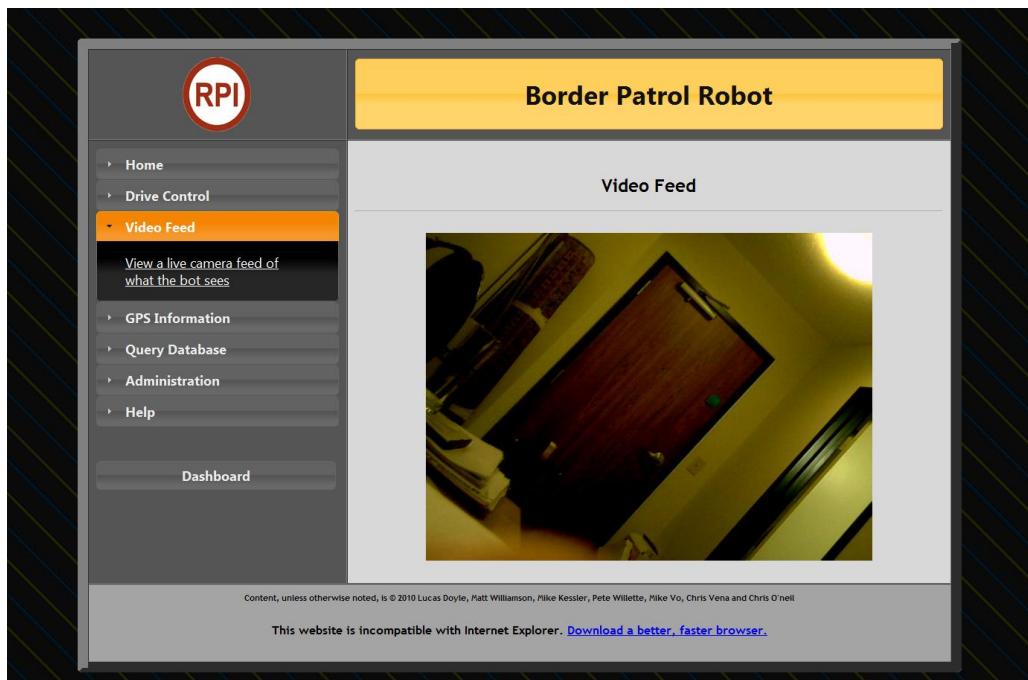


Figure 29: View of the Camera on Website

9.10 High Level Web Control

Lucas Doyle

What it means and why it is needed

High level control refers generally to the interaction of the website with the lower level subsystems of the robot. This is where all the PHP modules and serial interaction with the GPS and Arduino as well as the webcam are synthesized into one coherent package that is intuitively and easily usable by the user to carry out the robot's primary job: to identify illegal immigrants.

Serial communication

The interface of the high level components of the website written in PHP with the low level devices plugged into the USB ports (with the exception of the webcam) of the robot is managed through sending and receiving characters through the serial protocol. Serial is an old standard of communication between devices that transmits and receives data one bit at a time. It is an old standard that has been around as long as computers have, and is therefore a problem that has been solved many times by programmers.

Serial communication for PHP is made possible by the PHP serial class made available on phpclasses.org for free by Rémy Sanchez. This class allows PHP to open up a connection to a serial device, read and write characters and close the device. For example, when the user clicks on the control panel to move the camera up, the character 'w' is sent to the Arduino by the PHP serial class. The Arduino software receives the 'w' and then knows to move the appropriate servo slightly up. The GPS parser also needs to use the serial class, except instead of writing to the device it read coordinates.

Going with the PHP serial class was quick, free, and easy to use (only a few lines of code) by the programmer. Writing something similar to the PHP serial class would have been possible; however there is no reason to needlessly reinvent the wheel.

Putting it all together

The dashboard of the robot is the culmination of most of the modules of the robot and is what the customer uses to carry out the primary function of the robot. JQuery was used to create two popup windows that the user can interact with. One window houses the camera feed and the controls to the drive motors and pan / tilt mechanism while the other shows a real-time map with the robots position as well as the GPS controls. When the user spots an illegal immigrant in the camera feed while operating the robot, they can click the button on the GPS controls that saves the location of the robot in the database. The database can then later be searched by the user to discover trends and potential breaches in border security.

High Level Flowchart

Arrows represent the flow of control

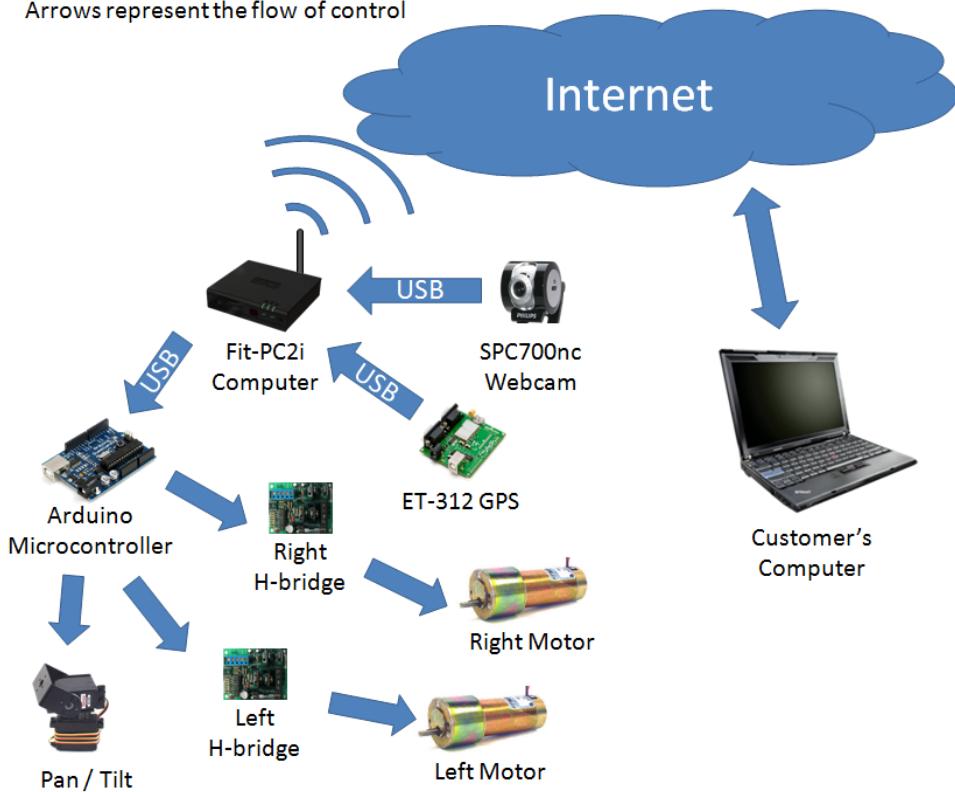


Figure 30: Flowchart Displaying Flow of Control Signals through the Robot

10 Results and Discussion

10.1 Significant Accomplishments

Group

Although the robot is not a complete success, the group still made several significant accomplishments on the project. Multiple subsystems such as the camera, GPS, motors, drive train, and chassis were successful according to the customer requirements. The group learned a lot from the following mistakes that were made in the design process, mechanical assembly, and the electrical components.

The mechanical portion of the project was planned out over the weeks before assembly. CAD drawings were made which gave the group a general idea of how the robot would be assembled. Despite this planning, everything was not accounted for. The design for mounting the electrical components were not properly accounted, which could have been easily avoided during the design process. Another problem that was encountered was the issue with the tracks. The manner in which the rivets were inserted into the bike chain caused every other rivet to come into contact with a tooth on the sprocket. With proper planning and calculations these mistakes could have been avoided.

If there is one thing this group has gotten out of this project, it is that nothing replaces proper planning. Time must be taken in order to make sure that everything is taken into account and nothing is left up to chance or put off until the assembly. A lot was learned about the design process, and the group is now better prepared for future projects because of this assignment.

10.2 Results and Conclusions

Chris O'Neil

Much has been learned about the design process from working on this project. Meeting the customer specifications was the most important part of the design process. Many of the customer requirements were met by the team. Intuitive control was achieved by a web server running on an onboard computer to create a user-friendly website interface. Traversal of varied conditions was met by creating a track system. Distinguishing a person at 100 yards was achieved by mounting a camera on the top the robot. Weather resistance would have been achieved by covering the robot chassis with protective Lexan and covering the camera with an Acrylic dome. The ability to be transported in a pickup truck was accomplished by limiting the weight of the robot to around 30 pounds, constraining the dimensions of the robot to a maximum of six feet long by four feet wide, and by utilizing a robust design. The ability to determine location of the robot was achieved by having the GPS coordinates sent to the onboard computer, which sent this signal the website to be plotted on Google Maps.

Four customer requirements were excluded because they were not feasible. The operation of the robot in an inverted position was not achieved because of a lack of components needed to rotate the camera, which would allow it to see upside down, and because the software required to invert the image was too complex. Making the robot more resistant and defensible to human attacks was another requirement excluded. This could have been achieved by building the robot with more expensive and durable materials to withstand greater impact, or by implementing a weapon system such as a taser or more lethal device. Both ways to achieve this were beyond the scope of the project and the implementation of a weapon system would not be looked highly upon because this is a school project. The last two requirements that were excluded were optical zoom for very long range camera capabilities of the robot and the ability to triangulate a humans' actual position. Both could have been achieved by upgrading the robot with much more expensive components and more advanced software. However this was far beyond the scope of the project and the knowledge of the group, respectively.

While many customer requirements were achieved in design, a couple of problems occurred during the assembly process which hindered the progress of the project. First, there was a long waiting period in order to get parts cut on the water jet machine. There was an error in the track system because the sprockets were designed for chains with no rivets, therefore when the rivets were added it caused a change in the geometry of the chain causing every third spoke to hit a rivet. Make-shift wheels were the only reasonable way to counter this problem with the time available. Another problem experienced was with the H-bridges that were provided in the design laboratory. They were very unreliable and very prone to shorting-out. Also, it was not clearly defined how to mount the electronics within the chassis which led to last minute designs for doing so. Finally, time was the most important factor in hindering the weather-proofing of the robot. An acrylic dome for covering the camera was ordered but never arrived and the lexan panels to cover the robot were not received on time. Finally, initial testing showed that the IR LED's did not provide enough illumination to make the camera usable in nighttime applications, and were thus removed from the final design.

Next steps to be taken would be to redesign the track system, design a better way for mounting all of the electronics and add the weather-resistant Lexan parts. If the project was to be continued by this group or the next, a definite improvement would be to use a speed controller instead of the H-bridges. Also, the lead acid batteries accounted for the majority of the robot's weight and looking into the use of lithium ion batteries would be a good idea to improve the design.

Also, a better night-vision system should be implemented, such as those used in night-vision goggles.

Overall, this project was very much a learning experience for all the group members. Many mistakes were made in the design process, and these only served to demonstrate the importance of following this process. Additionally, the importance of beginning the prototyping and testing process as early in the overall design process as possible was emphasized by the failures in this project. Finally, this project demonstrated the difficulties created by the many conflicting time commitments of the group members. Without all the group members present at every meeting, design decisions were more difficult to make, and valuable input was lost. These lessons will serve each member of the group well throughout their college and professional careers.

11 References

(n.d.). Retrieved from <http://www.truthout.org/us-consulate-employee-slain-drug-ridden-mexican-city57692>

All-Battery.com. (2009). Retrieved May 5, 2010, from All-Battery.com: <http://www.all-battery.com/Li-Ion18650111V10400mAhBatteryPackwithPCB31800.aspx>

Anzor Stainless Steel Fasteners. (2007). *Tensile Strength Comparison Chart*. Retrieved April 7, 2010, from anzor.com: <http://www.anzor.co.nz/?T=40>

Arduino. (2010, February 5). *Arduino Duemilanove*. Retrieved May 8, 2010, from <http://www.arduino.cc/en/Main/ArduinoBoardDuemilanove>

BatteryJunction.com. (2006). Retrieved May 5, 2010, from BatteryJunction.com: <http://www.batteryjunction.com/12v10nibapaf.html>

BatteryMart.com. (2010). Retrieved May 5, 2010, from BatteryMart.com: <http://www.batterymart.com/p-12v-14ah-sealed-lead-acid-battery.html>

City Rating. (2002). *CityRating*. Retrieved May 8, 2010, from City Rating: <http://www.cityrating.com/citytemperature.asp?City=El+Paso>

CompuLab. (2009). *fit-PC2 Specifications*. Retrieved May 5, 2010, from <http://www.fit-pc.com/web/fit-pc2/fit-pc2-specifications/>

Covert, A. (2008, July 21). *Waterproof Gadget Coating is Invisible, Mystifying, Mind Boggling Witchcraft*. Retrieved April 3, 2010, from Gizmodo: <http://gizmodo.com/5027545/waterproof-gadget-coating-is-invisible-mystifying-mind-boggling-witchcraft>

Hall, K. G. (2010, March 14). Retrieved May 5, 2010, from Truthout: <http://www.truthout.org/us-consulate-employee-slain-drug-ridden-mexican-city57692>

Hirzel, T. (2008, April 10). *PWM*. Retrieved April 15, 2010, from Arduino: <http://arduino.cc/en/Tutorial/PWM>

Hornback, P. (1998, April). Retrieved March 25, 2010, from fas.org/man/dod-101/sys/land/docs/2wheels98.pdf

JJenco. (2006). *EBOX Weatherproof Electronics Enclosure*. Retrieved May 4, 2010, from jjenco.com: <http://www.jjenco.com/Products/Electronics/ebox.htm>

Kod*Lab. (n.d.). *Kod*Lab*. Retrieved May 9, 2010, from University of Pennsylvania: <http://kodlab.seas.upenn.edu/RHex/Home>

My Fox Atlanta. (2010, May 2). *My Fox Atlanta*. Retrieved May 7, 2010, from My Fox Atlanta web site: http://www.myfoxatlanta.com/dpps/news/gang-violence-leaves-25-dead-near-mexico-us-border-dpgonc-km-20100502_7344313

RC Time Constant. (2010, April 14). Retrieved May 7, 2010, from Wikipedia.org:
http://en.wikipedia.org/wiki/RC_time_constant

Robot Stock News. (2006, May 31). *Robot Stock News*. Retrieved May 9, 2010, from Robot Stock News : <http://robotstocknews.blogspot.com/2006/05/irobot-wins-64-millions-navy-contract-html>

Wikipedia. (n.d.). *Continuous Tracks*. Retrieved May 5, 2010, from Wikipedia:
http://en.wikipedia.org/wiki/Continuous_track

Wikipedia. (n.d.). *Wheel*. Retrieved May 5, 2010, from Wikipedia:
<http://en.wikipedia.org/wiki/Wheel>

WowWee Group Limited. (n.d.). Retrieved May 7, 2010, from WowWee: Astonishing Imagination: <http://wowwee.com/en/products/tech/telepresence/rovio/rovio>

Appendix A: Selection of Team Project

Group

In the beginning of the project, it was not known which of the available projects would be best suited to the group's skill set or which was of the most interest. As a result of the dilemma, a decision matrix was created in which the projects were broken down into their positive and negative aspects relative to the group. The projects that the group ruled out based on each members skill-sets, or lack there-of, were excluded from being broken down in these categories. The results are as follows:

Eliminated- Biometric Monitoring, GPS Social Networking, Intruder Detection, Shot Detection, Sustainable Cooking, Remote Power Generation, Wind Power, Tracking Solar Collector

GPS Robot Car	
Pros:	Cons:
Challenges all group members	Potentially Expensive
Strong group interest	Heavy Systems integration
Sponsored	
Large customer base	
Home Automation	
Pros:	Cons:
Physically simple	Lack of interest
Relatively inexpensive	Small customer base
Challenges all group members	
Water Purification	
Pros:	Cons:
Strong benchmarking	No mechanical aspect
	No chemical engineers in group
Campus Recycling	
Pros:	Cons:
Great preliminary idea- Make cups from recycled materials	Very complex system
Real-worl problem	Potentially harmful to environment
Robotic Transporter	
Pros:	Cons:
Challenges all group members	Heavy systems integration

Table 10: Pros and Cons of Design Project Choices

Based on the table above, and because the skill sets present in the design group were heavily geared towards mechanical and computer engineering and programming, the group chose to pursue the Lockheed sponsored project entitled 'Real-Time Tracking and Control Via GPS Technology'.

Appendix B: Customer Requirements and Technical Specifications

Group

Customers:

- The main customer that we focused on was the U.S. Border patrol, with a concentration on the U.S. and Mexico border.
- Other viable customers for this product are included but not limited to any customer that needs remote surveillance that is out of the range of one person to monitor the entirety of the facility. Some examples follow:
 - Large Factories
 - Weapons Facilities
 - Prisons
 - Power Plants
 - Mental Institutions

Customer Requirements: In this project some of the customer requirements were within the scope of the project while others due to such things as time constraints and low budgets made requirements out of the scope of this project:

Requirements Inside Scope:

Intuitive Control – This was defined to having an average user operate the robot with no more than ten minutes of training. This was accomplished by a simple web server housed on an onboard computer that is able to create a user friendly website interface.

Transversal of Varied Conditions – This was defined as being able to negotiate a three inch high obstacle, and different surfaces such as sand, gravel, grass and up to three inch stones. This was accomplished by creating a track system that is composed of two bicycle chains, two sprockets and multiple treads on each side of the robot. This was connected with hubs and shafts to the motors that are then connected to the electrical system.

Long Range Cameras – This was defined as the user being able to distinguish a person from 100 yards away in the daylight. This is achieved by a camera being mounted on top of the robot. The camera is mounted to two motors that control its movement both in the horizontal and vertical planes and covered by an acrylic dome that will allow for clear vision, while protecting the camera itself. The camera data is then fed back to an onboard computer and back to the user at the base.

Night vision – This was defined as the robot being fully able to function at night. This is achieved by several infrared LEDs mounted on the shell of the robot, along with a camera that can detect infrared light. This is then fed through the electronics and back to the base where the user can still maintain sight in the dark where a human would not be able to.

Weather Resistant – This is defined as being able to withstand direct sunlight, two inches of rain and winds up to ten mph. This is achieved by covering the robot with protective lexan panels which will protect the delicate electronics inside the robot from various weather conditions.

Able to be transported in a pickup truck – This was defined as the entire robot being no more than six feet long by four feet wide. This was achieved by keeping the robot as small as possible.

The robot is also fairly light, around 30lbs with two fully charged batteries inside, which most adults could handle as a pair. This robot is also able to handle various terrain so ramps leading into the back of a truck, large car and or carrier van would not pose a problem to it.

Ability to determine location – This was defined as the ability to send GPS coordinates to a base computer. This is achieved using the previously mentioned on board computer. The signals are fed to the computer, organized and transferred to the website to be plotted onto a map.

This is a summarized table of all of the requirements above that were outlined in detail.
 (The code with CR # is Customer Requirement and then the number is the prioritization ranking of each requirement.)

Customer Requirement	Technical Specification
CR 1. Intuitive Control	Average user can operate robot with no more than 10 minutes of training
CR 2. Weather Resistant	Able to withstand direct sunlight, 2 inches of rain and winds up to 10mph.
CR 3. Can Traverse Varied Conditions	Able to negotiate three inch high obstacle. Able to negotiate sand, gravel, grass, and three inch stones.
CR 4. Ability to determine Location	Able to send GPS coordinates to a base computer.
CR 5. Long range Cameras	User can distinguish a person from 100 yards away in daylight
CR 6. Able to be transported in a pickup truck	Six feet long by four feet wide max.
CR 7. Night vision	User can distinguish a person from 50 yards away at night

Table 11: Key Customer Requirements and Relevant Technical Specifications

Requirements out of the Scope of this Project:

Fully operational in an inverted position – This would be a desirable feature due to the fact if the robot is traveling in the desert miles away from a base and it negotiates terrain that forces it to flip over, as it operates now a agent would have to be sent out in order to flip it back over. However if it could operate in an inverted position it would still be able to function even if it was upside down. This could have been completed by either software or an extra motor. The software was beyond the education of the students on this team and the extra motor was not an option due to the hardware that had already been used to its capacity with the motors that were already used.

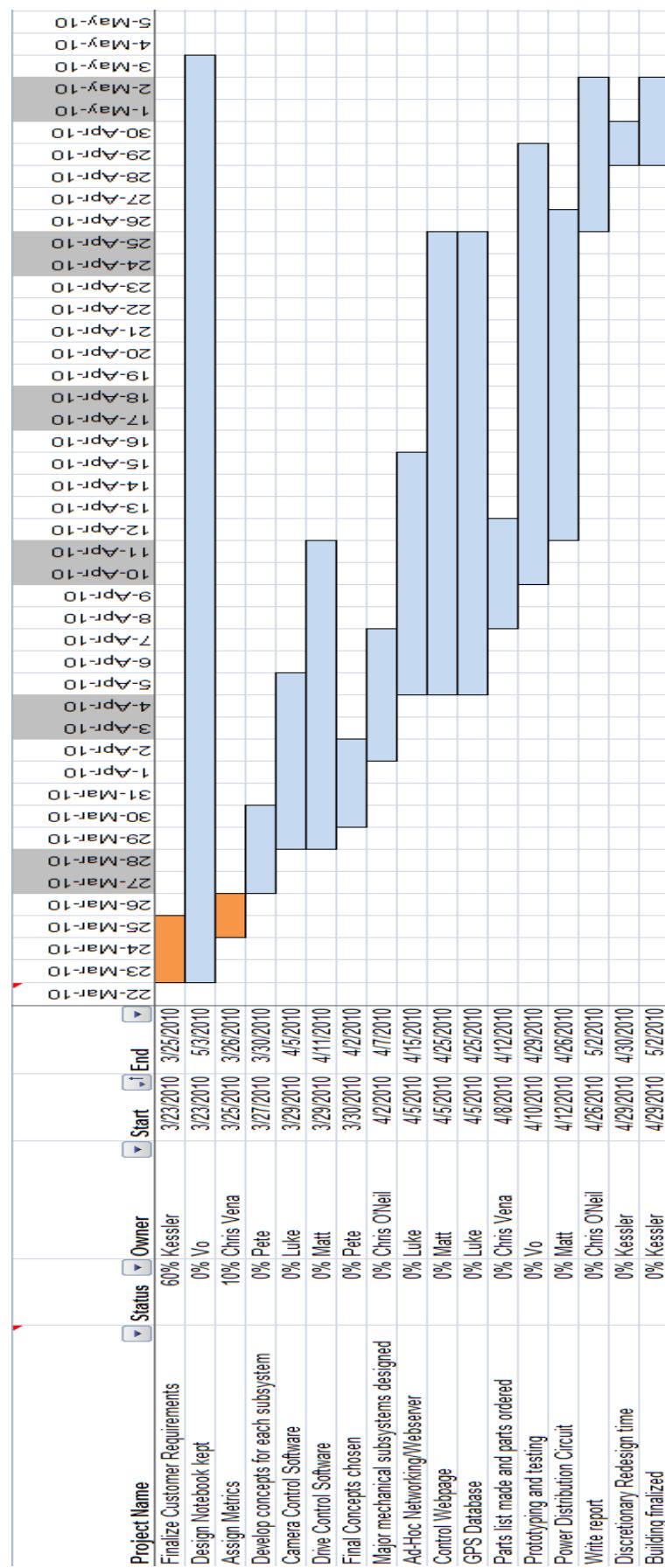
Resistant and defensible to human attacks – Resistance to human attack is desirable due to the fact there is nothing stopping someone who sees the robot surveying them from approaching it and destroying it, whether it be with a rock, baseball bat, or even some sort of gun. If the robot were to be destroyed however, at least the location of that person would be marked allowing for officers to

report to that location and apprehend the suspect. Resistance to attack would just have meant covering the robot with more expensive materials with a greater resistance to impact. As far as the robot defending itself, both lethal and non lethal force would not be looked highly upon as this is a school project. If implemented in a real situation such things as tazers and other more lethal weaponry could be incorporated into the design.

Very long capabilities – This would be helpful in the fact the longer the range of the camera, the fewer robots that would be necessary in order to survey the same amount of land. If this were the case it would cut down on both the initial cost and the cost of maintenance. This could be achieved by simply upgrading components. For the scope of this project it was not necessary and out of the budget range of the team.

Triangulation of actual humans position – This would allow for the pinpointing of the human target, unlike it is set up not to simply tag the location of the robot. There are two possible solutions to this problem. The first being the addition of an electronic range finder, which is very expensive and out of the budget range of this project. An alternative would be to add more advanced software, which is out of the range of knowledge of this group.

Appendix C: Gantt Chart



Appendix D: Expense Report

Chris Vena

Cost Estimate	Item Costs
Computer - compulab fitPC2	\$ 245.00
Micro-controller-Arduino	\$ 30.00
Servos - Hitec Hs422 x 2	\$ 26.00
Pan/ tilt camera mount - Lynxmotion pan/tilt kit	\$ 10.00
Webcam - Logitech C500	\$ 40.00
LM338K voltage regulator	\$ 8.25
Coaxial power plug for computer	\$ 1.95
IR LED's x 12	\$ 6.02
Tax	\$ 1.30
Wireless baby monitor	\$ 60.00
Acrylic dome	\$ 42.50
Chain	\$ 15.00
Metal	\$ 3.00
Sheet Steel	\$ 11.50
Bike Chains	\$ 32.23
Hardware	\$ 6.38
Plastic	\$ 23.64
Aluminum Stock	\$ 28.00
Garden hose	\$ 10.00
Rivets x 3 (1/8" X 1/4")	\$ 12.00
Rivets x 4 (1/8" X 1/2")	\$ 21.44
Solder	\$ 6.87
Waterjet parts	\$ 48.86
Washers	\$ 4.40
Total Cost	\$ 694.34

Table 12: Total Project Costs

Robots	6
Border patrol agent pay, per year	45000
Border patrol agent pay, per day	173.0769231
Money saved by 6 robots, per day	865.3846154
Days 6 robots paid off in	4.814090667

Table 13: Time to Pay Off the Purchase of Six Robots

For the most part, the group spent money wisely on supplies. The vast majority of the purchases were used on the robot. However, there were some items that were not used on the final design of the robot. The rivets, chains, and garden hose were supposed to be treads, but they ended up not being used. With proper planning and closer examination of the design of the treads and sprockets, this expense could have been spared. There are some items on the list that the group

would have utilized if more time was allotted. The plastic and the acrylic dome would have been implemented on the final product, but due to time constraints they were not used.

Despite these examples of excess spending, the overall cost of this project could have been much higher. A number of the potentially most expensive components, such as the onboard computer, webcam and Arduino were already owned by group members, and this saved the group money.

Appendix E: Statement of Work

- | Group |
|--|
| • Project Selection <ul style="list-style-type: none">○ All members of the group contributed ideas for selecting a final project. The project selected was Mike Kessler's idea, but was agreed upon by the group. |
| • Customer Requirements <ul style="list-style-type: none">○ This part of the project was headed by Mike Kessler who laid down the foundation for the requirements. Additional research was done by Chris O'neil who called the border patrol to get specific requirements. |
| • Project Design <ul style="list-style-type: none">○ Everyone in the group brainstormed about what should be a part of the robot and how it would be implemented.○ The CAD was primarily done by Mike Kessler and Lucas Doyle. |
| • Project Assembly <ul style="list-style-type: none">○ All the electrical components were done by Lucas Doyle and Matt Williamson.<ul style="list-style-type: none">▪ Lucas Doyle was primarily in charge of high-level software coding, notably the website, GPS database and NMEA parser, and camera streaming software while Matt Williamson built the power and control circuits and wrote the software running on the Arduino.○ As for the mechanical portion, it was split between the rest of the members<ul style="list-style-type: none">▪ Peter Willette and Chris Vena cut the angle iron for the frame, while Mike Kessler welded it together. Chris O'neil, Chris Vena, Mike Vo, and Peter Willette all worked together on cutting the garden hose and riveting them to the bike chain for the treads. Peter Willette made the front and rear hubs, and Mike Kessler and Chris Vena made the bearing cups.○ The actual assembly of the robot was done by everyone in the group. |
| • Final Report <ul style="list-style-type: none">○ Subsection authors as indicated in report○ Final compilation and editing:<ul style="list-style-type: none">▪ Luke Doyle and Matt Williamson |

Appendix F: Lessons Learned

Matt Williamson and Chris O'Neil:

In the mini project, we learned that sufficient time needs to be left to build. We left our building to the last week, and this made testing and writing a technical report very difficult and time constrained. Unfortunately, the group dynamic for this project left us in the same position, and there wasn't enough time to fully assemble the robot. Additionally, the technical report was left until the last week, which left too much work to be done in a very short time frame.

Mike Vo:

One of the biggest blunders made in the Mini-Project was not following the design process. Some of the steps such as benchmarking and lack of research of the project was ignored. Ignoring these steps led to some unforeseen problems in the beginning of the mini-project such as the idea of designing a trebuchet instead of the catapult. The building of the Mini-Project was a “Build-and-Bust” type process. Doing this type of design process wasted valuable time and resulted in a product that did not satisfy the main customer requirement, which was to hurl a marshmallow 20 feet.

Using the lessons learned and the experience from the mini project, the attitude going into this project was much different. The design process was explored more in depth and not much time was wasted due to inappropriate management of it. Almost all of the problems were foreseen from the beginning, though some were missed. Although there was a few setbacks and other needs that were not met, the accomplishment in completing this final project outweighs any failures, though it was disappointing in not seeing this robot succeed.

Mini-Project

- Did not stick to the design process
- Build and bust
- Worked at last minute
- Didn't prepare for anything
- Wasted many time focusing on building rather than designing
- Inadequate research conducted
 - Chose the hardest design

Major Project

- Followed design process
- Succeeded for the most part; failed with tracks, did not take the electronic mountings into the planning design
- Met most of the customer requirement; did not make it operate upside down/ did not make it see at night/ was not able to give it tracks...had to make improvised wheels...human error making wheels...

Luke Doyle:

As IED has gone from the mini project on to the final project, a major lesson sticks out in my mind. In the mini project, the technical challenges were comparatively small next to the final project. One group member could therefore get away with controlling all aspects of the subsystem and solve all the technical challenges themselves in the mini project without any action from their partners. In the final project, however, the technical challenges are immense, and no single member of the group can hope to try to grasp all the details of all the subsystems. This can prove disastrous to a person who is used to understanding systems in this level of detail, such as me. All in all, many lessons were learned over the course of the semester, and now that the semester has ended I'm sure I will reflect back on IED and learn something new every time from a past observations.

Peter Willette:

I learned that team work makes a large project much easier. In the final group project this actually hurt my mentality. I thought that whatever we took on would be accomplishable. However as a larger team, not everyone was as motivated as I was, this lead the team to select an overly complex project for the time frame that we had. I also learned that the project may not work as it did on testing day when it is presented. This made me skeptical of the performance of the teams final product and rightfully so, which kept with the golden rule that everything that can go wrong will go wrong.

Appendix G: User Manual

Peter Willette, Matt Williamson, Lucas Doyle

Mechanical Subsystem Troubleshooting:

Chain system: If there is a failure in the chain system then depending on how bad the failure is either one link or a whole track is able to be replaced.

1. Simply remove the broken link or entire track by using a chain splitter in order to remove a pin in the chain, or if the chain has a master link it can be take apart at the master link.
2. Place new link onto the existing track or place the new track over the sprockets.
3. Replace the master link.

Motor Failure: If there is a failure in one or both of the motors they can be replaced.

1. Set the robot on a supporting device that allows for free rotation of the tracks.
2. Remove both tracks by using a chain splitter in order to remove a pin in the chain, or if the chain has a master link it can be take apart at the master link.
3. Remove the bearing cup and supports on the hubs that are attached to the motor using an Allen wrench.
4. Using a screw driver remove the hubs attached to the motor by loosening the screw that secures them to the drive shaft of the motor.
5. Loosen the screws that fasten the motors to the brackets that are attached to the frame.
6. Disconnect the wires that are running to the motors from the connection inside the robot.
7. Gently remove them from the robot, while being sure not to damage any of the other electronics.

Front Hub Failure: If there is a failure such as cracking in the front hubs they can be replaced.

1. Set the robot on a supporting device that allows for free rotation of the tracks.
2. Remove both tracks by using a chain splitter in order to remove a pin in the chain, or if the chain has a master link it can be take apart at the master link.
3. Loosen the set screw that holds the black collar onto the shaft. Remove the collar.
4. Remove the hub.
5. If desired the sprockets can be removed from the hubs using an Allen wrench and loosening the screws that attach the hub and sprocket.

Rear Hub Failure: If there is a failure such as cracking in the rear (drive) hubs they can be replaced.

1. Set the robot on a supporting device that allows for free rotation of the tracks.
2. Remove both tracks by using a chain splitter in order to remove a pin in the chain, or if the chain has a master link it can be take apart at the master link.
3. Remove the bearing cup and supports on the hubs that are attached to the motor using an Allen wrench.
4. Using a screw driver remove the hubs attached to the motor by loosening the screw that secures them to the drive shaft of the motor.
5. Remove the screws that fasten the hubs and sprockets and replace the hub as needed.

Sprocket Failure: If there is a failure such as cracking in the front any of the sprockets they can be replaced.

1. Set the robot on a supporting device that allows for free rotation of the tracks.
2. Remove both tracks by using a chain splitter in order to remove a pin in the chain, or if the chain has a master link it can be take apart at the master link.
3. Pending which hub the sprocket is on that fails follow that procedure for removing that hub from the robot.
4. Remove the screws that fasten the hubs and sprockets and replace as needed.

Frame Failure: This would entail disassembly of the entire robot.

1. Set the robot on a supporting device that allows for free rotation of the tracks.
2. Remove both tracks by using a chain splitter in order to remove a pin in the chain, or if the chain has a master link it can be take apart at the master link.
3. Remove the bearing cup and supports on the hubs that are attached to the motor using an Allen wrench.
4. Using a screw driver remove the hubs attached to the motor by loosening the screw that secures them to the drive shaft of the motor.
5. Loosen the screws that fasten the motors to the brackets that are attached to the frame.
6. Disconnect the wires that are running to the motors from the connection inside the robot.
7. Gently remove them from the robot, while being sure not to damage any of the other electronics.
6. Loosen the set screw that holds the black collar onto the shaft. Remove the collar.
7. Remove the hub.

Electrical System Troubleshooting:

The electrical system is not designed to be user serviceable. If trouble is encountered, first ensure that the batteries are fully charged, and the onboard computer is powered on. In addition, ensure that the wireless internet connection to the robot is strong. If the robot is still malfunctioning, contact the manufacturer for further troubleshooting assistance.

Appendix H: Configuring Ubuntu to login and connect to Wi-Fi at boot

Lucas Doyle

The below instructions are a guide to configuring the robot to automatically connect to a wireless network when it first powers on. These instructions assume fresh install of Ubuntu with a keyboard, mouse and monitor plugged into the robot. The procedure of installing Ubuntu is beyond the scope of this document, but copies of Ubuntu as well as installation instructions can be downloaded here at <http://www.ubuntu.com>.

- From the desktop in a fresh copy of Ubuntu, press alt + f2 to open a run dialog.
- Enter the command **gksu /usr/sbin/gdmsetup**
- *Click run and enter the administrator password (initially set up when Ubuntu was first installed)*
- *In the Login Window Preferences, select the security tab and click the check box next to enable automatic login*
- *Click close to save the settings*
- *Next, click on the networking icon in the top panel and then select the desired network that the customer wants the robot to automatically connect to when powered on.*
- *Enter the authentication details for the network, click connect*
- *Wait and ensure a successful connection*
- *After a successful connection has been made, right click on the networking icon again and select edit connections*
- *Click the wireless tab and highlight the network the robot is currently connected to*
- *Click 'edit' and check the box that says 'connect automatically'*
- *Click close and then click close again to save all the settings*

Appendix I: Intro to HTML

Lucas Doyle

HTML stands for hypertext markup language, and is used by every website to format and deliver content that a web browser renders into a web page. It is the most low level aspect of web programming, meaning that it doesn't get any simpler than this. HTML's structure is focused around tags. This is an example of a simple HTML web page with structured with tags:

```
<!doctype html>
<html>
  <head>
    <title>Hello HTML</title>
  </head>
  <body>
    <p>Hello World!</p>
    <!-- This is a comment. It does nothing to the page -->
  </body>
</html>
```

The above is called a web page's "source code" and is saved in a file on a webserver. At the beginning of all source code, a doctype should be declared (in this example, it is generic html). This is so the webserver can tell what kind of document it is trying to render on screen for the user.

For the sake of this example, the source code will be stored in a file called “something.html” and stored on a server connected to the internet with its domain set to www.example.com. A user can therefore get to this page by navigating their web browser to <http://www.example.com/something.html>. This example produces a web page that simply displays the text “Hello World!” and a title (this appears in the title bar of the web browser) of “Hello HTML”.

From the example above, it is evident that the generic form for structuring tags is `<tag>content to be rendered</tag>`. Many tags, both common and uncommon, exist to define the elements of the web page and how they look and behave.

This is only a basic introduction to HTML, but it is all that is required to understand the general theory of how PHP works with respect to this project when generating HTML. To embed PHP into a web page, the following would be used:

```
<!doctype html>
<html>
  <head>
    <title>Hello HTML</title>
  </head>
  <body>
    <p>
      <?php
        // this is a comment, and does appear on the page.
        // PHP only evaluates uncommented code the webserver
        // sees inside of the <?php ?> tag. For example, the
        // following statement would print "Hello World!"
        // into the body of the webpage, making it appear
        // identical to the example above
        echo 'Hello World!';
      ?>
    </p>
  </body>
</html>
```

The 'echo' command is just one of thousands of potential functions PHP recognizes. This concludes the section on HTML, including the basics of embedding PHP in HTML. This basic technique is used throughout the entirety of the border patrol robot's operating code, except with more advanced and complex PHP and HTML. For examples, see the high level section of the main code appendix for examples of advanced PHP programming used to operate the robot.

Appendix J: Selection of Video Streaming Technique

Lucas Doyle

Streaming video to a web page in as close to real-time as possible is a problem that has been solved many different ways with varying degrees of success. The vast number of ways to go about solving this problem relates to the difference in hardware, software, operating systems and technical abilities of those involved in running the software. No universal solution exists at this time.

The following table details the team's various approaches to solving the camera problem. The problem is inherently difficult due to all of the different components needed to solve it. Therefore, this table assumes knowledge about common web programming techniques, networking knowledge, open source programs and Linux familiarity:

Approach	Performance
Taking pictures repeatedly and storing them on web server. Use AJAX to continually update picture.	Worked in principle, but proved to be jerky / too slow / randomly stop reloading the static image
VNC (provided a snapshot of the computer's desktop) embedded in a web browser with the webcam feed running full screen locally on the computer	Absolutely the most inefficient and improper use of processing power, however worked surprisingly well with respect to everything that was tried. Ultimately still did not give performance desired by the group.
Free web hosted broadcasting streaming service (AKA ustream, livestream, etc)	All the broadcasters were flash based and did not load properly. Even if they had, there would have advertisements in the webcam feed.
Ubuntu's built in webcam_server package	Java applet repeatedly did nothing when embedded on test web page despite following documentation
Flash streaming of any kind	Flash streaming is incompatible with Linux, specifically the V4L2 libraries
VLC streaming	HTTP streaming is the only way an image was able to be transmitted. Frame rate and quality was good, however there a massive delay due to a buffer that we really cannot change existed. Tried to get UDP and RTP streaming to work instead of http, but couldn't manage it. Also, real-time trans-coding of the video took up a most of the CPU
SSH with X11 display forwarding	VLC under X forwarding has a shared memory error. Mplayer sometimes works, but is too delayed / choppy to be useful
MJPEG streaming	Used mjpg-streamer – easy to set up and provided excellent quality and frame rate. The only encoding going on is jpeg compression so CPU load was negligible.
Skype auto answer	Latency issues with Skype's servers as well as poor Linux support of the Skype client. Only uses V4L as opposed to V4L2

Table 14: Potential Approaches for Streaming Remote Video from Robot to Customer

Appendix K: Installing a Linux Web Server

Lucas Doyle

In Ubuntu, installing and configuring a web server and associated programs is a one line command from the terminal:

- Click programs → accessories → terminal
- At the command prompt, enter sudo tasksel install lamp-server
- Enter the administrator password and follow the prompts

This command installs what is called the LAMP application stack. LAMP stands for Linux, Apache, PHP and MySQL. In concert, these four different programs provide free, powerful tools for making dynamic, modern websites. Briefly a little about each program and what it does:

- The Linux operating system is what all the programs in the stack are running on top of. This may seem redundant to include in the acronym, however other similar application stacks such as WAMP (where 'W' stands for windows) exist, so by including Linux in the acronym this confusion is avoided.
- Apache is an immensely popular and free web server. It processes requests and transmits files stored on the computer over the internet into a client's web browser. Other web servers exist, but Apache accounts for an overwhelming percentage of the web servers out there, and is very well documented.
- PHP is a popular high level web programming language originally designed to produce dynamic web pages. Since its inception, PHP has grown widely popular and its functionality has been expanded into other areas. It is used extensively throughout the project and without it; the group would have been unable to write the custom control programs necessary to control the robot from the internet.
- MySQL is a database server, able to quickly and efficiently organize and store a massive amount of data. Websites like facebook and Google maintain massive interconnected databases that work behind the scenes for users of the website. While the border patrol robot does incorporate some simple database functionality, it was determined that using MySQL would have been overkill for the needs of the project.

These four programs are very tightly integrated with each other, and have been around so long and are so widely used and well documented that no worthwhile alternative made sense to use for the robot. Normally, these four programs require a significant amount of configuration in order to work properly with one another. Fortunately, by installing them all at once in Ubuntu using the tasksel command, these settings and configuration files are automatically set, which saved the programmer an immense amount of time.

Appendix L: NMEA Sentence Structure

Lucas Doyle

Given the NMEA sentence:

\$GPRMC,040302.663,A,3939.7,N,10506.6,W,0.27,358.86,200804,*1A

Commas separate information in the sentence. The NMEA parser implemented in the border patrol robot looks for information between these commas. Here is a step by step breakdown of the contents of the sentence that are extracted by the parser that are important to the robot:

- \$GPRMC – This is the command word, indicating the type of NMEA sentence. Different types of NMEA sentences contain different information formatted in different ways. For the robot, the command word \$GPRMC is the only one that matters. In this case, the characters 'GP' indicate the sentence is being issued from the GPS and the 'RMC' means that the command contains the recommended minimum content for GPS location.
- 3939.7 – this is the latitude decimal degrees. It represents the distance north or south of the equator. This word is in the format "HHMM.M" where HH represents hours and MM.M represents minutes. This sentence says that the latitude is "39°39.7'N". The parser needs to divide the minutes by 60 and append this to the hours (e.g. 39.6617) in order to be understood by the mapping component of the software. All coordinates are manipulated and stored in this fashion once they are parsed from the sentence.
- N – Indicates which hemisphere the latitude is being measured from. The parser looks at this value to determine whether or not a minus sign should be added to the coordinates so they are plotted in a different hemisphere. 'N' is above the equator and therefore the coordinates need to be a positive value. Conversely, 'S' indicates that the coordinate should be negative
- 10506.6 – the longitude decimal degrees. Handled exactly the same way as latitude, although it indicates the distance from the prime meridian as opposed to the equator.
- W – the hemisphere longitude is measured in. 'W' means the value is measured to the left of the prime meridian, which means that the parser needs to negate the longitude distance in order for the coordinate to be useful on the map.

Appendix M: Motor Documentation



GM9236S020

Lo-Cog® DC Gearmotor

Assembly Data	Symbol	Units	Value	
Reference Voltage	E	V	24	
No-Load Speed	S _{NL}	rpm (rad/s)	236	(24.7)
Continuous Torque (Max.) ¹	T _C	oz-in (N-m)	153	(1.1E+00)
Peak Torque (Stall) ²	T _{PK}	oz-in (N-m)	860	(6.1E+00)
Weight	W _M	oz (g)	20.3	(576)
Motor Data				
Torque Constant	K _T	oz-in/A (N-m/A)	6.49	(4.58E-02)
Back-EMF Constant	K _E	V/krpm (V/rad/s)	4.80	(4.58E-02)
Resistance	R _T	Ω	2.49	
Inductance	L	mH	2.63	
No-Load Current	I _{NL}	A	0.16	
Peak Current (Stall) ²	I _P	A	9.64	
Motor Constant	K _M	oz-in/V/W (N-m/VW)	4.11	(2.90E-02)
Friction Torque	T _F	oz-in (N-m)	0.80	(5.6E-03)
Rotor Inertia	J _M	oz-in-s ² (kg-m ²)	1.0E-03	(7.1E-06)
Electrical Time Constant	τ _E	ms	1.06	
Mechanical Time Constant	τ _M	ms	8.5	
Viscous Damping	D	oz-in/krpm (N-m-s)	0.053	(3.5E-06)
Damping Constant	K _D	oz-in/krpm (N-m-s)	12.5	(8.5E-04)
Maximum Winding Temperature	θ _{MAX}	°F (°C)	311	(155)
Thermal Impedance	R _{TH}	°F/watt (°C/watt)	56.3	(13.5)
Thermal Time Constant	τ _{TH}	min	13.5	
Gearbox Data				
Reduction Ratio			19.7	
Efficiency ²			0.84	
Maximum Allowable Torque		oz-in (N-m)	500	(3.53)
Encoder Data				

1 - Specified at max. winding temperature at 25°C ambient without heat sink. 2 - Theoretical values supplied for reference only.

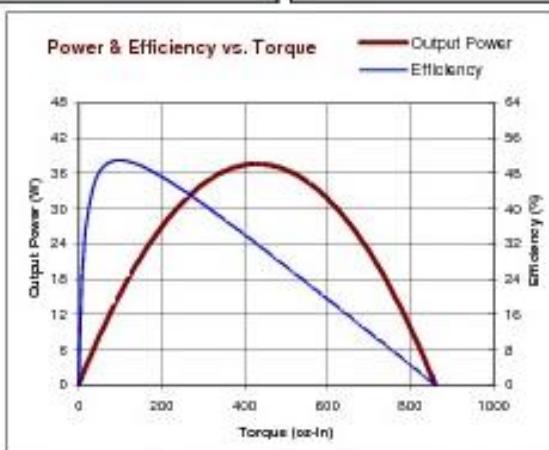
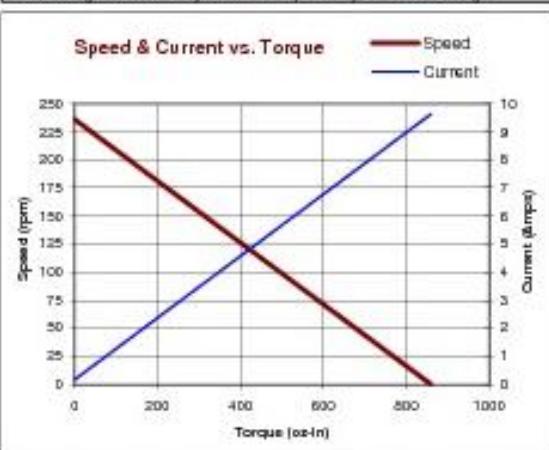
3 - Effective gearbox efficiency for the unit improved by use of ball bearings.

Included Features

- 2-Pole Stator
- Ceramic Magnets
- Heavy-Gauge Steel Housing
- 7-Slot Armature
- Silicon Steel Laminations
- Stainless Steel Shaft
- Copper-Graphite Brushes
- Diamond Turned Commutator
- Motor Ball Bearings
- Output Ball Bearing
- Wide Face Gears

Customization Options

- Alternate Winding
- Sleeve or Ball Bearings
- Modified Output Shaft
- Custom Cable Assembly
- Special Brushes
- EMI/RFI Suppression
- Alternate Gear Material
- Special Lubricant
- Optional Encoder
- Fail-Safe Brake



All values are nominal. Specifications subject to change without notice. Graphs are shown for reference only.

© 2001 Pittman.

UNCONTROLLED

Appendix N: GPS Documentation



PRODUCT USER MANUAL

GPS ENGINE BOARD

ET-312

GlobalSat Technology Corporation

Headquarters (Far East Century Park)

16F., No. 186, Jian-Yi Road, Chung-Ho City, Taipei Hsien 235, Taiwan

Tel: 886-2-8226-3799/ Fax: 886-2-8226-3899

E-mail : service@globalsat.com.tw

Website: www.globalsat.com.tw

Specification:

General

Chipset	SiRF Star III
Frequency	L1, 1575.42 MHz
C/A code	1.023 MHz chip rate
Channels	20 channel all-in-view tracking
Sensitivity	-159 dBm

Accuracy

Position	10 meters, 2D RMS
	5 meters, 2D RMS, WAAS enabled
Velocity	0.1 m/s
Time	1us synchronized to GPS time

Datum

Default	WGS-84
---------	--------

Acquisition Time

Reacquisition	0.1 sec., average
Hot start	1 sec., average
Warm start	38 sec., average
Cold start	42 sec., average

Dynamic Conditions

Altitude	18,000 meters (60,000 feet) max
Velocity	515 meters /second (1000 knots) max
Acceleration	Less than 4g
Jerk	20m/sec **3

Power

Main power input	3.3V +/- 5% DC input
Power consumption	80mA (Continuous mode)
	65mA(Trickle power mode)

Interface

Dimension	27.9mm * 20mm * 2.9mm
Baud rate	4,800 to 57,600 bps adjustable
Output message	SiRF binary or NMEA 0183 GGA, GSA, GSV, RMC, VTG, GLL

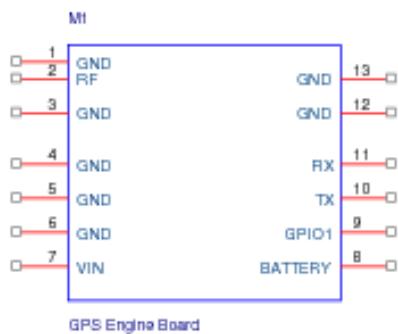
Antenna

Active or passive antenna

Environmental

Operating Temp	-40°C to +85°C
----------------	----------------

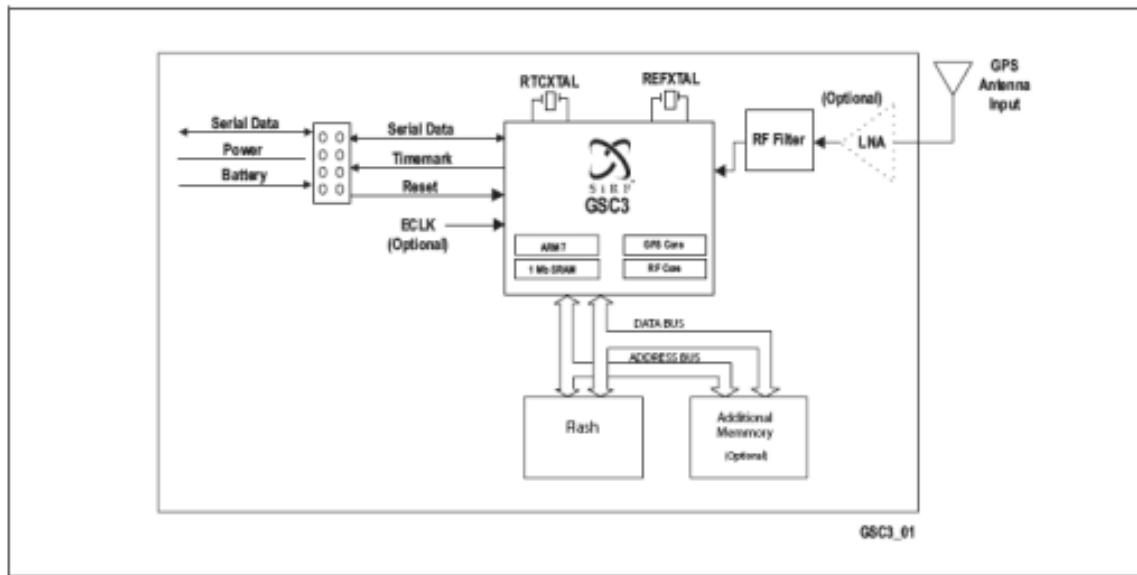
Pin Assignment



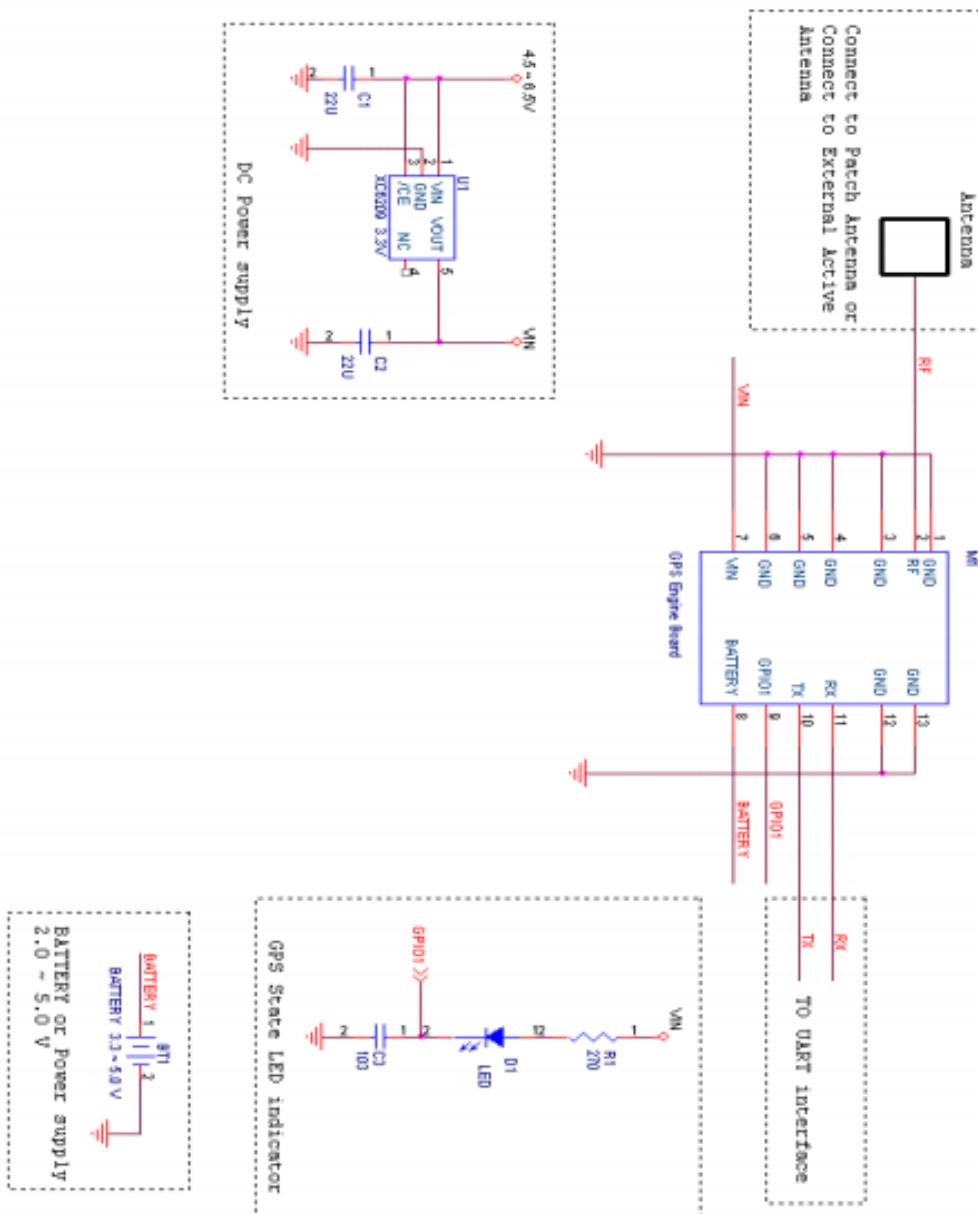
Pin description

Pin NO.	Pin Name	Remark
1.	GND	Ground.
2.	RF	Connect to Patch Antenna or Connect to External Active Antenna. While external antenna is used, the optional power is needed.
3.	GND	Ground.
4.	GND	Ground.
5.	GND	Ground.
6.	GND	Ground.
7.	VIN	This is the main DC supply for a 3.3V +- 5% DC input power module board.
8.	BATTERY	This is the battery backup input that powers the SRAM and RTC when main power is removed. Typical current draw is 15uA. Without an external backup battery, the module/engine board will execute a cold start after every turn on. To achieve the faster start-up offered by a hot or warm start, a battery backup must be connected. The battery voltage should be between 2.0v and 5.0v.
9.	GPIO1	User can use this I/O pin for special function. For example, on/off LED
10.	TX	This is the main transmits channel for outputting navigation and measurement data to user's navigation software or user written software. Output TTL level, 0V ~ 2.85V
11.	RX	This is the main receive channel for receiving software commands to the engine board from SiRFdemo software or from user written software.
12.	GND	Ground.
13.	GND	Ground.

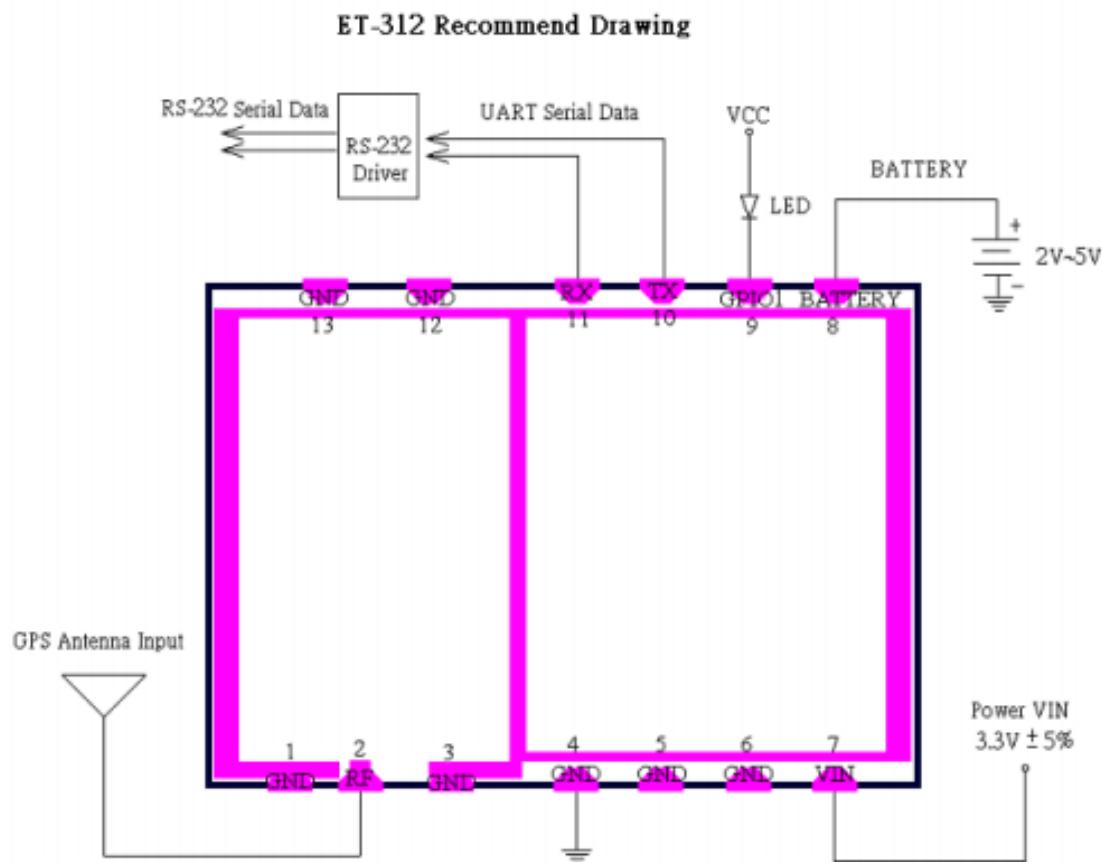
Block Diagram



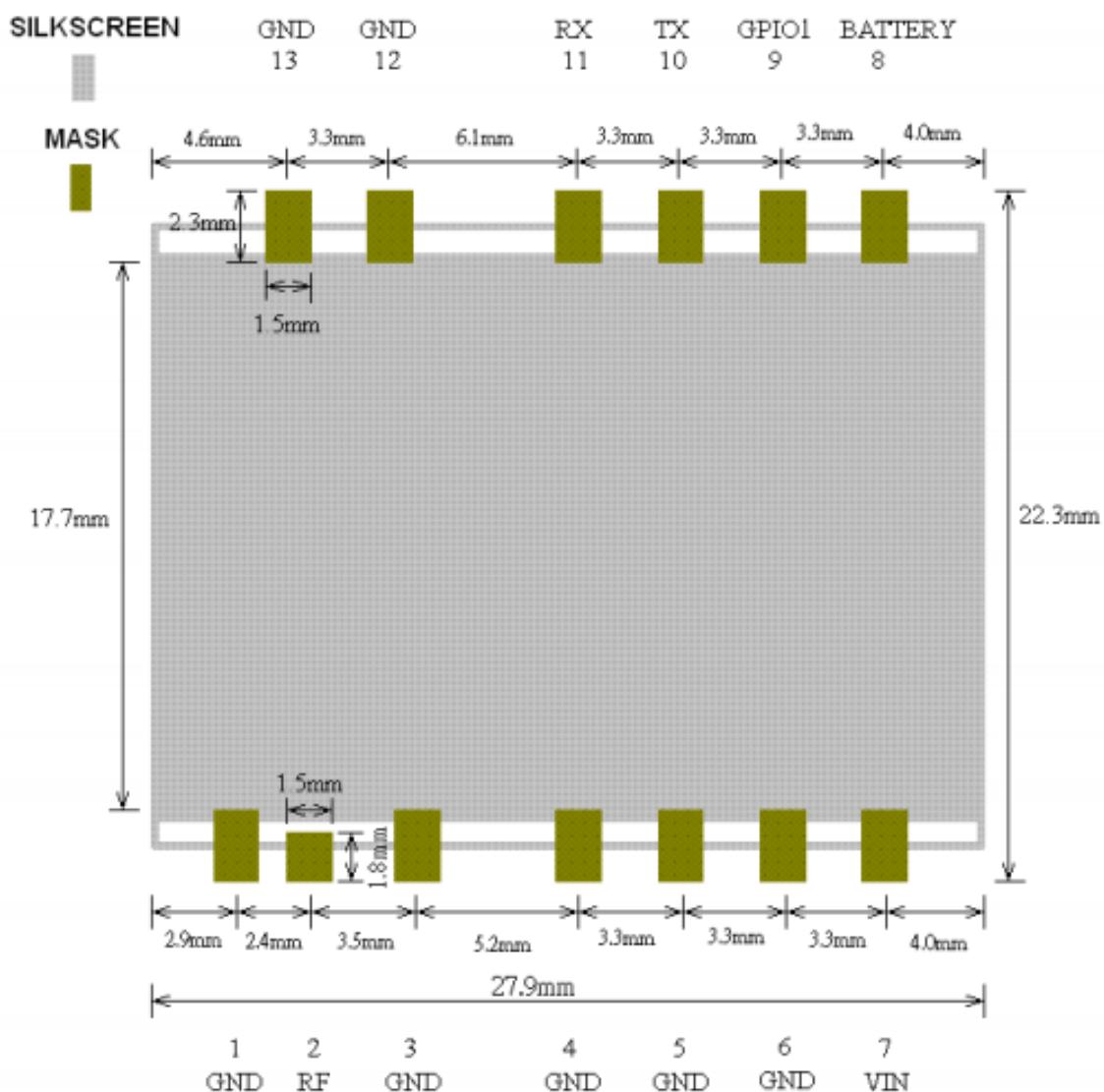
Reference Design



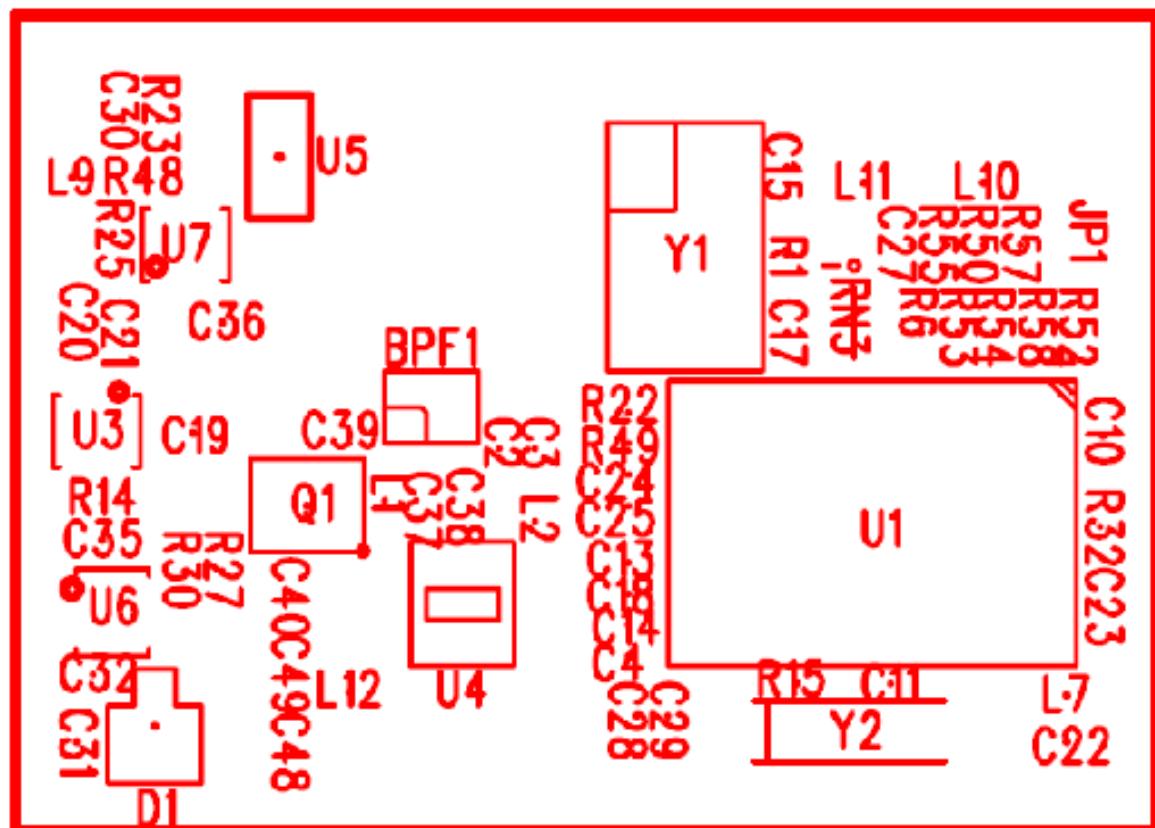
Recommend Drawing



Recommend Layout PAD



Placement Drawing



Appendix O: Battery Documentation



SHENZHEN SUNNYWAY BATTERY TECH CO.,LTD.

SWE12140(12V14AH)



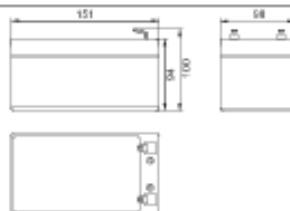
● Specifications

Nominal Voltage		12V
Rated capacity (20 hour rate)		14Ah
Dimensions	Total Height	100 mm (3.94 inches)
	Height	94 mm (3.70 inches)
	Length	151 mm (5.94 inches)
	Width	98 mm (3.86 inches)
Weight Approx		4.2 Kg (9.28 lbs)

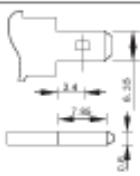
● Characteristics

Capacity 25°C(77°F)	20 hour rate(0.70 A)	14 AH
	10 hour rate(1.28 A)	12.8 AH
	5 hour rate(2.40 A)	12.0 AH
	1 hour rate(8.40 A)	8.40 AH
Internal Resistance	Full charged Battery at 25°C(77°F)	12.5 mΩ
Capacity affected by Temperature (20hour rate)	40°C(104°F) 25°C(77°F) 0°C(32°F) -15°C(5°F)	102% 100% 85% 65%
Self-Discharge at 25°C(77°F)	Capacity after 3 month storage Capacity after 6 month storage Capacity after 12 month storage	91% 82% 64%
Terminal	T251	
Charge (constant Voltage)	Initial Charging Current less than 5.60A Voltage 14.40-14.70V	
Float	Voltage 13.50-13.80V	

● Outer dimensions (mm)

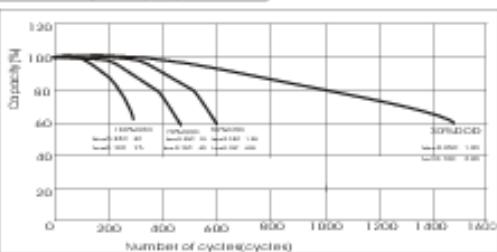


● Terminal Type (mm)

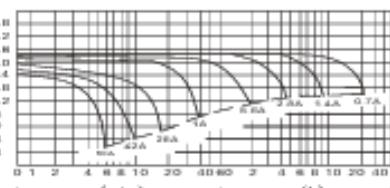


T251

● Curves of cycle life(25°C)



● Discharge Curves 25°C(77°F)



Constant Current(Amp) and Constant Power(Watt) Discharge Table at 25°C(77°F)

Time		5min	10min	15min	30min	1h	2h	3h	4h	5h	8h	10h	20h
9.60V	A	50.4	33.0	24.5	16.1	8.4	5.0	3.6	2.8	2.50	1.6	1.3	0.72
	W	594.4	373.3	282.3	170.9	96.8	56.7	41.7	33.5	28.4	18.7	15.3	8.40
10.20V	A	46.2	31.6	22.5	15.3	7.8	4.7	3.5	2.8	2.40	1.60	1.30	0.71
	W	559.4	353.5	265.4	169.8	91.0	54.5	40.6	32.4	27.9	18.5	15.0	8.20
10.50V	A	42.1	29.5	21.0	14.8	7.63	4.6	3.4	2.66	2.40	1.60	1.30	0.7
	W	540.2	343.0	253.8	168.0	88.4	53.4	39.8	30.8	27.8	18.3	15.0	8.10
10.80V	A	40.4	28.2	19.6	14.4	7.37	4.5	33.8	2.62	2.30	1.53	12.50	0.69
	W	473.7	332.5	244.4	167.4	85.8	52.2	39.4	30.5	26.50	17.5	14.6	7.90
11.10V	A	37.4	26.6	18.2	14.0	7.2	4.4	3.2	2.60	2.20	1.50	1.23	0.67
	W	457.9	321.4	232.8	166.3	84.5	52.0	38.2	30.3	26.0	17.0	14.30	7.88

SUNNYWAY®

Appendix P: DC-DC Converter Documentation



May 1998

LM138/LM338 5-Amp Adjustable Regulators

General Description

The LM138 series of adjustable 3-terminal positive voltage regulators is capable of supplying in excess of 5A over a 1.2V to 32V output range. They are exceptionally easy to use and require only 2 resistors to set the output voltage. Careful circuit design has resulted in outstanding load and line regulation—comparable to many commercial power supplies. The LM138 family is supplied in a standard 3-lead transistor package.

A unique feature of the LM138 family is time-dependent current limiting. The current limit circuitry allows peak currents of up to 12A to be drawn from the regulator for short periods of time. This allows the LM138 to be used with heavy transient loads and speeds start-up under full-load conditions. Under sustained loading conditions, the current limit decreases to a safe value protecting the regulator. Also included on the chip are thermal overload protection and safe area protection for the power transistor. Overload protection remains functional even if the adjustment pin is accidentally disconnected.

Normally, no capacitors are needed unless the device is situated more than 6 inches from the input filter capacitors in which case an input bypass is needed. An output capacitor can be added to improve transient response, while bypassing the adjustment pin will increase the regulator's ripple rejection.

Besides replacing fixed regulators or discrete designs, the LM138 is useful in a wide variety of other applications. Since the regulator is "floating" and sees only the input-to-output differential voltage, supplies of several hundred volts can be regulated as long as the maximum input to output differential is not exceeded, i.e., do not short-circuit output to ground. The part numbers in the LM138 series which have a K suffix are packaged in a standard Steel TO-3 package, while those with a T suffix are packaged in a TO-220 plastic package. The LM138 is rated for $-55^{\circ}\text{C} \leq T_j \leq +150^{\circ}\text{C}$, and the LM338 is rated for $0^{\circ}\text{C} \leq T_j \leq +125^{\circ}\text{C}$.

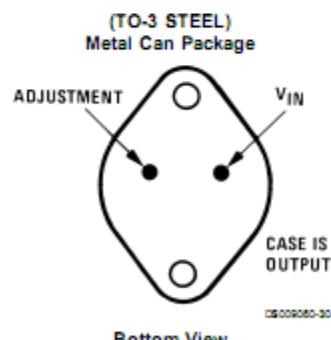
Features

- Guaranteed 7A peak output current
- Guaranteed 5A output current
- Adjustable output down to 1.2V
- Guaranteed thermal regulation
- Current limit constant with temperature
- P+ Product Enhancement tested
- Output is short-circuit protected

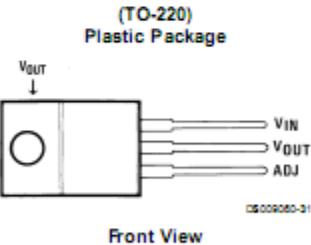
Applications

- Adjustable power supplies
- Constant current regulators
- Battery chargers

Connection Diagrams (See Physical Dimension section for further information)



Order Number LM138K STEEL or LM338K STEEL
See NS Package Number K02A



LM138/LM338 5-Amp Adjustable Regulators

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office or Distributors for availability and specifications.

(Note 4)

Power Dissipation	Internally limited
Input/Output Voltage Differential	+40V, -0.3V
Storage Temperature	-65°C to +150°C

Lead Temperature		
Metal Package (Soldering, 10 seconds)	300°C	
Plastic Package (Soldering, 4 seconds)	260°C	
ESD Tolerance	TBD	

Operating Temperature Range

LM138	-55°C ≤ T _J ≤ +150°C
LM338	0°C ≤ T _J ≤ +125°C

Electrical Characteristics

Specifications with standard type face are for T_J = 25°C, and those with boldface type apply over full Operating Temperature Range. Unless otherwise specified, V_{IN} - V_{OUT} = 5V; and I_{OUT} = 10 mA. (Note 2)

Symbol	Parameter	Conditions	LM138			Units
			Min	Typ	Max	
V _{REF}	Reference Voltage	3V ≤ (V _{IN} - V _{OUT}) ≤ 35V, 10 mA ≤ I _{OUT} ≤ 5A, P ≤ 50W	1.19	1.24	1.29	V
V _{RLINE}	Line Regulation	3V ≤ (V _{IN} - V _{OUT}) ≤ 35V (Note 3)		0.005	0.01	%/V
V _{LOAD}	Load Regulation	10 mA ≤ I _{OUT} ≤ 5A (Note 3)		0.1	0.3	%
	Thermal Regulation	20 ms Pulse		0.002	0.01	%/W
I _{ADJ}	Adjustment Pin Current			45	100	μA
ΔI _{ADJ}	Adjustment Pin Current Change	10 mA ≤ I _{OUT} ≤ 5A, 3V ≤ (V _{IN} - V _{OUT}) ≤ 35V		0.2	5	μA
ΔV _{RT}	Temperature Stability	T _{MIN} ≤ T _J ≤ T _{MAX}		1		%
I _{LOAD(Min)}	Minimum Load Current	V _{IN} - V _{OUT} = 35V		3.5	5	mA
I _{OL}	Current Limit	V _{IN} - V _{OUT} ≤ 10V DC		5	8	A
		0.5 ms Peak	7	12		A
		V _{IN} - V _{OUT} = 30V		1	1	A
V _N	RMS Output Noise, % of V _{OUT}	10 Hz ≤ f ≤ 10 kHz		0.003		%
$\frac{\Delta V_R}{\Delta V_{IN}}$	Ripple Rejection Ratio	V _{OUT} = 10V, f = 120 Hz, C _{ADJ} = 0 μF		60		dB
		V _{OUT} = 10V, f = 120 Hz, C _{ADJ} = 10 μF	60	75		dB
	Long-Term Stability	T _J = 125°C, 1000 Hrs		0.3	1	%
θ _{JC}	Thermal Resistance, Junction to Case	K Package			1	°C/W
θ _{JA}	Thermal Resistance, Junction to Ambient (No Heat Sink)	K Package		35		°C/W

Electrical Characteristics

Symbol	Parameter	Conditions	LM338			Units
			Min	Typ	Max	
V _{REF}	Reference Voltage	3V ≤ (V _{IN} - V _{OUT}) ≤ 35V, 10 mA ≤ I _{OUT} ≤ 5A, P ≤ 50W	1.19	1.24	1.29	V
V _{RLINE}	Line Regulation	3V ≤ (V _{IN} - V _{OUT}) ≤ 35V (Note 3)		0.005	0.03	%/V
V _{LOAD}	Load Regulation	10 mA ≤ I _{OUT} ≤ 5A (Note 3)		0.1	0.5	%
	Thermal Regulation	20 ms Pulse		0.002	0.02	%/W
I _{ADJ}	Adjustment Pin Current			45	100	μA
ΔI _{ADJ}	Adjustment Pin Current Change	10 mA ≤ I _{OUT} ≤ 5A, 3V ≤ (V _{IN} - V _{OUT}) ≤ 35V		0.2	5	μA

Electrical Characteristics (Continued)

Symbol	Parameter	Conditions	LM338			Units	
			Min	Typ	Max		
ΔV_{RT}	Temperature Stability	$T_{MIN} \leq T_J \leq T_{MAX}$		1		%	
$I_{LOAD(Min)}$	Minimum Load Current	$V_{IN} - V_{OUT} = 35V$		3.5	10	mA	
I_{CL}	Current Limit	$V_{IN} - V_{OUT} \leq 10V$		5	8	A	
		DC		7	12	A	
		0.5 ms Peak			1	A	
V_N	RMS Output Noise, % of V_{OUT}	$V_{IN} - V_{OUT} = 30V$			0.003	%	
		10 Hz $\leq f \leq 10$ kHz					
$\frac{\Delta V_R}{\Delta V_{IN}}$	Ripple Rejection Ratio	$V_{OUT} = 10V, f = 120$ Hz, $C_{ADJ} = 0$ μF		60		dB	
		$V_{OUT} = 10V, f = 120$ Hz, $C_{ADJ} = 10$ μF	60	75		dB	
Long-Term Stability			$T_J = 125^\circ C$, 1000 hrs		0.3	1	%
θ_{JC}	Thermal Resistance Junction to Case	K Package			1	°C/W	
		T Package			4	°C/W	
θ_{JA}	Thermal Resistance, Junction to Ambient (No Heat Sink)	K Package		35		°C/W	
		T Package		50		°C/W	

Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. Operating Ratings indicate conditions for which the device is intended to be functional, but do not guarantee specific performance limits. For guaranteed specifications and test conditions, see the Electrical Characteristics.

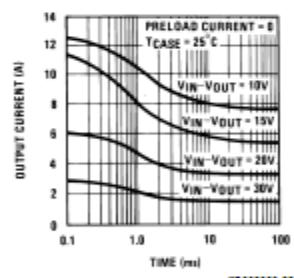
Note 2: These specifications are applicable for power dissipations up to 50W for the TO-3 (K) package and 25W for the TO-220 (T) package. Power dissipation is guaranteed at these values up to 15V input-output differential. Above 15V differential, power dissipation will be limited by internal protection circuitry. All limits (i.e., the numbers in the Min. and Max. columns) are guaranteed to National's AOQL (Average Outgoing Quality Level).

Note 3: Regulation is measured at a constant junction temperature, using pulse testing with a low duty cycle. Changes in output voltage due to heating effects are covered under the specifications for thermal regulation.

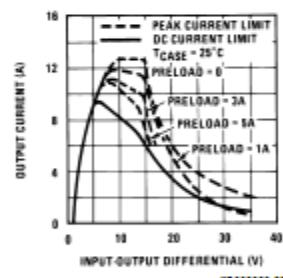
Note 4: Refer to RETS138K drawing for military specifications of LM138K.

Typical Performance Characteristics

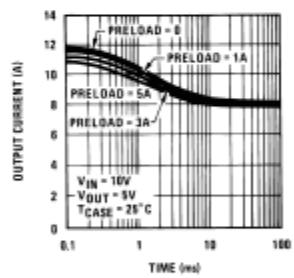
Current Limit



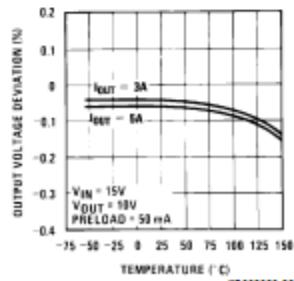
Current Limit



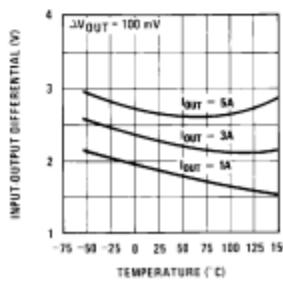
Current Limit



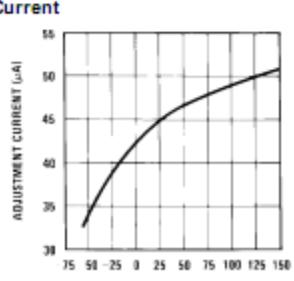
Load Regulation



Dropout Voltage

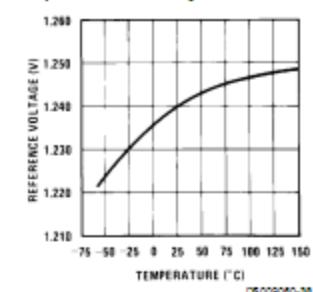


Adjustment Current

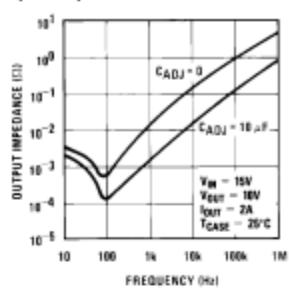


Typical Performance Characteristics (Continued)

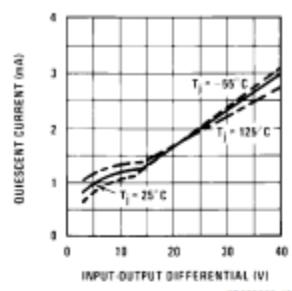
Temperature Stability



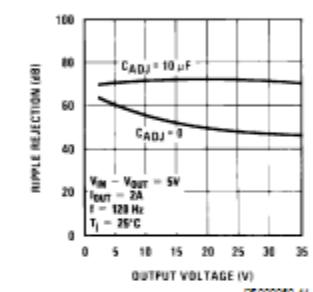
Output Impedance



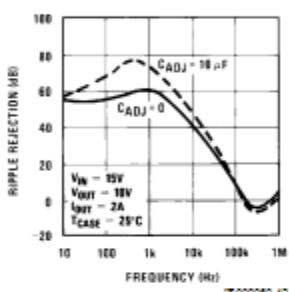
Minimum Operating Current



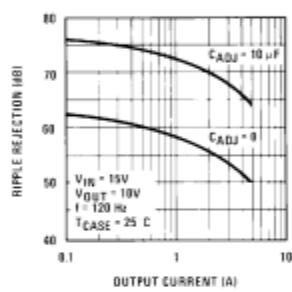
Ripple Rejection



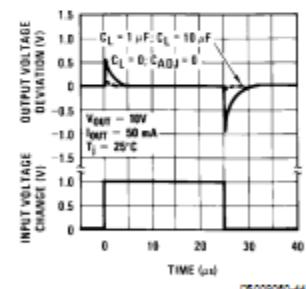
Ripple Rejection



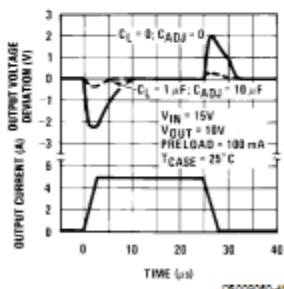
Ripple Rejection



Line Transient Response



Load Transient Response



Application Hints

In operation, the LM138 develops a nominal 1.25V reference voltage, V_{REF} , between the output and adjustment terminal. The reference voltage is impressed across program resistor R1 and, since the voltage is constant, a constant current I_1 then flows through the output set resistor R2, giving an output voltage of

$$V_{OUT} = V_{REF} \left(1 + \frac{R_2}{R_1} \right) + I_{ADJ} R_2.$$

Application Hints (Continued)

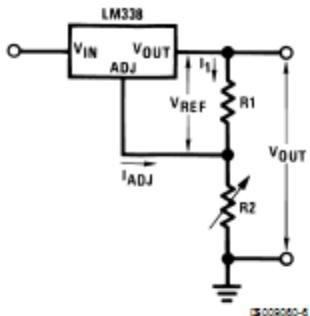


FIGURE 1.

Since the 50 μ A current from the adjustment terminal represents an error term, the LM138 was designed to minimize I_{ADJ} and make it very constant with line and load changes. To do this, all quiescent operating current is returned to the output establishing a minimum load current requirement. If there is insufficient load on the output, the output will rise.

External Capacitors

An input bypass capacitor is recommended. A 0.1 μ F disc or 1 μ F solid tantalum on the input is suitable input bypassing for almost all applications. The device is more sensitive to the absence of input bypassing when adjustment or output capacitors are used but the above values will eliminate the possibility of problems.

The adjustment terminal can be bypassed to ground on the LM138 to improve ripple rejection. This bypass capacitor prevents ripple from being amplified as the output voltage is increased. With a 10 μ F bypass capacitor 75 dB ripple rejection is obtainable at any output level. Increases over 20 μ F do not appreciably improve the ripple rejection at frequencies above 120 Hz. If the bypass capacitor is used, it is sometimes necessary to include protection diodes to prevent the capacitor from discharging through internal low current paths and damaging the device.

In general, the best type of capacitors to use are solid tantalum. Solid tantalum capacitors have low impedance even at high frequencies. Depending upon capacitor construction, it takes about 25 μ F in aluminum electrolytic to equal 1 μ F solid tantalum at high frequencies. Ceramic capacitors are also good at high frequencies; but some types have a large decrease in capacitance at frequencies around 0.5 MHz. For this reason, 0.01 μ F disc may seem to work better than a 0.1 μ F disc as a bypass.

Although the LM138 is stable with no output capacitors, like any feedback circuit, certain values of external capacitance can cause excessive ringing. This occurs with values between 500 μ F and 5000 μ F. A 1 μ F solid tantalum (or 25 μ F aluminum electrolytic) on the output swamps this effect and insures stability.

Load Regulation

The LM138 is capable of providing extremely good load regulation but a few precautions are needed to obtain maximum performance. The current set resistor connected between the adjustment terminal and the output terminal (usually 240 Ω) should be tied directly to the output of the regulator (case) rather than near the load. This eliminates line drops from appearing effectively in series with the reference and degrading regulation. For example, a 15V regulator with 0.05 Ω resistance between the regulator and load will have a load regulation due to line resistance of $0.05\Omega \times I_L$. If the set resistor is connected near the load the effective line resistance will be $0.05\Omega (1 + R_2/R_1)$ or in this case, 11.5 times worse.

Figure 2 shows the effect of resistance between the regulator and 240 Ω set resistor.

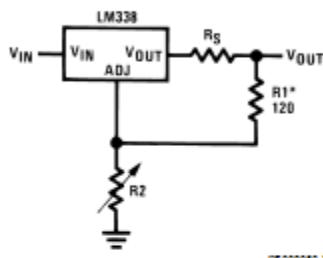


FIGURE 2. Regulator with Line Resistance in Output Lead

With the TO-3 package, it is easy to minimize the resistance from the case to the set resistor, by using 2 separate leads to the case. The ground of R2 can be returned near the ground of the load to provide remote ground sensing and improve load regulation.

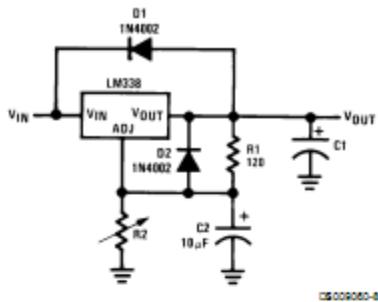
Protection Diodes

When external capacitors are used with any IC regulator it is sometimes necessary to add protection diodes to prevent the capacitors from discharging through low current points into the regulator. Most 20 μ F capacitors have low enough internal series resistance to deliver 20A spikes when shorted. Although the surge is short, there is enough energy to damage parts of the IC.

When an output capacitor is connected to a regulator and the input is shorted, the output capacitor will discharge into the output of the regulator. The discharge current depends on the value of the capacitor, the output voltage of the regulator, and the rate of decrease of V_{IN} . In the LM138 this discharge path is through a large junction that is able to sustain 25A surge with no problem. This is not true of other types of positive regulators. For output capacitors of 100 μ F or less at output of 15V or less, there is no need to use diodes.

The bypass capacitor on the adjustment terminal can discharge through a low current junction. Discharge occurs when either the input or output is shorted. Internal to the LM138 is a 50 Ω resistor which limits the peak discharge current. No protection is needed for output voltages of 25V or less and 10 μ F capacitance. Figure 3 shows an LM138 with protection diodes included for use with outputs greater than 25V and high values of output capacitance.

Application Hints (Continued)



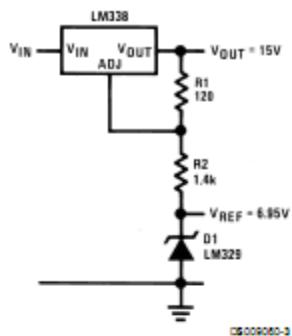
D1 protects against C1
D2 protects against C2

$$V_{OUT} = 1.25V \left(1 + \frac{R_2}{R_1} \right) + I_{ADJ}R_2$$

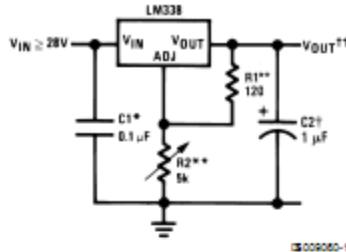
FIGURE 3. Regulator with Protection Diodes

Typical Applications

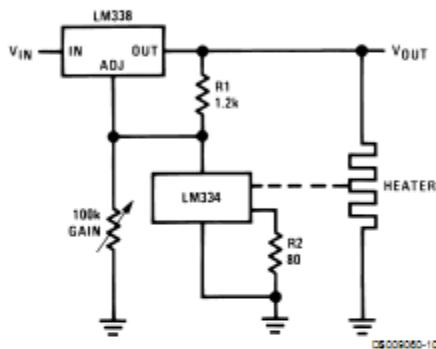
Regulator and Voltage Reference



1.2V–25V Adjustable Regulator



Temperature Controller



Full output current not available
at high input-output voltages

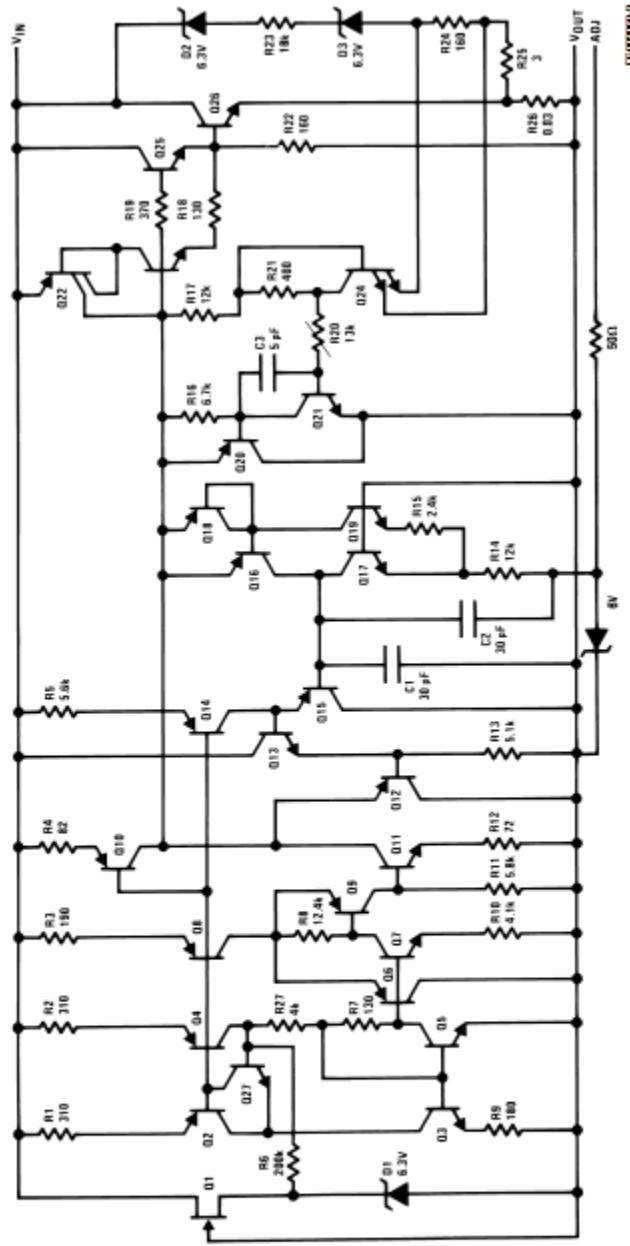
[†]Optional—Improves transient response. Output capacitors in the range of 1 μF to 1000 μF of aluminum or tantalum electrolytic are commonly used to provide improved output impedance and rejection of transients.

^{*}Needed if device is more than 6 inches from filter capacitors.

$$\dagger\dagger V_{OUT} = 1.25V \left(1 + \frac{R_2}{R_1} \right) + I_{ADJ} (R_2)$$

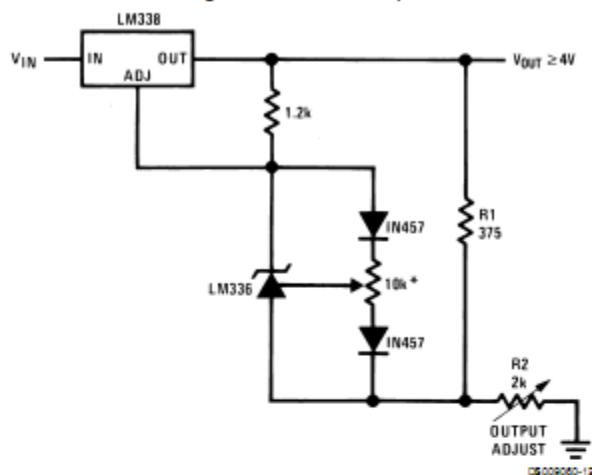
^{**}R1 = 240Ω for LM138. R1, R2 as an assembly can be ordered from Bourns:
MIL part no. 7105A-AT2402
COMM part no. 7105A-AT7402

Schematic Diagram



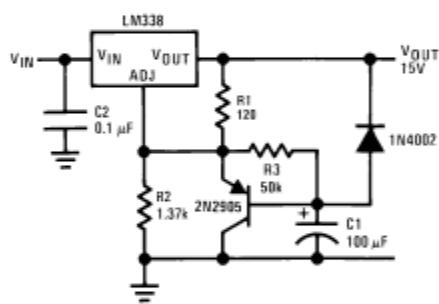
Typical Applications

Precision Power Regulator with Low Temperature Coefficient



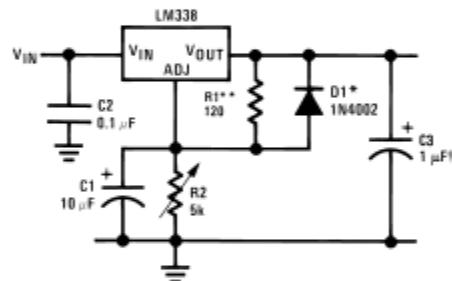
* Adjust for 3.75 across R1

Slow Turn-On 15V Regulator



DS009080-13

Adjustable Regulator with Improved Ripple Rejection



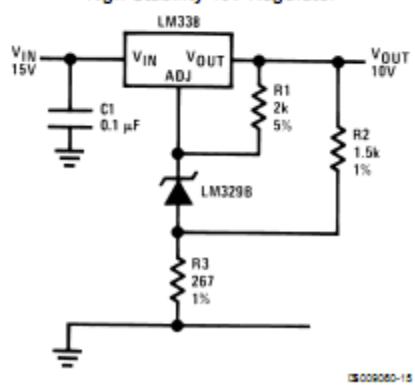
DS009080-14

† Solid tantalum

*Discharges C1 if output is shorted to ground

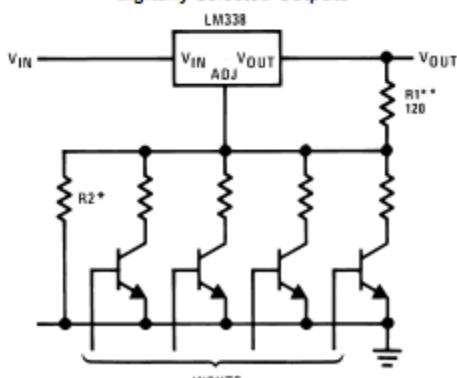
**R1 = 2400 for LM138

High Stability 10V Regulator



DS009080-15

Digitaly Selected Outputs



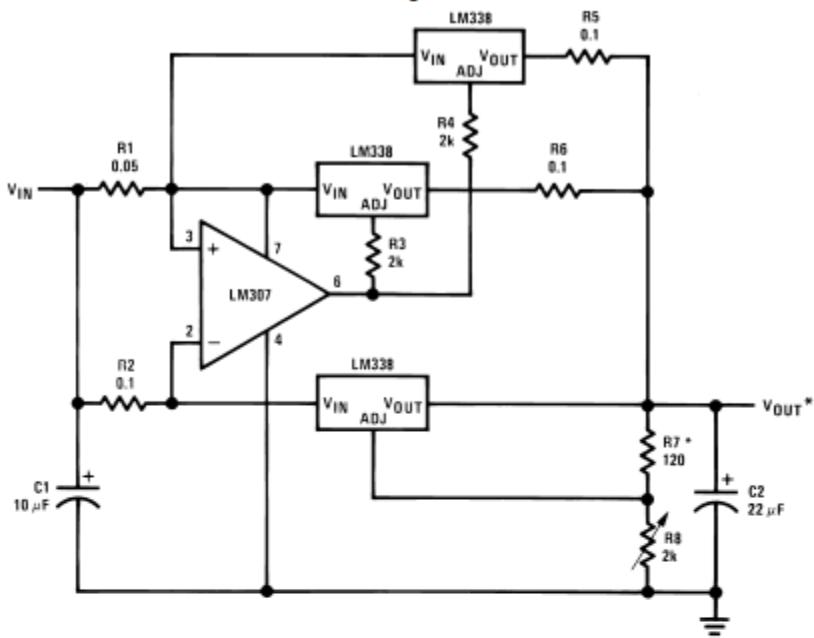
DS009080-16

* Sets maximum VOUT

**R1 = 2400 for LM138

Typical Applications (Continued)

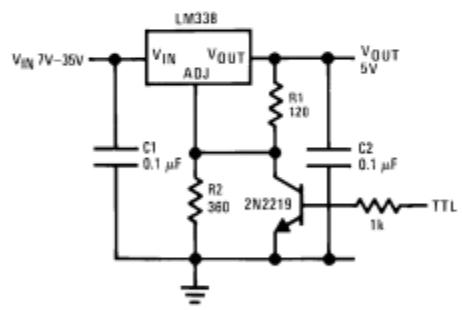
15A Regulator



CS009050-17

* Minimum load = 100 mA

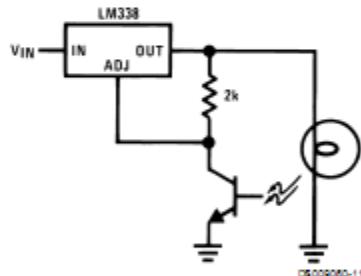
5V Logic Regulator with Electronic Shutdown**



CS009050-18

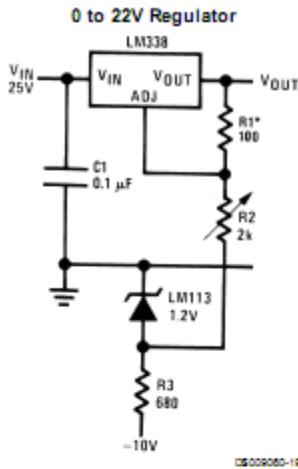
** Minimum output = 1.2V

Light Controller

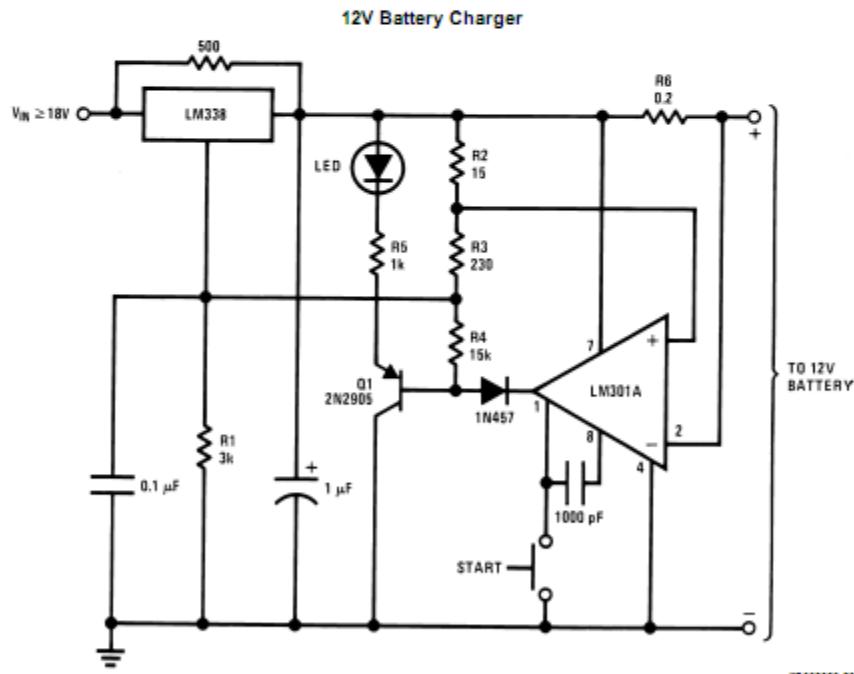


CS009050-11

Typical Applications (Continued)

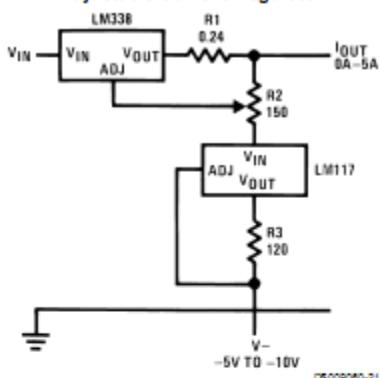


* R1 = 2400, R2 = 5k for LM138
Full output current not available
at high input-output voltages

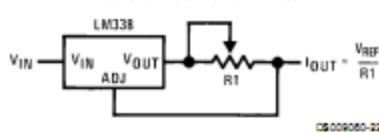


Typical Applications (Continued)

Adjustable Current Regulator

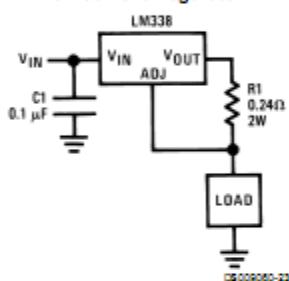


Precision Current Limiter



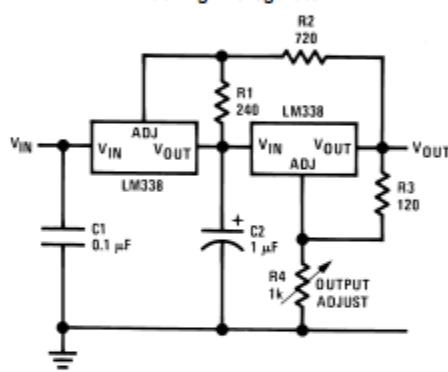
DS009080-22

5A Current Regulator



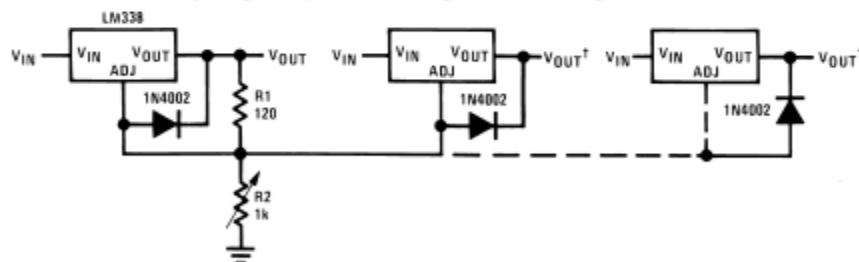
DS009080-23

Tracking Preregulator



DS009080-24

Adjusting Multiple On-Card Regulators with Single Control*

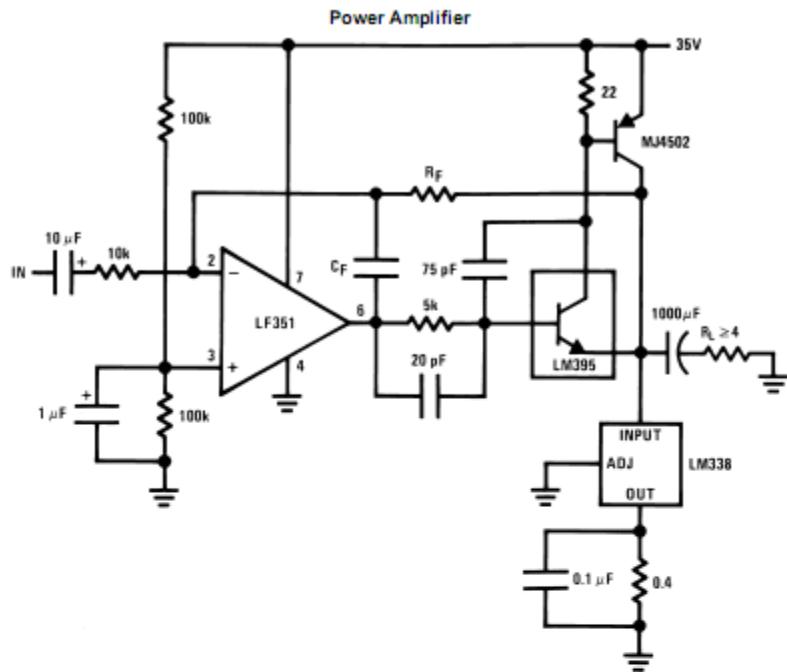


DS009080-25

* Minimum load — 10 mA.

* All outputs within ± 100 mV

Typical Applications (Continued)



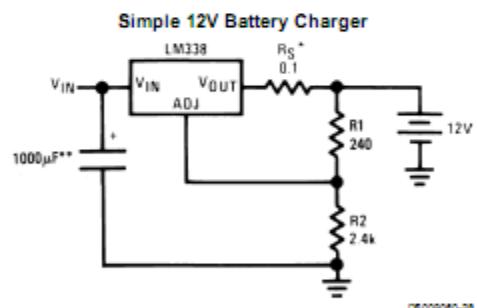
DS009050-27

$$A_V = 1, R_F = 10k, C_F = 100 \mu F$$

$$A_V = 10, R_F = 100k, C_F = 10 \mu F$$

Bandwidth ≥ 100 kHz

Distortion $\leq 0.1\%$

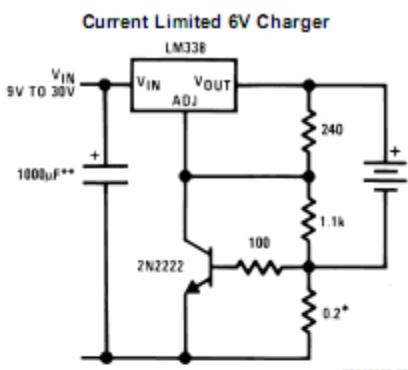
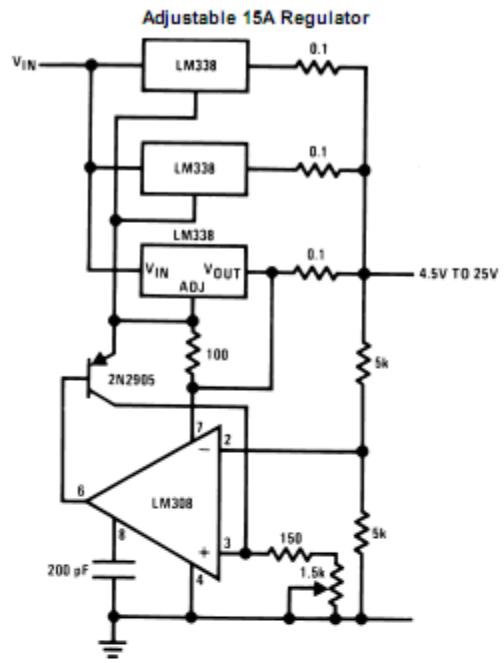


DS009050-28

$$*R_S \text{--- sets output impedance of charger } Z_{OUT} = R_S \left(1 + \frac{R_2}{R_1} \right)$$

Use of R_S allows low charging rates with fully charged battery.
**The 1000 μF is recommended to filter out input transients.

Typical Applications (Continued)

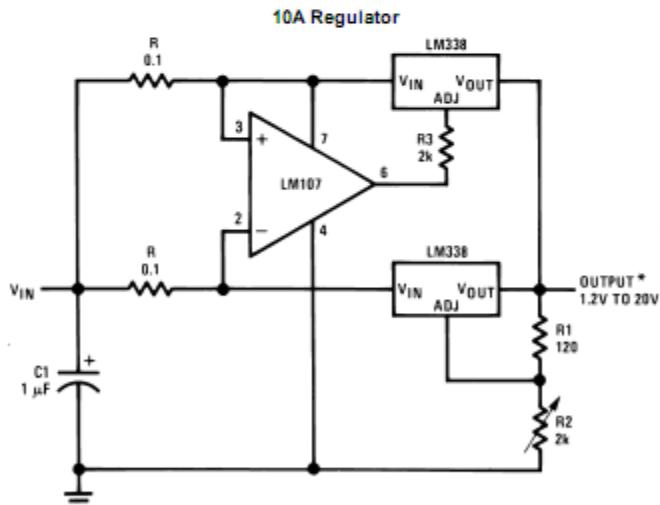


* Set max charge current to 3A

** THE 1000 μ F is recommended to filter out input transients.

DS009060-28

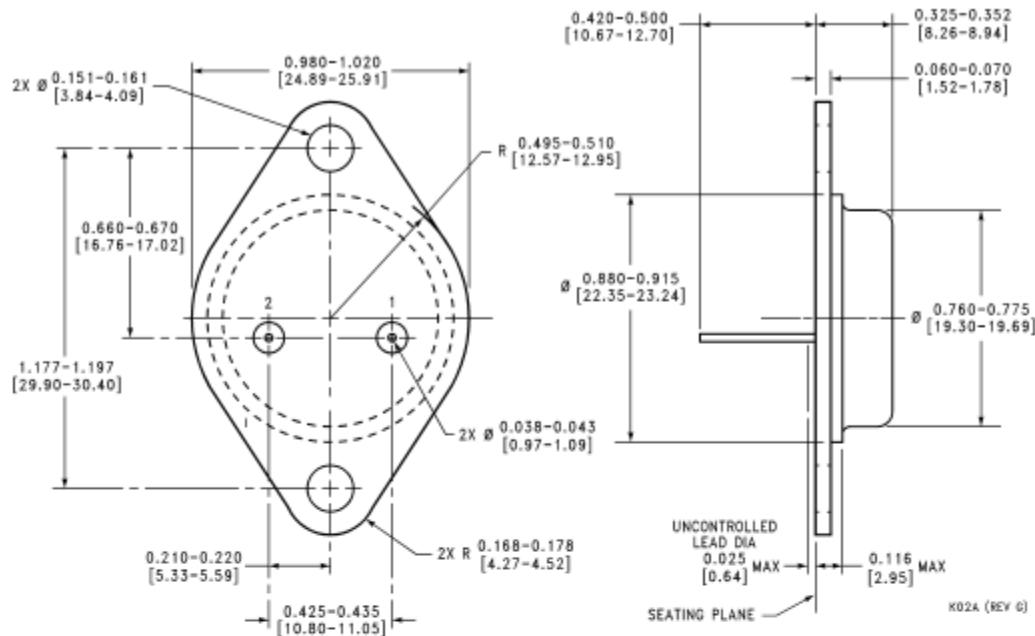
DS009060-29



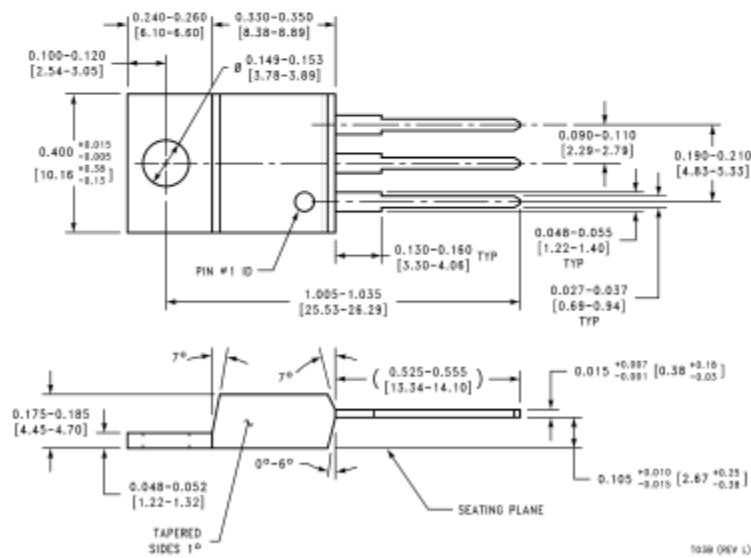
DS009060-3

* Minimum load = 100 mA

Physical Dimensions inches (millimeters) unless otherwise noted



2 Lead TO-3 Metal Can Package (K)
Order Number LM138K or LM338K STEEL
NS Package Number K02A



3 Lead Molded TO-220 (T)
Order Number LM338T
NS Package Number T03B

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.



National Semiconductor
Corporation
Americas
Tel 1-800-272-9959
Fax 1-800-737-7018
Email: support@nsc.com
www.national.com

National Semiconductor
Europe
Fax: +49 (0) 1 80-930 85 86
Email: europe.support@nsc.com
Deutsch Tel: +49 (0) 1 80-930 85 86
English Tel: +49 (0) 1 80-932 78 32
Français Tel: +49 (0) 1 80-932 93 58
Italiano Tel: +49 (0) 1 80-934 16 80

National Semiconductor
Asia Pacific Customer
Response Group
Tel 65-2544468
Fax 65-2604468
Email: apac.support@nsc.com

National Semiconductor
Japan Ltd.
Tel: 81-3-5639-7580
Fax: 81-3-5639-7507

National does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and National reserves the right at any time, without notice, to change said circuitry and specifications.

Appendix Q: IED H-Bridge Documentation

The IED H-Bridge Circuit Board (v.3)

Last Rev. 9/28/05 Stephen Rock & Matthew Rosmarin

You should thoroughly read and understand this document before attempting to use the IED H-Bridge Board. Else, you may find that fixing a damaged board is a miserable experience.

Introduction

A circuit known as an H-bridge can be used to achieve speed and direction control when using permanent magnet DC (direct-current) motors such as those common in toys, cell phones, and cars. (See: <http://electronics.howstuffworks.com/motor1.htm> for a good introduction on motors.)

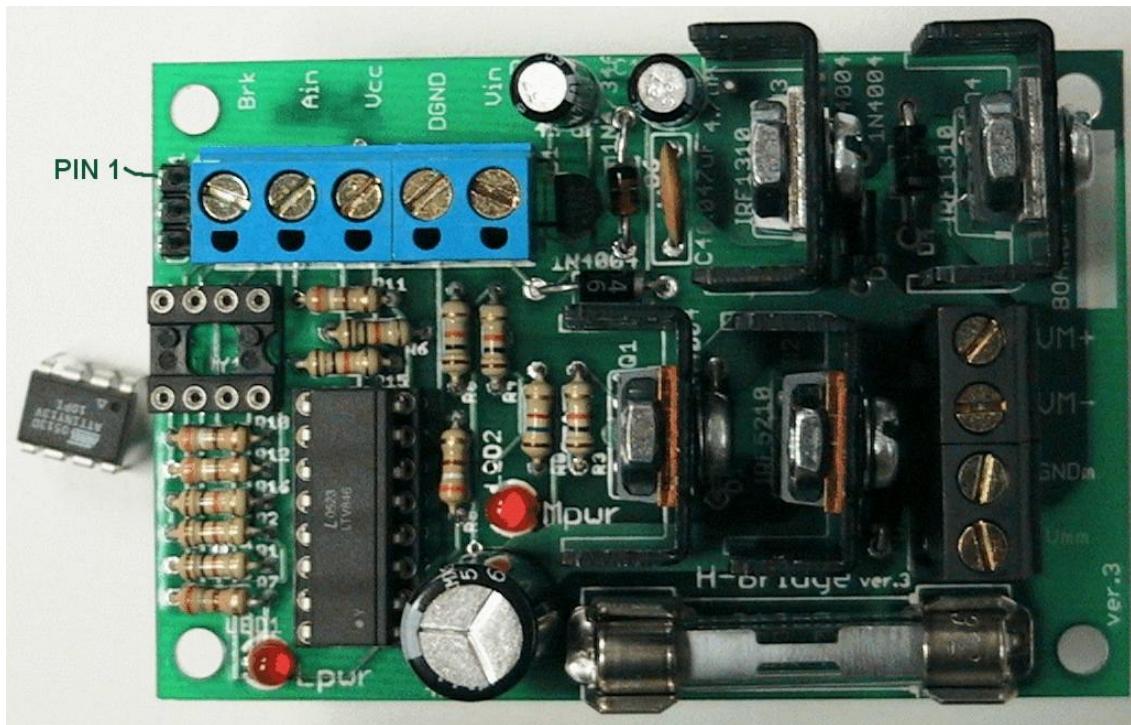
IED students have historically had a need to control a range of motor sizes, and since voltage ranges allowable in the class are limited to 24VDC, permanent magnet brushed DC motors are the usual choice for rotary motion. If on-off control is all that is required, a relay could be used. If forward-off-reverse control is required, one relay can be used to control the on-off function, and a second relay (or a DPDT, that's double pole double throw, switch) can be used to reverse the direction of the motor. (See the powerpoint slides on reserve in the library for more detail and beware that when an energized coil goes open circuit it can generate a large voltage spike capable of shocking you or destroying transistors that are not protected by a diode as shown in the slides.)

The same forward-off-reverse functionality can be achieved using a transistor-based H-bridge circuit. However, since the transistors can switch considerably faster than relays, they can also be switched in such a way that only partial power (on average) is applied to the motor. The motor is turned on for a fraction of a second, and then back off, and this is repeated thousands of times a second to achieve a particular speed. This is known as pulse width modulation, and more on this subject can be found at <http://www.embedded.com/story/OEG20010821S0096> or from a search of this term or "PWM" on your favorite web search engine.

For very small motors, it may be practical to use an integrated circuit chip such as the L293 (see: <http://www.st.com/stonline/products/literature/ds/1330/l293d.pdf> or <http://rocky.digikey.com/WebLib/Texas%20Instruments/Web%20data/L293,%20L293D.pdf>). However, these are limited to relatively low currents (max of 1 amp, or 2 amps if combined), which can easily be exceeded when a motor is under heavy load (or stalled – where the rotor is kept from spinning). Exceeding the current can destroy the driver chip, and this usually happens at the last possible minute when some unforeseen event occurs. (Hint: It is a very good idea to test the stall current of your motor before selecting a driver.)

Overview

We have developed an H-Bridge driver circuit board for IED that is capable of switching considerably higher currents than chips such as the L293. This circuit board is shown in the photo below (with one chip removed – more on this later).



IED H Bridge Board – shown with microcontroller removed

The IED H-Bridge board (HB, hereafter) has been designed to switch loads of 25Amps at 24VDC, but the installed fuse is 20A and this current should not be exceeded without consulting with your instructor and increasing the fuse amperage rating.

IMPORTANT: Under no circumstances should you bypass the fuse or replace it with one of a different type or capacity without your instructor's permission – you are responsible for all damage to the HB boards! (They are not fun to repair!) The gearmotor supplied for IED projects draws less than an ampere when free running, but it has a stall current of almost 10A. If you decide to use an alternate motor, be sure the STALL CURRENT does not exceed 20A. Likewise, if you will be attempting to quickly reverse large inertial loads, it is possible that you can even exceed the stall current of the motor because the load will cause the motor to act like a generator and resist the rapid direction change.

Additionally, it is important to note that the AC power supplies provided in IED for bench testing are limited at 6.5A and will fault (and remain off until they are power cycled) should this current be exceeded. This, too, serves to protect you and the HB board from destruction. Be aware that batteries can source *MUCH LARGER* currents, at least for a short time. Car batteries can source hundreds of amperes if the terminals are shorted (think arc welder, see: <http://www.safari4x4.com.au/80scool/tech/batweld/batweld.html>).

The high current 24VDC side of the HB is optically isolated from the logic side of the circuit for added protection. Only light (inside the large IC on the board) signals when the H bridge transistors should function – there is no electrical connection. Motor power is supplied with the positive terminal of a battery or power supply connected to the Vmm screw terminal. The negative terminal of the battery or power supply can be connected to the GNDm terminal. Vmm should never be at a voltage lower than GNDm (e.g. don't accidentally reverse the power supply or battery leads – convention in engineering is that red wire is positive and black wire is negative or in this case ground).

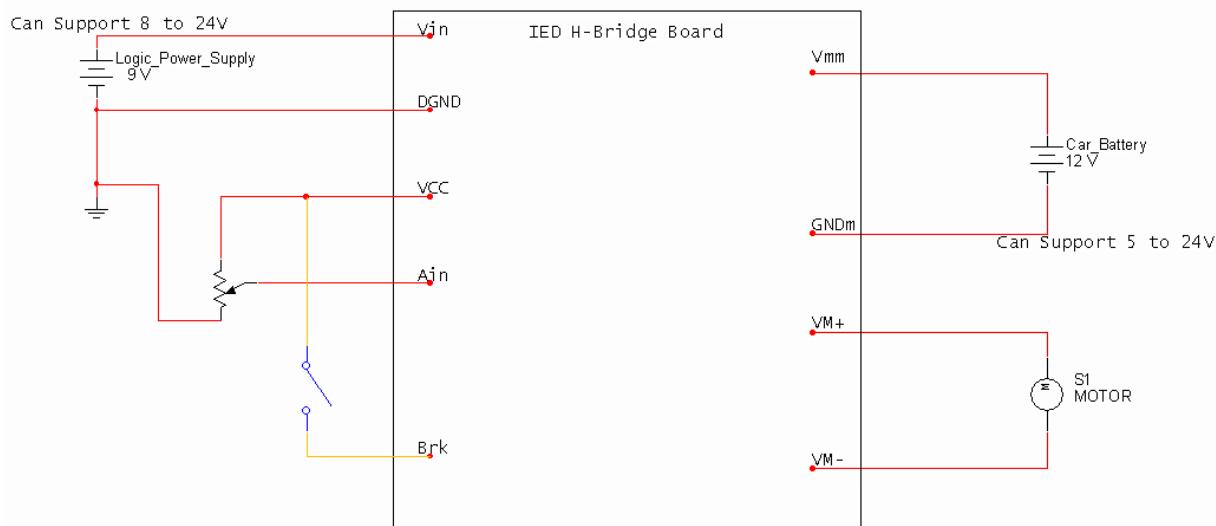
Note that the circuit board was designed so that one of the four screws (the one closest to a transistor heat sink) that mounts the board to the base plate connects with the GNDm signal. It should not be used to supply ground to your motor; however, you should take care to ensure this bolt does not come in contact with any other electrical signals or voltage sources.

Finally, the motor must be connected to terminals VM+ and VM-. If you connect the motor and then find you'd prefer forward and reverse to be swapped, simply swap the motor leads going to VM+ and VM-.

Modes of Operation

The IED HB can be used in three different ways. First, the board is equipped with a microcontroller that is pre-programmed to implement a PWM controller. This means that it accepts a DC voltage in and generates the PWM signals to control motor speed. This is the simplest method for using the HB. For those wishing to learn more about PWM, it is possible to remove the microcontroller (the 8 pin IC) and directly drive the optoisolator with TTL level (5V) logic signals. This is more complicated, as it is possible to present a combination of three signals that cause the bridge to short rail to rail (a direct short from Vmm to GNDm, not good, but if current limited by the lab supply to 6.5A, it will not destroy the bridge). Finally, it is possible to purchase an Atmel Attiny13 microcontroller of your own, program it, and substitute it on the board. This would also require considerable skill, and could prove limiting as there are few I/O pins on the microcontroller selected for this project. While this last option is not recommended, if you pursue this, please be sure to retain and protect the microcontroller furnished with the board (do not reprogram it!). You are responsible for returning a working HB at the end of the semester.

On-board PWM Generation (*the simplest option*)



Sample Circuit.
Switch Triggers Brake
Potentiometer controls pulsedwidth and direction

The low power side of the circuit contains a microcontroller chip (the small IC) that has been programmed to accept an analog signal between 0-5V and generate Pulse Width-Modulated (PWM) motor control signals for the H-bridge based on this voltage. The motor should be off when a command voltage of approximately 2.5V is given, at full speed in one direction at 0V and fully reversed at 5V. **It is VERY IMPORTANT that the analog input not go below 0V (a negative voltage relative to DGND) or exceed 5V.** If this input is not connected, a voltage divider has been added to the circuit to set this at approximately 2.5 volts; however, due to manufacturing tolerances in the resistors used to create this voltage divider, the motor may not be completely off (but will be close). This ensures that were the wire carrying the analog speed signal to the board to become disconnected, the motor would not be commanded at full speed as would be the case when a zero volt signal is forced on this input.

There is a voltage regulator on the board so that the logic and microcontroller can be powered by a source between 8-24VDC. For this we suggest a 9V battery, or similar voltage wall-wart DC power supply (beware that some are simply transformers producing AC voltage). You need a DC power supply. It should say xVDC on the supply, where x is the voltage, and it should be capable of delivering at least 100mA of current. Note that this is *different* from the 24V 6.5A supply provided for bench testing in IED and it is each team's responsibility to furnish this power supply or battery. In theory, the regulator supplied could be used with the 24V supply provided for powering the motor; however, it is bad practice to share the motor power with the digital logic because of switching noise and the possibility for microcontroller malfunction.

The motor command voltage can be created manually by using a potentiometer (a variable resistor, often abbreviated 'pot'). The maximum value of the pot will determine how much current it draws, so since current is limited, we should stick with a pot having a resistance of at least 10k ohms. A pot between 10k and 20k ohms would be a good choice. The center terminal of the pot sweeps from one side of the resistor to the other, so by connecting this to Ain, you've created a variable voltage divider than can be used to adjust motor speed and direction.

Alternatively, we could generate the analog motor speed command voltage for Ain using a computer with a digital to analog converter or an interface such as the Labjack. When doing this, it is very important to connect the ground from the labjack to the digital ground DGND on the HB because voltage is relative and the two grounds, if isolated, could be at different potentials (i.e. from electrical noise). If using a computer generated input to Ain, **it is critical that the programmed voltage not be allowed to exceed 5V**, and the output of any programmed device should be tested using a volt meter *prior* to connecting to the HB.

Please note that the regulated 5V supply on this board is of very limited capacity (100mA) and is provided on the terminal strip only for use by the board – it should not be taken from the board and used to power any additional hardware such as TTL integrated circuits or external indicator lamps or LEDs (light emitting diodes). This would quickly destroy the regulator IC on the HB board. Please construct your own 5V regulator if necessary by copying the design here or using a higher current regulator IC (see: <http://bit.ly/cL78ey>).

Last, but not least, one input on the microcontroller (labeled Brk on the logic terminal block) implements a brake for the motor by stopping generation of PWM signals and shorting the motor leads together. This causes what is known as "back EMF" (search on the term if curious) to make the motor slow down. It is not a true mechanical brake. However, braking action will be proportional to motor speed. This signal is active high, meaning that 5V (and no more) should

be applied to this screw terminal to cause the braking action. If you do not wish to use the brake feature, just leave it disconnected, there is no need to ground it as it is pulled low on board by a pull-down resistor. This signal can be generated using a contact closure (switch or relay) to Vcc, or it can be connected to a logic output port on a computer or Labjack. Again, be sure to connect a ground between the HB and whatever is generating the Brk signal.

Direct H-Bridge Control

It is also possible to directly control the H-bridge by removing the microcontroller chip (the only 8 pin DIP packaged IC on the board – shown removed in the first photo) and connecting to the three header pins next to the logic terminal strip. The header pins are denoted pins 1,2, and 3, with pin 1 labeled on the circuit board silk screening and oriented furthest from the microcontroller IC.

These pins are standard 0.100" spacing pins, and you can find compatible pins and connector bodies by visiting www.digikey.com and searching for the part number provided on the bill of materials given in the appendix of this document.

Before attempting to drive the bridge yourself, we recommend that you observe the signals on these three pins while the microcontroller is still connected and operating. It is possible to view the PWM signals generated by the microcontroller at these pins using an oscilloscope.

Of the eight (2^3) possible signal combinations on these pins, only four are safe combinations that correspond to coast (off), brake, forward, and reverse as shown in the table below.

Logic (at 3 header pins)			Optoisolator LEDs (as wired)				Inverted Opto LEDs				Corresponding Transistor State				Bridge Function	
Fwd	Rev	Enable	A	C	B	D	~A	~C	~B	~D	A	C	B	D		
0	0	0	0	0	0	0	1	1	1	1	0	0	1	1	Brake	
0	0	1	0	0	1	1	1	1	0	0	0	0	0	0	Coast	
0	1	0	1	0	0	0	0	1	1	1	1	0	1	1	X	
0	1	1	1	0	1	0	0	1	0	1	1	0	0	1	Reverse	
1	0	0	0	1	0	0	1	0	1	1	0	1	1	1	X	
1	0	1	0	1	0	1	1	0	1	0	0	1	1	0	Forward	
1	1	0	1	1	0	0	0	0	1	1	1	1	1	1	X	
1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	X	

Inputs: Logic 0 = 0V, Logic 1 = 5V; Transistor State 0 = not conducting, 1 = conducting

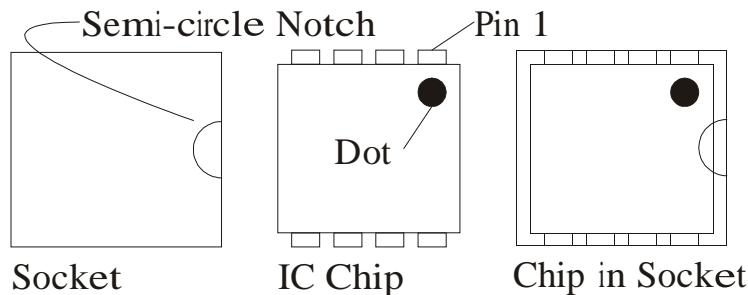
Note: Illegal bridge states denoted X short the bridge and should never be entered.

Truth table for H-bridge control signals and corresponding function.

The other illegal states will cause the bridge to short and care should be taken to ensure these states are never reached – even for very short durations. Aside from potential circuit damage, this would drain any batteries faster than otherwise necessary. When generating your own PWM signals, please be sure to use only a single power supply limited to 6.5A for initial testing. This way, if your programming or wiring does cause the bridge to enter an illegal state and short, it will fault the power supply and protect the bridge transistors from destruction. Incorrect operation when connected to a battery power source capable of delivering greater than 20 amps would likely have disastrous consequences.

If review of the schematic, attached program, and bill of materials (all in the appendix) is confusing, we recommend against attempting to directly drive the bridge. Please seek help *before* trying to implement your own PWM control if you are unsure about how this circuit works.

Before removing the microcontroller from the board, please note it's orientation. It appears "up side down" based on most board silk screen labeling and should be inserted in the socket so that the dot denoting pin 1 of the chip is toward the notched end of the IC socket as shown below.



Microcontroller Programming

For those who may be interested in replacing the microcontroller with one having their own program, the following is provided; however, this is not recommended and will be of limited utility given the low I/O count of the controller selected.

The Atmel Attiny13 can be programmed using the Atmel AVR programmer. This programmer comes with a complete development environment for a very reasonable single unit price of \$29/ea and is available at Digikey.com. A C compiler for various AVR microcontrollers is available for free (when used for non-commercial purposes) from:

<http://www.hpinfotech.ro/html/download.htm> and the code size limit does not impact use with the Attiny13 chip. The Atmel Attiny13 microcontroller in the HB was programmed in C using the program shown in the appendix. Please do not reprogram the microcontroller – purchase and work with your own if you wish to experiment with microcontroller programming on the HB board. Note that the HB does not facilitate in circuit programming of the microcontroller – the chip will have to be removed from the board for programming.

Physical Integration

The HB circuit boards are attached to a plastic backer and should not be removed. This serves three purposes. First, it protects the boards with live electrical connections exposed on the back from being inadvertently set on a wrench or other conductive surface (yes, this has been done in IED in the past with far more expensive circuit boards). Second, it provides a more durable point at which to attach the HB to your projects and minimizes risk that the printed circuit board of the HB itself becomes damaged (e.g. over-tightening a screw, or using a single screw and snapping a corner off the board). Finally, and most important for the continued success of your circuits in the 11th hour, it provides a means of strain relief for all cables going to the board. The photo below shows how small tie wraps can be used to secure wires going to the HB board, thus making them less likely to be torn out of the terminal strips.



Partial view of H-bridge board backer and strain reliefs

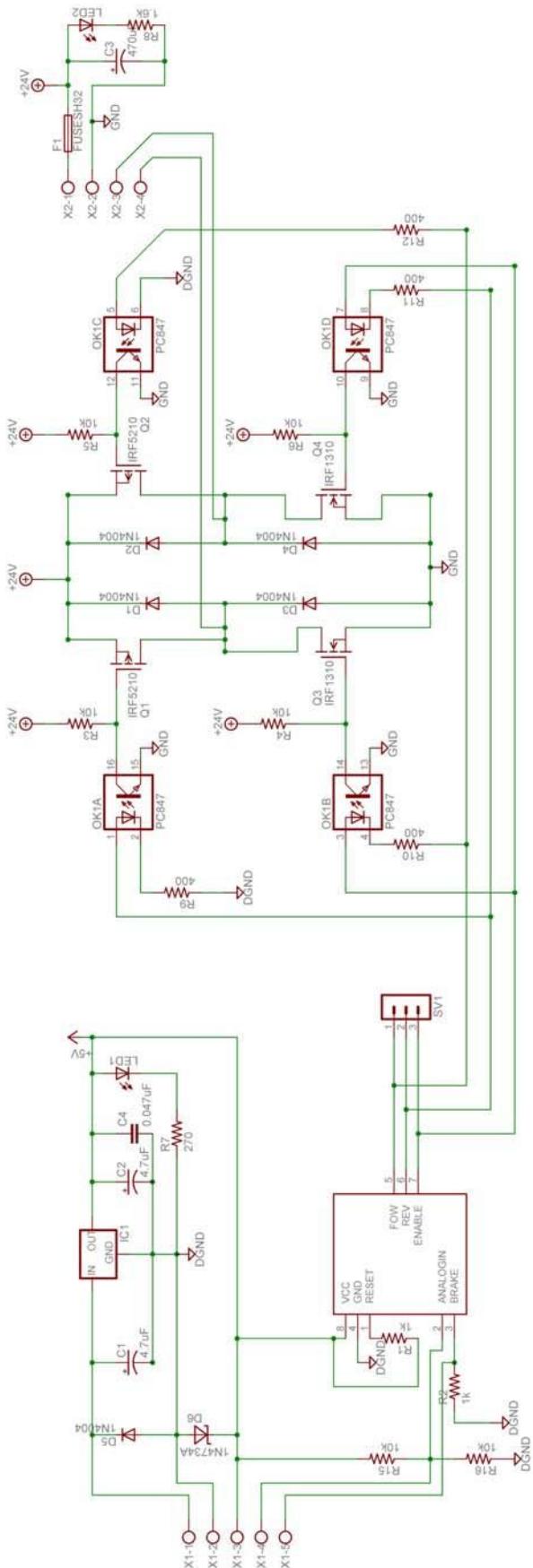
WARNINGS

1. The HB boards are sensitive to static electricity (e.g. in the dry air of winter, you take off a sweater, or scuff your feet across some carpet, then touch a refrigerator or other grounded surface and feel/see/hear a spark jump). To prevent damage to the board, please touch something grounded, such as a water faucet, or electrical conduit, before handling or working with your board when the relative humidity in the air is low (especially in the winter).
2. The components rising above the HB board are not made for rough handling. In particular, the heat sinks attached to the bridge transistors will easily catch and be destroyed by shoving this board in a backpack, etc. It is important to protect the board as though it's going to save you 10's of hours of work – a working board can do this. Please find a box or case to transport the board in and consider how you will protect the board once it is attached to your IED project.
3. The bridge transistor heat sinks are *not* electrically insulated and must not be touching one another or any other conductor. They can easily be folded into one another by less than careful handling of the board.
4. Heat sink temperature should be periodically monitored and if the motor will run under heavy load and draw close to the 25ampere design limit, it may be necessary to supply a forced air cooling fan to more efficiently dissipate heat from the transistors.

Appendix

- 1 – Circuit Schematic
- 2 – PWM Program
- 3 – PWM and Motor Voltage Example Plots
- 4 – Board Layout
- 5 – Bill of Materials
- 6 – So you want to make your own printed circuit board...

Appendix O1 – Circuit Schematic



Appendix O2 – PWM Program

```
*****  
IED H-Bridge PWM Firmware (v.3)  
By Matthew Rosmarin and Stephen Rock  
Last Updated on Sept 27, 2005  
*****  
  
//Include Statements  
#include <tiny13.h>  
#include <delay.h>  
#define ADC_VREF_TYPE 0x00  
*****  
  
// Analog to Digital Conversion Function  
// Converts 0V to +5V into a numeric value from 0 to 1024  
// "adc_input" is the ADC pin number you wish to convert  
// Use: read_adc(3); // would return the result for ADC pin 3  
unsigned int read_adc(unsigned char adc_input)  
{ ADMUX=adc_input|ADC_VREF_TYP  
E;  
// Start the AD conversion  
ADCSRA|=0x40;  
// Wait for the AD conversion to complete  
while ((ADCSRA & 0x10)==0);  
ADCSRA|=0x10;  
return ADCW;  
}  
*****  
  
// Start of program  
void main(void)  
{  
// Declare Variables  
int command;  
int pa,pb ;  
*****  
  
//Initialization Code (this code sets up the microcontroller)  
// Crystal Oscillator division factor: 1  
CLKPR=0x80;  
CLKPR=0x00;  
// Input/Output Ports initialization  
// Port B initialization  
// Func5=In Func4=In Func3=In Func2=Out Func1=Out Func0=Out  
// State5=T State4=T State3=T State2=0 Statel=0 State0=0  
PORTB=0x00;  
DDRB=0x07;  
// Timer/Counter 0 initialization  
// Clock source: System Clock  
// Clock value: 150.000 kHz  
// Mode: Phase correct PWM top=FFh  
// OC0A output: Non-Inverted PWM  
// OC0B output: Non-Inverted PWM  
TCCR0A=0xA1;  
TCCR0B=0x03;  
TCNT0=0x00;  
OCR0A=0x00;  
OCR0B=0x00;  
// External Interrupt(s) initialization  
// INT0: Off  
// Interrupt on any change on pins PCINT0-5: Off  
GIMSK=0x00;  
MCUCR=0x00;  
// Timer/Counter 0 Interrupt(s) initialization  
TIMSK0=0x00;  
// Analog Comparator initialization
```

```

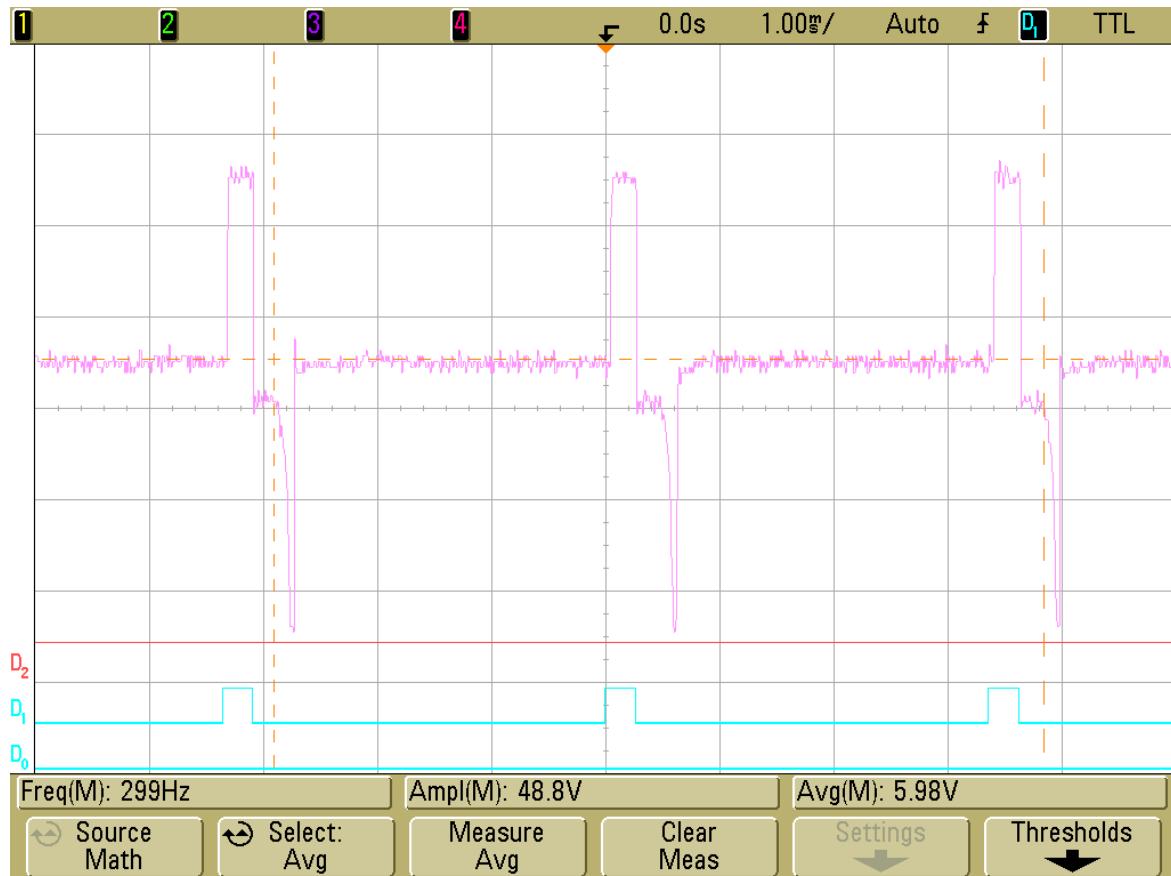
ACSR=0x80;
ADCSRB=0x00;
// ADC initialization
// ADC Clock frequency: 150.000 kHz
// ADC Bandgap Voltage Reference: Off
// ADC Auto Trigger Source: None
// Digital input buffers on ADC0: On, ADC1: On, ADC2: On, ADC3: On,
// ADC4: On
DIDR0=0x00;
ADMUX=ADC_VREF_TYPE;
ADCSRA=0x86;
//*********************************************************************
// End of Initialization code

// Main Program Loop (runs forever)
while (1)
{
    //Compute Desired Command Values
    command=read_adc(3);      // Get command value
    command=command-512;      // Scale Value
    command=command/2;
    //***** *****
    if(read_adc(2)>100) //check to activate break
    {
        pa=0x00;
        pb=0x00;
        OCR0A=0x00;
        OCR0B=0x00;
        delay_ms(8); //Ensures no illegal state while switching
        PORTB=0x00;
    }
    else //break mode not selected
    {
        if(command>0)          // Forward mode
        {
            PORTB=0x04;
            pa=command;
            pb=0x00;
        }
        else                    //Reverse mode
        {
            PORTB=0x04;
            pa=0x00;
            pb=-command;
        }
    }
    //Check to make sure output value is valid
    if(pa>255) pa=255;
    if(pa<0) pa=0;
    if(pb>255) pb=255;
    if(pb<0) pb=0;
    //***** *****
    //Set Output
    OCR0A=(char)pa;
    OCR0B=(char)pb;
    //***** *****
}
}

```

Appendix O3 – PWM and Motor Voltage Example Plots

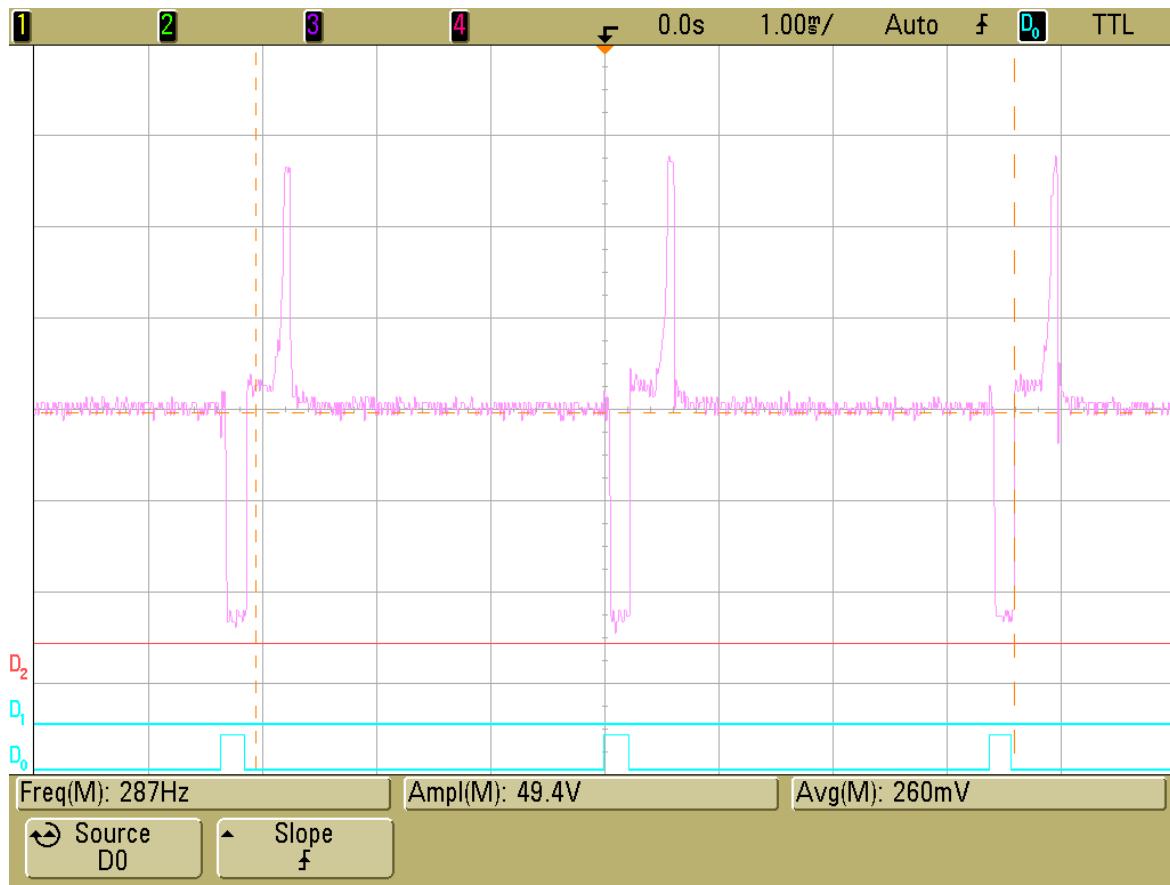
The following plots show the voltage across the motor on the top (analog) trace and also show the state of the three bridge inputs where D0=FWD, D1=REV, and D2=ENABLE. In this case the motor is being driven in “reverse” with respect to the bridge as the FWD line is logic low, the enable signal on D2 is high, and the pulses occur on the REV line shown by D1.



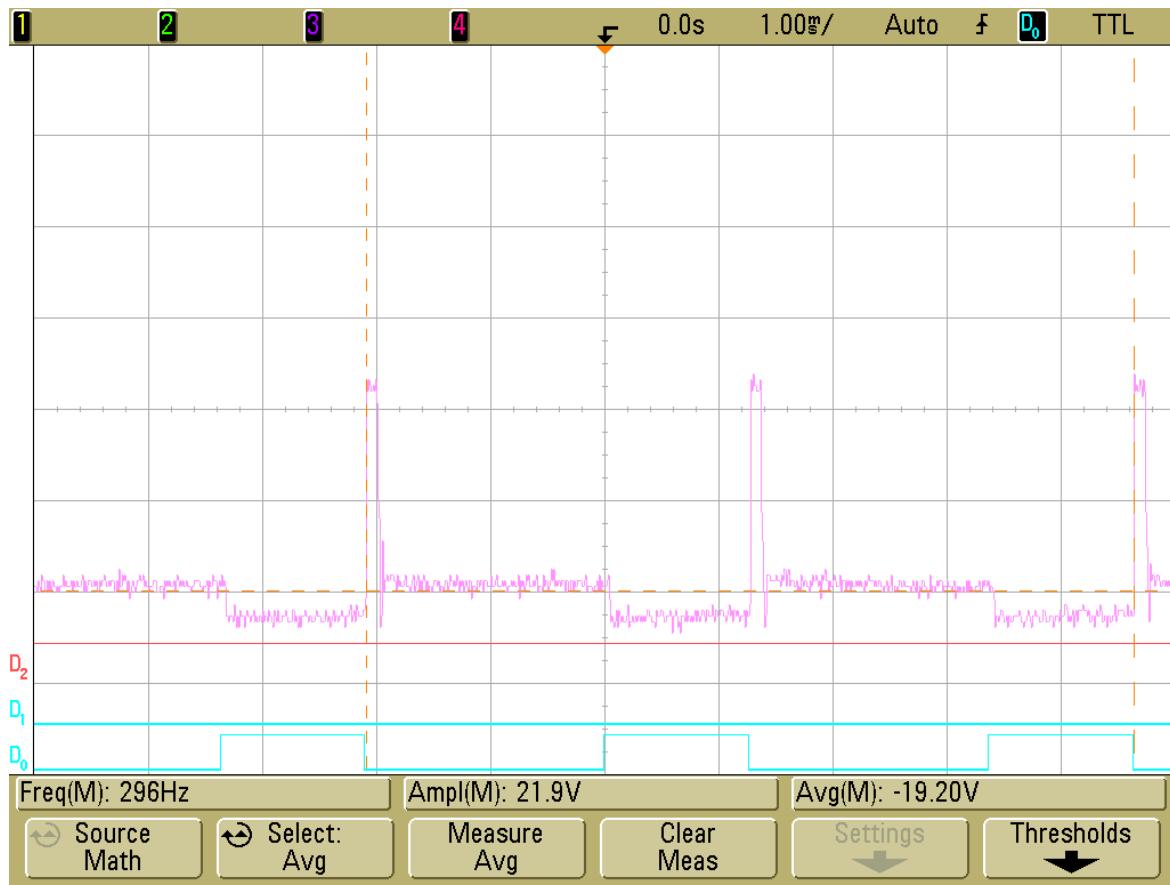
This plot shows a running motor (not stalled) that has some voltage above zero (the centerline of the plot) because the motor is free to spin and generates a voltage proportional to its speed. When the PWM signal to the bridge energizes the transistors, the voltage spikes up for the duration of the PWM pulse. The large negative going pulse observed after the bridge shuts off is due to back EMF in the motor. The motor voltage then returns to the level corresponding to its speed of rotation. Note that as the motor becomes more heavily loaded, the average voltage will decrease for a given PWM duty cycle.

In order to “control” speed, we would require feedback and a control system, so although our final motor speed is a function of our PWM duty cycle, it is also dependent on the motor loading. Feedback control would be required to maintain a precise speed under varying loads.

The following plot shows the motor operating with pulses on the FWD line. Note that the motor voltage is reversed, and that is why the waveform appears to be inverted.

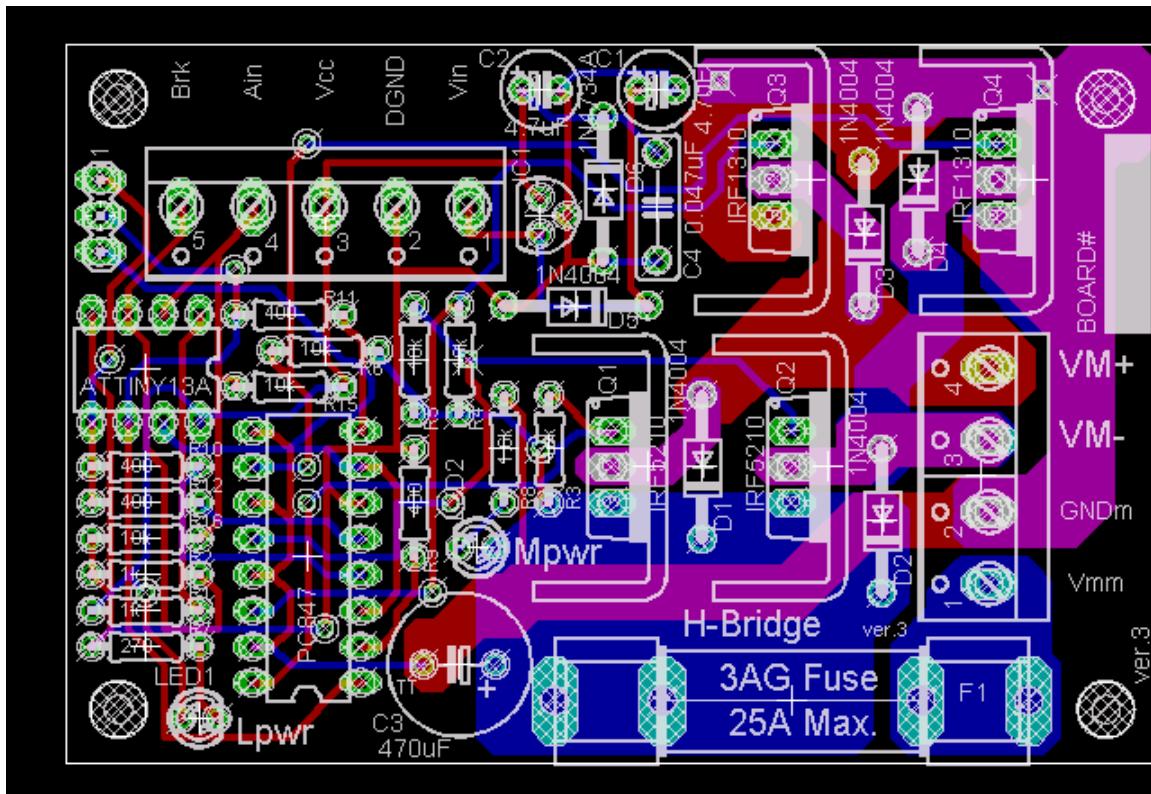


This plot shows an average voltage of only 0.260V and corresponds to a motor that just barely has enough power to rotate. If a higher speed were required, the duty cycle of the PWM pulses would be increased as shown in the following plot.

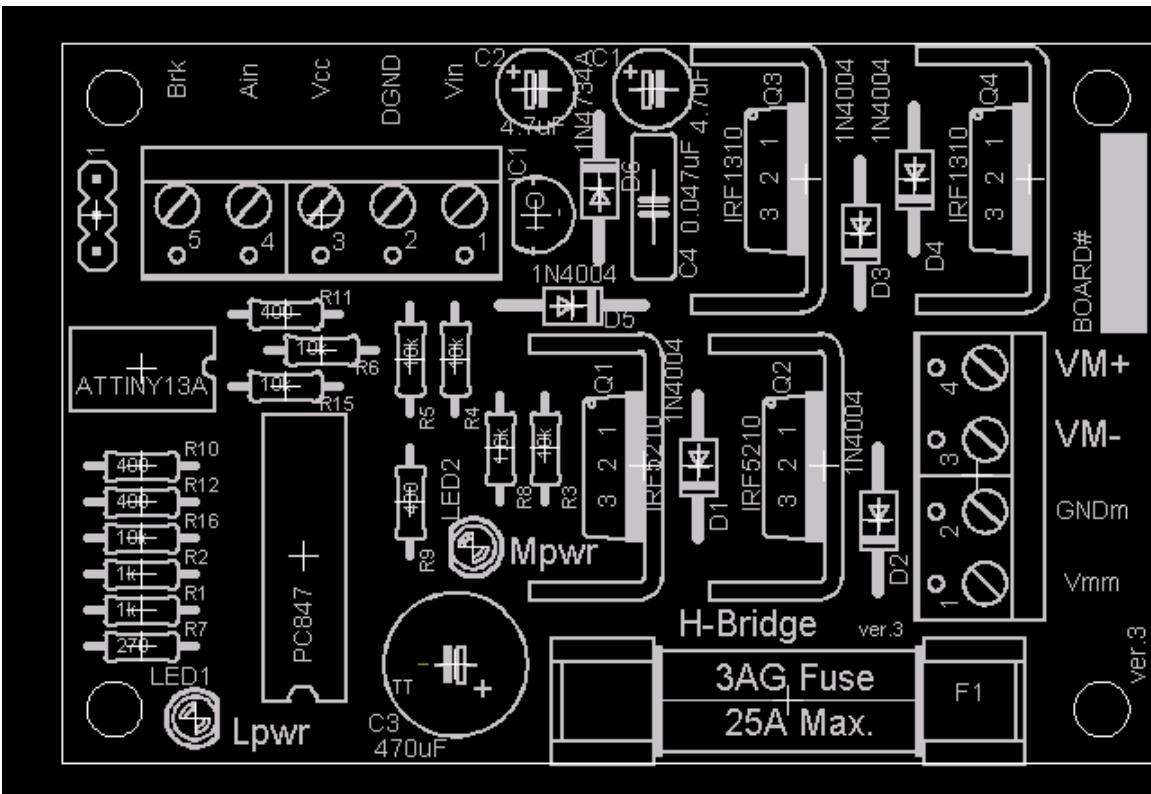


In this case the average voltage is considerably greater (in magnitude) because the motor is running. Also of note is that the PWM frequency is very near the constant 300Hz programmed, but it is the duty cycle, or length of time the pulse is on vs. off, that is changed to vary the speed of the motor.

Appendix O4 – Board Layout



Circuit board with top and bottom traces shown.



Circuit board silk screen only (for component ID and placement)

Appendix O5 – Bill of Materials

IED/Mechatronics H-Bridge Board Bill of Materials
Last Rev. 9/26/05

Line	Item	Manufacturer	Mfg. Part No.	Source	Source Part No.
1	HEX/MOS N-CH 100V 42A TO-220AB	IRF	IRF1310N	Digikey	IRF1310N-ND
2	HEX/MOS P-CH -100V -40A TO-220AB	IRF	IRF5210	Digikey	IRF5210-ND
3	Heatsink TO-220 Bolt On	Wakefield	273-AB	Digikey	345-1022-ND
4	SCREW MACHINE SLOTTED 6-32X1/4	Building Fastners	H154-ND	Digikey	PMS 632 0025 SL
5	NUT HEX 6-32 ZINC PLATED	Building Fastners	H220-ND	Digikey	HNZ632
6	#6 internal tooth lockwasher	Building Fastners	H240-ND	Digikey	INT LWZ 006
7	Diode	General Semiconductor	1N4004	Mouser	512-1N4004
8	Terminal block (motors)PCB 5.08MM 4POS	Weidmuller	999392	Digikey	281-1437-ND
9	FUSE 20A 32V 3AG FAST ACT	Littelfuse Inc	0312020.H	Mouser	576-0312020.H
10	CLIP FUSE 1/4" EAR PC MOUNT	Littelfuse	01020071H	Digikey	F040-ND
11	Quad optoisolator	LITE-ON INC	LTV-846	Digikey	160-1364-5-ND
12	Dip socket	Mill-Max Manufacturing	110-99-316-41-001000	Digikey	ED3116-ND
13	RES 270 OHM 1/4W 5% CARBON FILM	Yageo	CFR-25JB-270R	Digikey	270QBK-ND
14	RES 390 OHM 1/4W 5% CARBON FILM	Yageo	CFR-25JB-390R	Digikey	390QBK-ND
15	RES 1.0K OHM 1/4W 5% CARBON FILM	Yageo	CFR-25JB-1K0	Digikey	1.0KQBK-ND
16	RES 1.6K OHM 1/4W 5% CARBON FILM	Yageo	CFR-25JB-1K6	Digikey	1.6KQBK-ND
17	RES 10K OHM 1/4W 5% CARBON FILM	Yageo	CFR-25JB-10K	Digikey	10KQBK-ND
18	CAP 470UF 50V ELECT SMG RAD	United Chemi-Con	ESMG500ELL471MJ2	Digikey	565-1113-ND
19	Molex .100 K.K. Connectors 3pos	Molex	22-28-4030	Mouser	538-22-28-4030
20	STANDOFF HEX 6-32THR ALUM .250"	Keystone Electronics	2208	Digikey	2208K-ND
21	6-32x3/16 pan head mach screw	?		Fastenal	28781
22	6-32x5/16 pan head mach screw	?		Fastenal	28789
23	#6 nylon washers	?		Fastenal	76058
24	4" wire ties (for strain relief on backers)	?		Fastenal	63121
25	Terminal block (controller)			Digikey	ED2228-ND
26	Atmel ATtiny13 Microcontroller	Atmel	ATTINY13V-10PI-ND	Digikey	ATTINY13V-10PI-ND
27	DIP socket - 8 pin	Mill-Max Manufacturing	110-99-308-41-001000	Digikey	ED3108-ND
28	5V regulator (100mA LIMIT)	TI	UA78L05ACLP	Digikey	296-1365-ND
29	CAP 4.7UF 50V ELECT SMG RAD	United Chemi-Con	ESMG500ELL4R7ME	Digikey	565-1105-ND
30	.047 UFD 25V DISC CAP	Panasonic - ECG	ECK-F1E473ZVE	Digikey	P4307-ND
31	5.6V zener	Vishay	1N4734A	Mouser	78-1N4734A
32	LED 3.1MM 650NM RED DIFFUSED	Rohm	SLR-342VR3F	Digikey	511-1250-ND

Appendix O6 – So you want to make your own printed circuit board...

There are a variety of board houses that will turn printed circuit boards in very short times and at a far higher quality than one can ever hope to manually produce. We used Advanced Circuits for this project. They offer a service for prototypes that has an especially attractive price for students, see: www.barebonespcb.com and when it was time to order production boards (www.4pcb.com) they did an excellent job.

The circuit boards on which the HB is built have copper traces that are 3x the typical thickness (specified as 3oz copper per square foot, instead of the more typical 1oz copper board). This was necessary to carry the 25A maximum current for which the boards were designed.

These boards are 2 sided (no internal layers in the boards), have plated through holes, a solder mask, and silk screening.

We used Eagle schematic capture and board layout software from CadSoft (<http://www.cadsoft.de/freeware.htm>), which is free for non-commercial use on boards below a certain size. We downloaded the software on a Wednesday morning and had the first round of boards out for prototyping by Friday night. The software was very easy to learn (but it helped that we had previous experience with other software and the board design process). It took two additional iterations before we had all components properly defined and the final board design ready for a production order. The design process is often iterative.

Finally, this board was soldered using lead free solder. Kester (www.kester.com) recommended and generously supplied two lead free solders, both of which worked very well. To those experienced with soldering using conventional solder, the joints look like “cold joints” due to their dull appearance; however, this appearance is the norm for lead free solder. We’d recommend using lead-free solder for all future projects – it’s better for your health, better for the environment, and it’s the future.

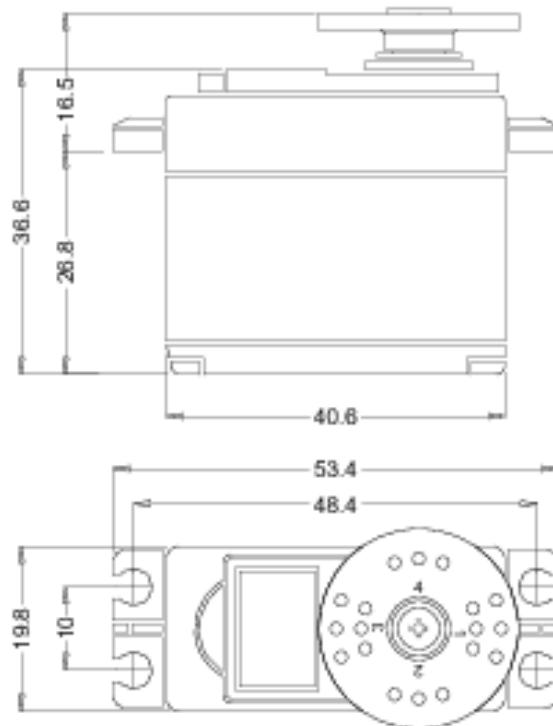
Good luck!

Appendix P: Servo Documentation

ANNOUNCED SPECIFICATION OF HS-422 STANDARD DELUXE SERVO

1.TECHNICAL VALUES

CONTROL SYSTEM	: +PULSE WIDTH CONTROL 1500usec NEUTRAL	
OPERATING VOLTAGE RANGE	: 4.8V TO 6.0V	
OPERATING TEMPERATURE RANGE	: -20 T0 +60°C	
TEST VOLTAGE	: AT 4.8V	AT 6.0V
OPERATING SPEED	: 0.21sec/60° AT NO LOAD	: 0.18sec/60° AT NO LOAD
STALL TORQUE	: 3.3kg.cm(45.82oz.in)	: 4.1kg.cm(58.93oz.in)
OPERATING ANGLE	: 45°ONE SIDE PULSE TRAVELING 400usec	
DIRECTION	: CLOCK WISE/PULSE TRAVELING 1500 TO 1900usec	
CURRENT DRAIN	: 8mA/IDLE AND 150mA/NO LOAD RUNNING	
DEAD BAND WIDTH	: 8usec	
CONNECTOR WIRE LENGTH	: 300mm(11.81in)	
DIMENSIONS	: 40.6x19.8x36.6mm(1.59x0.77x1.44in)	
WEIGHT	: 45.5g(1.6oz)	



2.FEATURES

- 3-POLE FERRITE MOTOR
- LONG LIFE POTENTIOMETER
- DUAL OILITE BUSHING
- INDIRECT POTENTIOMETER DRIVE

3.APPLICATIONS

- AIRCRAFT 20-60 SIZE
- 30 SIZE HELICOPTERS
- STEERING AND THROTTLE SERVO FOR CARS
- TRUCK AND BOATS

Appendix Q: Low Level Code

Matthew Williamson

Arduino Control Code:

```
/*
 * File:          control code.c
 * Description:  An program for controlling the IED H-Bridges and servos for our
Border Patrol Robot
 * Author:        Matt Williamson
 * Creation Date: April 15, 2010
 * Revision Date: April 27, 2010
 */

#include <Servo.h>

// Pins the 2 H-Bridges will be connected to
int PWML      = 3;
int PWMR      = 11;
int RBrake    = 5;
int LBrake    = 6;

//Variables for the 2 camera Servos
Servo updown;
Servo leftright;

//These define the maximum value that a pulselength can be
int minPulseL = 0;
int maxPulseL = 255;
int minPulseR = 0;
int maxPulseR = 255;
int PulseL     = 127;
int PulseR     = 127;
int angleLR   = 89;
int angleUD   = 89;

char moveServo;           // raw user input

void setup() {
    // Set up motor control pins as an output
    pinMode(PWML, OUTPUT);
    pinMode(PWMR, OUTPUT);
    pinMode(RBrake, OUTPUT);
    pinMode(LBrake, OUTPUT);

    //Attach the camera servos on pins 2 and 3
    updown.attach(4);
    leftright.attach(2);

    //sets timer to make PWM frequency 3906.25 Hz
    //TCCR2B = TCCR2B & 0b11111000 | 0x02;

    //Sets the initial position of the camera to be centered
    updown.write(angleUD);
    leftright.write(angleLR);

    //set up the PWM counters
    TCCR2A = _BV(COM2A1) | _BV(COM2B1) | _BV(WGM21) | _BV(WGM20);
    TCCR2B = _BV(CS22);

    // begin serial communication
    Serial.begin(9600);
}
```

```

void loop(){
    // read the incoming byte:
    moveServo = Serial.read();
    if (moveServo == 'a') {
        angleLR = angleLR - 10; }

    // move camera right
    else if (moveServo == 'd') {
        angleLR = angleLR + 10; }

    // move camera up
    else if (moveServo == 'w') {
        angleUD = angleUD - 10; }

    // move camera down
    else if (moveServo == 's') {
        angleUD = angleUD + 10; }

    // center the camera
    else if (moveServo == 'c') {
        angleUD = 89;
        angleLR = 89; }

    //drive the robot forward
    else if (moveServo == 'i') {
        PulseL = 255;
        PulseR = 0;
        digitalWrite(LBrake, LOW);
        digitalWrite(RBrake, LOW); }

    // turn right
    else if (moveServo == 'j') {
        PulseL = 255;
        PulseR = 134;
        digitalWrite(RBrake, HIGH);
        digitalWrite(LBrake, LOW); }

    // reverse
    else if (moveServo == 'k') {
        PulseL = 0;
        PulseR = 255;
        digitalWrite(LBrake, LOW);
        digitalWrite(RBrake, LOW); }

    // turn left
    else if (moveServo == 'l') {
        PulseR = 0;
        PulseL = 134;
        digitalWrite(LBrake, HIGH);
        digitalWrite(RBrake, LOW); }

    // neutral
    else if (moveServo == 'n') {
        PulseL = 127;
        PulseR = 127; }

    //ensure the camera servo angle is valid
    //and correct it if it is not
    if(angleUD < 0){
        angleUD = 0; }
    if(angleUD > 134){
        angleUD = 134; }
    if(angleLR < 0){
```

```

        angleLR = 0; }
if(angleLR > 179){
    angleLR = 179; }

        //send the angle value to the camera servos
updown.write(angleUD);
leftright.write(angleLR);

        //Set the pulsedwidth for the Motors
OCR2A = PulseR;
OCR2B = PulseL;
}

```

Appendix R: High Level Code

Lucas Doyle

File: Template.php

Description:

This file is an example of a basic web page written within the robot's framework and consistent with the robot's visual theme. Its short length yet high degree of functionality is indicative of the ease and modularity of programming in the architecture implemented by the team.

```

<html>
    <?php
        include_once("php/header.php");
        include_once("php/navigation.php");
    ?>

    <!-- MAIN -->
    <td valign=top bgcolor=D8D8D8>
        <p>Hello World!!</p>
    </td>

    <?php
        include_once("php/footer.php")
    ?>
</html>

```

End template.php

File: Header.php

Description:

This is a utility file that displays the top part of the website every time it is included in the body of a template.

```

<?php

// Include the stylesheet and JavaScript
include_once("css_javascript.php");

?>

<body width=1000em background="img/stripe.png">
<table align=center cellpadding=10 border=14 cellspacing=0 width=83% bgcolor=ffffff
class="ui-corner-all">
    <tr>
        <td align=center border = 4 bgcolor="555555">
            <a href="index.php">
                

```

```

        </a>
    </td>
<td width=80% border = 4 align=center valign=center bgcolor="#555555">
    <b>
        <div class="ui-widget">
            <div class="ui-state-error ui-corner-all" style="padding: 0 .7em;">
                <h1>Border Patrol Robot</h1>
            </div>
        </div>
    </b>
</td>
</tr>

```

End Header.php

File: css_javascript.php

Description:

Pulls in jQuery, the jQuery UI, and the jQuery CSS used to define how all elements of the webpage look. Additionally, this is where the behavior of various elements (Buttons, menu, icons, pop up windows, etc) are defined.

```

<head>
    <title>IED Border Patrol Robot</title>
    <!-- Include the jQuery generated CSS -->
    <link type="text/css" href="css/ui-darkness/jquery-ui-1.8.custom.css"
rel="stylesheet" />
    <!-- Include the jQuery library itself -->
    <script type="text/javascript" src="js/jquery-1.4.2.min.js"></script>
    <!-- Include the jQuery UI library -->
    <script type="text/javascript" src="js/jquery-ui-1.8.custom.min.js"></script>
    <script type="text/javascript">
        $(function() {
            // Here is where the elements that require behavior initializations
            // have their rules defined

            // Accordion
            $("#accordion").accordion({ header: "h3",
                                         animated: "bounceslide",
                                         navigation: true,
                                         autoHeight: true
                                         });

            // Tabs
            $('#tabs').tabs();

            // Dialog
            $('#dialog').dialog({
                autoOpen: false,
                width: 640,
                buttons: {
                    "Close": function() {
                        $(this).dialog("close");
                    }
                }
            });

            // Dialog Link
            $('#dialog_link').click(function() {
                $('#dialog').dialog('open');
                return false;
            });
        });
    </script>

```

```

// Dialog 2
$('#dialog2').dialog({
    autoOpen: false,
    width: 640,
    buttons: {
        "Close": function() {
            $(this).dialog("close");
        }
    }
});

// Dialog Link
$('#dialog_link2').click(function(){
    $('#dialog2').dialog('open');
    return false;
});

// Datepicker
$('#datepicker').datepicker({
    inline: true
});
$('#datepicker2').datepicker({
    inline: true
});

// Slider
$('#slider').slider({
    range: true,
    values: [17, 67]
});

// Progressbar
$("#progressbar").progressbar({
    value: 20
});

//hover states on the static widgets
$('#dialog_link, ul#icons li').hover(
    function() { $(this).addClass('ui-state-hover'); },
    function() { $(this).removeClass('ui-state-hover'); }
);

});
</script>

<style type="text/css">
    .fg-button { outline: 0; margin: 0 4px 0 0; padding: .4em 6.5em; text-decoration:none !important; cursor:pointer; position: relative; text-align: center; zoom: 1; }
    .fg-button .ui-icon { position: absolute; top: 50%; margin-top: -8px; left: 50%; margin-left: -8px; }

    a.fg-button { float:left; }

    /* remove extra button width in IE */
    button.fg-button { width:auto; overflow:visible; }

    .fg-button-icon-left { padding-left: 2.1em; }
    .fg-button-icon-right { padding-right: 2.1em; }
    .fg-button-icon-left .ui-icon { right: auto; left: .2em; margin-left: 0; }
    .fg-button-icon-right .ui-icon { left: auto; right: .2em; margin-left: 0; }

```

```

/* solo icon buttons must have block properties for the text-indent to work */
.fg-button-icon-solo { display:block; width:20px; height:20px; padding: 1em 1em;
text-indent: -9999px; }

.fg-buttonset { float:left; }
.fg-buttonset .fg-button { float: left; }
.fg-buttonset-single .fg-button,
.fg-buttonset-multi .fg-button { margin-right: -1px; }

.fg-toolbar { padding: .5em; margin: 0; }
.fg-toolbar .fg-buttonset { margin-right:1.5em; padding-left: 1px; }
.fg-toolbar .fg-button { font-size: 1em; }

h2 { clear: both; padding-top:1.5em; margin-top:0; }
.strike { text-decoration: line-through; }
</style>

<script type="text/javascript">
$(function(){
    //all hover and click logic for buttons
    $(".fg-button:not(.ui-state-disabled)").hover(
        function(){
            $(this).addClass("ui-state-hover");
        },
        function(){
            $(this).removeClass("ui-state-hover");
        }
    )
    .mousedown(function(){
        $(this).parents('.fg-buttonset-single:first').find(".fg-button.ui-state-active").removeClass("ui-state-active");
        if( $(this).is('.ui-state-active.fg-button-toggleable, .fg-buttonset-multi .ui-state-active') ){ $(this).removeClass("ui-state-active"); }
        else { $(this).addClass("ui-state-active"); }
    })
    .mouseup(function(){
        if( !$(this).is('.fg-button-toggleable, .fg-buttonset-single .fg-button, .fg-buttonset-multi .fg-button') ){
            $(this).removeClass("ui-state-active");
        }
    });
});
</script>

<style type="text/css">
body{ font: 62.5% "Trebuchet MS", sans-serif; margin: 50px; }
.demoHeaders { margin-top: 2em; }
#dialog_link {padding: .4em 1em .4em 20px;text-decoration: none;position: relative;}
#dialog_link span.ui-icon {margin: 0 5px 0 0;position: absolute;left: .2em;top: 50%;margin-top: -8px;}
ul#icons {margin: 0; padding: 0; }
ul#icons li {margin: 2px; position: relative; padding: 4px 0; cursor: pointer;float: left; list-style: none; }
ul#icons span.ui-icon {float: left; margin: 0 4px; }
</style>
</head>

End css_javascript.php

```

File: navigation.php

Description:

Utility file that generates the accordion menu on the left side of the website that serves as the primary mode of navigation around the web site.

```
<tr valign=top><td valign=top bgcolor="#555555">
<!-- nav bar -->
<div id="accordion">
  <div>
    <h3><a href="#">Home</a></h3>
    <div>
      <a href="index.php">Welcome to the Robot's control panel</a>
    </div>
  </div>
  <div>
    <h3><a href="#">Drive Control</a></h3>
    <div>
      <a href="drive.php">Click here to drive the robot around and change drive settings</a>
    </div>
  </div>
  <div>
    <h3><a href="#">Video Feed</a></h3>
    <div>
      <a href="camera.php">View a live camera feed of what the bot sees</a>
    </div>
  </div>
  <div>
    <h3><a href="#">GPS Information</a></h3>
    <div>
      <a href="gps.php">Click here to view information about the robots location</a>
    </div>
  </div>
  <div>
    <h3><a href="#">Query Database</a></h3>
    <div>
      <a href="search.php">Click here to search database to discover trends</a>
    </div>
  </div>
  <div>
    <h3><a href="#">Administration</a></h3>
    <div>
      <a href="vnc.php">Click here to view the robot's desktop</a>
    </div>
  </div>
  <div>
    <h3><a href="#">Help</a></h3>
    <div>
      <a href="help.php">Click here to read documentation and help files</a>
    </div>
  </div>
</div>
<br><br>
<a href="command.php" class="fg-button ui-state-default ui-corner-all">Dashboard</a>
</td>
```

End navigation.php

File: footer.php

Description:

This is a utility file that generates the bottom of the page.

```

</tr>
<tr>
    <th colspan="2" valign="center" bgcolor=A4A4A4><h6>Content, unless otherwise noted, is
    &copy; 2010 Lucas Doyle, Matt Williamson, Mike Kessler, Pete Willette, Mike Vo, Chris
    Vena and Chris O'neil</h6>
    <strong><h4>This website is incompatible with Internet Explorer. <a
    href="http://www.google.com/chrome/">Download a better, faster
    browser.</a></h4></strong></th>
</tr>
</table>
</body>

```

End footer.php

File: livegps.php

Description:

The live GPS module -- This page repeatedly displays a map of the current robot's location in the center. It is designed to be used inside of an iframe on a higher level page.

```

<html>
<head>
    <!-- Refresh the map every 5 seconds -->
    <META HTTP-EQUIV="REFRESH" CONTENT="5">
    <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
    <!-- Include the Google Maps API. That long list of characters is the API key
    issued to the group by Google in order to be able to use the maps service -->
    <script
src="http://maps.google.com/maps?file=api&v=2&sensor=false&key=ABQIAAAA8uV
EzOEqvussyxO7iaunhBQBQLbvr-JTDXWowqsj3EbfbZ8-fBQH4BuD6AjCEly-VKDQXvuOHf6Oyw"></script>

<?php

// declare some size variables to control how big the map is
$sizeX = 545;
$sizeY = 320;

?>

<script type="text/javascript">
    function initialize() {
        // if the browser is compatible, don't bother with anything
        if (GBrowserIsCompatible()) {
            // declare a map object that is the correct size
            var map = new GMap2(document.getElementById("map_canvas"), { size: new
GSize(<?=$sizeX?>, <?=$sizeY?>) });

        <?php
        // This is dynamic PHP code that changes the source of the HTML
        // every time the page is loaded.

        // Include the GPS parser. This allows the variables
        // $lat (latitude) and $long (longitude) to come into
        // scope.
        require('readfromgps.php');

        // Set the map's center to the current latitude and longitude
        echo 'var CENTER_LAT = '.$lat.';';
        echo "\n";
        echo 'var CENTER_LONG = '.$long.';';
    }
?>

```

```

        // actually set the center in JavaScript
        map.setCenter(new GLatLng(CENTER_LAT, CENTER_LONG), 18);

        // Add controls to the map, and set it to hybrid mode
        // which makes both streets and satellite images visible
        map.addControl(new GSmallMapControl());
        map.addControl(new GMapTypeControl());
        map.setMapType(G_HYBRID_MAP);

        // Add a point to the center of the map, which corresponds
        // to the robots location
        var latlng = new GLatLng(CENTER_LAT, CENTER_LONG);
        map.addOverlay(new GMarker(latlng));
    }
}
</script>
</head>
<body onload="initialize()" onunload="GUnload()" style="font: 100% 'Trebuchet MS',
sans-serif; margin: 0px;" align=center>
<?php
    // More dynamic PHP. If latitude and longitude are null,
    // it means that the parser was not able to extract
    // anything valid from the GPS, therefore there is no GPS fix
    if ($lat == 0 || $long == 0)
    {
        // output the message to the user
        echo '<h1>Sorry, there is no GPS fix</h1>';
        echo '<h1>Please wait</h1>';
    }
    else
    {
        // otherwise display the map
        echo '<center>';
        echo '<div id="map_canvas" style="width: '.$sizex.'px; height:
'. $sizey .'px;"></div>';
        echo '</center>';
        include_once( 'gpscontrols.php' );
    }
?>

</body>
</html>

```

End livegps.php

File: readfromgps.php

Description:

The GPS NMEA parser module - what makes the GPS usable by the rest of the PHP here. It takes raw NMEA sentences from the GPS and extracts latitude and longitude for later use.

```

<?php
    // This is the NMEA parser. Not the most efficient parser, but writing
    // it all in PHP was definitely worth not having to deal with
    // interfacing with an external parser.

    // Require the PHP serial class, as
    // it is needed to communicate with the GPS.
    require("php_serial.class.php");

    // Create a new serial object from the PHP serial class.

```

```

$serial = new phpSerial();

// Set all the parameters of the connection appropriate to the GPS.
// These are all specified by the GPS manufacturer.

// GPS is located in this USB port always.
$serial->deviceSet("/dev/ttyUSB1");

// Baud rate: 4800
$serial->confBaudRate(4800);

// Parity (this is the "N" in "8-N-1")
$serial->confParity("none");

// Character length (this is the "8" in "8-N-1")
$serial->confCharacterLength(8);

// Stop bits (this is the "1" in "8-N-1")
$serial->confStopBits(1);

// Finally, open the connection:
$serial->deviceOpen();

// This is the getline function for the GPS. It returns one valid
// GPRMC sentence from the serial connection. An important thing to
// keep in mind when looking at this function is that it is asynchrounously
// communicating with the GPS, so tactics had to be employed to ensure
// a valid return value.
function gps_getline( & $serial )
{
    // $line is an initially empty string that
    // is added to as characters are read.
    $line = '';

    // $j represents the number of valid sentences
    // the parser has been through.
    $j = 0;

    // The function quits trying to read a line after 10000 attempts
    // at trying to read information from the GPS.
    // This is necessary due to the asynchrounous nature of the connection
    while ($i < 100000)
    {
        // Try to read a character from the port
        $read = $serial->readPort();

        // If that character is a $, then it
        // is the start of a valid NMEA sentence
        if ($read == '$')
        {
            // this needs to be more then the second line it has read in.
            // without this, the function can append useless data to the line
            if ( $j > 1 )
            {
                // NMEA sentences are comma seperated variables.
                // The GPRMC sentence contains a specific amount of commas
                // so we can partially determine validity by counting them
                $commas = 0;
                for ( $k = 0; $k < strlen($line); $k += 1 )
                    if ( $line[$k] == ',' )
                        $commas += 1;

                // if there are the correct number of commas and the

```

```

        // first characters of the sentence are GPRMC, we have a match
        if ( $commas == 11 && substr( $line, 1, 5 ) == "GPRMC" )
            return $line;
    }

    // regardless, if the character is the start of a NMEA sentence,
    // and the above code didn't return the line, the line should be
    // cleared in an attempt to find another, valid sentence. Also,
    // j needs to be incremented.
    $line = '';
    $j += 1;
}

// regardless of everything, append whatever was read (keep in mind,
// this could be null) and increment i.
$line .= $read;
$i += 1;
}

// A function that takes in a string of comma seperated variables
// and returns an array of the variables. The function was taken
// by an example presented by user "Jrg Wagner" from PHP's online
// manual here: http://php.net/manual/en/function.explode.php
function csv_explode($delim=',', $str, $enclose=''', $preserve=false)
{
    $resArr = array();
    $n = 0;
    $expEncArr = explode($enclose, $str);
    foreach($expEncArr as $EncItem)
    {
        if($n++%2)
        {
            array_push($resArr,
array_pop($resArr).($preserve?$enclose:'')).$EncItem.($preserve?$enclose:''));
        }
        else
        {
            $expDelArr = explode($delim, $EncItem);
            array_push($resArr, array_pop($resArr) . array_shift($expDelArr));
            $resArr = array_merge($resArr, $expDelArr);
        }
    }
    return $resArr;
}

// Finally get to the acutal parsing. Retreive a line from the GPS.
$command = gps_getline( $serial );
// Explode the command, separating it into chuncks
$gpsarr = csv_explode(',', $command);

// pull the "raw" variables from the array. See the GPRMC sentence structure
// if the 3,4,5,6 doesnt make sense. They are called raw variables
// because they are still in minutes format instead of decimal.
$rawlat = $gpsarr[3];
$latdir = $gpsarr[4];
$rawlong = $gpsarr[5];
$longdir = $gpsarr[6];

// Convert latitude and longitude into decimal format.
// also negate if measurements were taken in the appropriate
// hemisphere.
$lat1 = floatval(substr( $rawlat, 0, 2));

```

```

$lat2 = floatval(substr($rawlat, 2, strlen($rawlat)));
$lat = $lat1 + ($lat2/60);
$lat = floatval(substr($lat, 0, 7));
if ($latdir == 'S')
    $lat = -$lat;

// Do the same thing with longitude
$long1 = floatval(substr($rawlong, 1, 2));
$long2 = floatval(substr($rawlong, 3, strlen($rawlong)));
$long = $long1 + ($long2/60);
$long = floatval(substr($long, 0, 7));
if ($longdir == 'W')
    $long = -$long;

// Close the serial connection
$serial->deviceClose();

// Record the date and time for later use by the database if the
// coordinate is going to be saved.
$date = (date("m/d/Y"));
$time = (date("H:i"));

?>

```

End readfromgps.php

File: gpscontrols.php

Description:

This is an HTML form displayed after the parser and live GPS modules are called. It shows date, time, latitude and longitude of the robot. When the big "person spotted" button is pressed by the customer, all of the form fields are sent to personspotted.php.

```

<FORM action="personspotted.php" method="post">
    <strong>
        Date: <INPUT type="text" size="8" name="date"
value=<?=$date?>"onFocus="if(this.value == '<?=$date?>') {this.value = '';}"
onBlur="if (this.value == '') {this.value = '<?=$date?>';}" />
        Time: <INPUT type="text" size="4" name="time" value=<?=$time?>"onFocus="if(this.value == '<?=$time?>') {this.value = '';}" onBlur="if (this.value == '') {this.value = '<?=$time?>';}" />
        Latitude: <INPUT type="text" size="5" name="lat" value=<?=$lat?>"onFocus="if(this.value == '<?=$lat?>') {this.value = '';}" onBlur="if (this.value == '') {this.value = '<?=$lat?>';}" />
        Longitude: <INPUT type="text" size="5" name="long" value=<?=$long?>"onFocus="if(this.value == '<?=$long?>') {this.value = '';}" onBlur="if (this.value == '') {this.value = '<?=$long?>';}" />
        <br>
        <INPUT type=submit style="color:red; font: 162.5% 'Trebuchet MS', sans-serif;" value="Person spotted!">
    </strong>
</FORM>

```

End gpscontrols.php

File: personspotted.php

Description:

gpscontrols.php triggers this page to be loaded. It takes the variables posted to it and appends them to the database, and returns the user back to the live GPS page. In combination with livegps.php, an iframe is needed for things to be operated smoothly.

```

<?php

// Catch the incoming information from GPS controls and add it to the database

// Try to open up the file handle in append mode.
$myFile = "peoplepositions.dat";
$fh = fopen($myFile, 'a') or die("Can't open file");

// Construct the new line of the file by pulling from the server's POST variable
// This effectively transfers everything from the GPS controls form into the file.
$line = $_POST["date"].','.$_POST["time"].','.$_POST["lat"].','.$_POST["long"]."\n";

// Write the line and close
fwrite($fh, $line);
fclose($fh);

?>

<html>
  <head>
    <!-- Redirect the user back to the live GPS page (where they came from) -->
    <meta http-equiv="Refresh" content="1;url=livegps.php">
  </head>

  <body style="font: 100% 'Trebuchet MS', sans-serif; margin: 0px;">
    <p>
      <center>
        <h1>The person's position has<br>
        been added to the database</h1>
      </center>
    </p>
  </body>
</html>

End personspotted.php

```

File: search.php
 Description:
 Generates a form the user can query the robot's database with. Similar to the two files above, it posts results to a page that displays a result. In this case, the page posts to search_iframe.php.

```

<form action="php/search_iframe.php" method="post">
<center>
<h3>Generate a map of all persons spotted</h3>
<table width=100%>
<tr>
  <td align=center>
    <h3>From:</h3>
  </td>
  <td align=center>
    <h3>To:</h3>
  </td>
</tr>
<tr>
  <td align=center>
    Start Date: <input type="text" id="datepicker" name="startdate">
  </td>
  <td align=center>
    End Date: <input type="text" id="datepicker2" name="enddate">
  </td>

```

```

</tr>
<tr>
    <td align=center>
        Start time : <input type="text" name="starttime" /><br>
    </td>
    <td align=center>
        Start time : <input type="text" name="endtime" /><br>
    </td>
</tr>

</table>

<input type="submit" />
<h6>For times, please use military format (e.g. 1:15PM = 13:15)</h6>
</center>
</form>

End search.php

```

File: search_iframe.php
 Description:
 This page generates a Google map of specific points in the database. It is made to be used in conjunction with search.php in an iframe.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:v="urn:schemas-microsoft-com:vml">
    <head>
        <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
        <title>Live GPS position</title>
        <!-- Include the Google Maps API. That long list of characters is the API key issued to the group by Google in order to be able to use the maps service -->
        <script
src="http://maps.google.com/maps?file=api&v=2&sensor=false&key=ABQIAAA8uVEzOEqvussyx07iaunhBQBQLbvr-JTDVWowqsj3EbfZ8-fBQH4BuD6AjCEly-VKDQXvuOHf6Oyw"></script>
        <?php

        // Declare some size variables to control how big the map is.
        $sizeX = 600;
        $sizeY = 400;

    ?>
    <script type="text/javascript">
        function initialize() {
            // If the browser is compatible, don't bother with anything
            if (GBrowserIsCompatible()) {
                // Declare a map object that is the correct size
                var map = new GMap2(document.getElementById("map_canvas"), { size: new
GSize(<?=$sizeX?>, <?=$sizeY?>) });

            <?php

            // Set the center to be in the vicinity of where points are
            // the user can always move this so it is hard coded.
            echo 'var CENTER_LAT = 42.7312;';
            echo "\n";
            echo 'var CENTER_LONG = -73.6863;';

        ?>
            // Actually set the center of the map

```

```

map.setCenter(new GLatLng(CENTER_LAT, CENTER_LONG), 15);

// Add large map controls and set the mode to hybrid
map.addControl(new GLargeMapControl());
map.setMapType(G_HYBRID_MAP);

// Create data, time, lats, and longitude arrays.
// An element from each of these four arrays is
// what is needed to define a point on the map.
var dates = new Array();
var times = new Array();
var lats = new Array();
var longs = new Array();

<?php
    // This is dynamic PHP code that changes how the google maps
    // javascript behaves every time the page is loaded.

    // Open up a file handle to the database. Do nothing if this
    // is impossible. The database is formatted in plaintext
    // comma seperated variable, making it easy for PHP to read.
    if (($handle = fopen("peoplepositions.dat", "r")) !== FALSE)
    {
        $i = 0;
        // Read in each appropriate line of the file as an array
        // of comma seperated variables.
        while (($data = fgetcsv($handle, 1000, ",")) !== FALSE)
        {
            // Print the contents of each line to the source code
            // of the javascript. This effectively populates
            // the four arrays declaired up top with information
            // that will be used to define points.
            echo 'dates[".$i."] = "'.$data[0].'" ;' ."\n";
            echo 'times[".$i."] = "'.$data[1].'" ;' ."\n";
            echo 'lats[".$i."] = '.$data[2].';' ."\n";
            echo 'longs[".$i."] = '.$data[3].';' ."\n";
            $i++;
        }
        // Finally close the file handle
        fclose($handle);
    }
?>
// Back to JavaScript. Define a base icon that will be used
// over and over on the map.
var baseIcon = new GIcon(G_DEFAULT_ICON);
baseIcon.shadow = "http://www.google.com/mapfiles/shadow50.png";
baseIcon.iconSize = new GSize(25, 39);
baseIcon.shadowSize = new GSize(37, 34);
baseIcon.iconAnchor = new GPoint(9, 34);
baseIcon.infoWindowAnchor = new GPoint(9, 2);

// This function creates a marker whose info window displays
// the letter corresponding index in the four arrays up top.
function createMarker(point, index) {
    // Create an icon object for this point from the base icon
    var new_icon = new GIcon(baseIcon);

    // Set up the GMarkerOptions object. In particular, this
    // defines the date and time values associated with the marker.
    markerOptions = { icon:new_icon };
    var marker = new GMarker(point, markerOptions);
    var date = dates[index];

```

```

        var time = times[index];

        // Add an event listener to the marker that has a window pop
        // up that contains the date and time the suspect was spotted.
        GEvent.addListener(marker, "click", function() {
            marker.openInfoWindowHtml("Suspect found on " + date + " at " + time);
        });

        // Return the marker that was created
        return marker;
    }

    // Finally, it is time to populate the map with all the points
    // returned by PHP during the search.
    var i = 0;
    for ( i = 0 ; i < dates.length; i++ )
    {
        // For all the items in the arrays, create a new, correctly
        // formatted marker in the right location.
        var latlng = new GLatLng(lats[i], longs[i]);
        map.addOverlay(createMarker(latlng, i));
    }
}

</script>
</head>
<body onload="initialize()" onunload="GUnload()" style="font: 100% 'Trebuchet MS',
sans-serif; margin: 0px;" align=center>
<center>
<?php
    // Output the map at the correct size.
    echo '<div id="map_canvas" style="width: '.$sizex.'px; height:
'.$sizey.'px;"></div>';
?
</center>
</body>
</html>

```

End search_iframe.php

File: serial.php

Description:

This page displays and handles writing characters to the Arduino to control various hardware on the robot when buttons are clicked by the user on the web page.

```

<?php
    // Check the server's GET action variable
    // to see if an action has to be performed
    if (isset($_GET['action']))
    {
        // Action required! Need to issue the command to the arduino...
        $serchar;

        // A very large switch statement that writes the correct character
        // to the serial port given the appropriate command.
        switch ($_GET['action'])
        {
            case 'cameraup':
                $serchar = 'w';
                break;
            case 'cameradown':

```

```

        $serchar = 's';
        break;
    case 'cameraleft':
        $serchar = 'a';
        break;
    case 'cameraright':
        $serchar = 'd';
        break;
    case 'cameracenter':
        $serchar = 'c';
        break;
    case 'forward':
        $serchar = 'i';
        break;
    case 'back':
        $serchar = 'k';
        break;
    case 'stop':
        $serchar = 'n';
        break;
    case 'left':
        $serchar = 'l';
        break;
    case 'right':
        $serchar = 'j';
        break;
    }

    // Now that the correct character has been determined,
    // it needs to be written to the serial port.

    // Need the PHP serial class to communicate with the arduino
    require("php_serial.class.php");

    // Create a new serial object from the PHP serial class.
    $serial = new phpSerial();

    // Set all the parameters of the connection appropriate to the Arudino.
    // These are all specified in the Arduino documentation.

    // Arduino is located in this USB port always.
    $serial->deviceSet("/dev/ttyUSB0");

    // Baud rate: 9600
    $serial->confBaudRate(9600);

    // Parity (this is the "N" in "8-N-1")
    $serial->confParity("none");

    // Character length (this is the "8" in "8-N-1")
    $serial->confCharacterLength(8);

    // Stop bits (this is the "1" in "8-N-1")
    $serial->confStopBits(1);

    // Finally, open the connection, send the character, and then close it
    $serial->deviceOpen();
    $serial->sendMessage($serchar);
    $serial->deviceClose();
}

// Utility function that generates all the HTML necessary to create a
// jQuery button of the appropriate class and action type.

```

```

function generateicon($action, $iconclass)
{
    echo '<a href="'. $_SERVER['PHP_SELF'] .'?action='.$action.'" class="fg-button fg-
button-icon-solo ui-state-default ui-corner-all">';
    echo '<span class="ui-icon ui-icon-'.$iconclass.'"></span></a>';
}

?>

<!--
This is the table that houses two other 3x3 tables.
One controls the camera while the other controls the
movement of the robot.
-->



|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                                                                                                                      |                                                                                                                                                                                                 |                                                                                   |                                                                                          |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|
| Start of the table that handles camera control <table border="0" cellpadding="2px"> <tr> <td> <div align="center"> <!-- Here is where the up arrow for the camera appears --> &lt;?php     // Instead of hand coding each individual button for all     // the different icons and actions, it is much quicker     // and cleaner to functionalize the button making process     // and do all the hard work from PHP.     generateicon('cameraup','arrowthick-1-n'); ?&gt; </div> </td> <td> <div align="center"> <!-- Here is the middle row. Going across the row, it contains the left, center, and right controls --> &lt;?php generateicon('cameraleft','arrowthick-1-w'); ?&gt; </div> </td> <td> <div align="center"> &lt;?php generateicon('cameracenter','bullet'); ?&gt; </div> </td> </tr> <tr> <td> <div align="center"> &lt;?php generateicon('cameraright','arrowthick-1-e'); ?&gt; </div> </td> </tr> </table> | <div align="center"> <!-- Here is where the up arrow for the camera appears --> &lt;?php     // Instead of hand coding each individual button for all     // the different icons and actions, it is much quicker     // and cleaner to functionalize the button making process     // and do all the hard work from PHP.     generateicon('cameraup','arrowthick-1-n'); ?&gt; </div> | <div align="center"> <!-- Here is the middle row. Going across the row, it contains the left, center, and right controls --> &lt;?php generateicon('cameraleft','arrowthick-1-w'); ?&gt; </div> | <div align="center"> &lt;?php generateicon('cameracenter','bullet'); ?&gt; </div> | <div align="center"> &lt;?php generateicon('cameraright','arrowthick-1-e'); ?&gt; </div> |
| <div align="center"> <!-- Here is where the up arrow for the camera appears --> &lt;?php     // Instead of hand coding each individual button for all     // the different icons and actions, it is much quicker     // and cleaner to functionalize the button making process     // and do all the hard work from PHP.     generateicon('cameraup','arrowthick-1-n'); ?&gt; </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | <div align="center"> <!-- Here is the middle row. Going across the row, it contains the left, center, and right controls --> &lt;?php generateicon('cameraleft','arrowthick-1-w'); ?&gt; </div>                                                                                                                                                                                      | <div align="center"> &lt;?php generateicon('cameracenter','bullet'); ?&gt; </div>                                                                                                               |                                                                                   |                                                                                          |
| <div align="center"> &lt;?php generateicon('cameraright','arrowthick-1-e'); ?&gt; </div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                                                                                                                                                                                                                                                                                                                                                      |                                                                                                                                                                                                 |                                                                                   |                                                                                          |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                                                                                                                      |                                                                                                                                                                                                 |                                                                                   |                                                                                          |


```

```

        </div>
    </td>
    <td>
        <div align="center">
            <?php generateicon('cameradown','arrowthick-1-s'); ?>
        </div>
    </td>
    <td>
        <div align="center">
            <?php generateicon('forward','arrowthick-1-n'); ?>
        </div>
    </td>
    <td>
        <div align="center">
            <?php generateicon('left','arrowthick-1-w'); ?>
        </div>
    </td>
    <td>
        <div align="center">
            <?php generateicon('stop','bullet'); ?>
        </div>
    </td>
    <td>
        <div align="center">
            <?php generateicon('right','arrowthick-1-e'); ?>
        </div>
    </td>
    <td>
        <div align="center">
            <?php generateicon('back','arrowthick-1-s'); ?>
        </div>
    </td>
    <td>
        <div align="center">

```

```

        </div>
    </td>
</tr>
</table>
</td>
</tr>
</table>

```

End serial.php

File: command.php

Description:

This page displays a high level dash board that the user can use to interact with most parts of the robot.

```

<html>
<?php

include_once("php/header.php");
include_once("php/navigation.php");

?>

<td valign=top bgcolor=D8D8D8>
<!-- MAIN, put content here -->
<h2 align=center>Command Center</h2>
<hr>
<table align=center cellpadding=20 border=0 cellspacing=0 width=100%>
<tr>
<td>

<table align=center cellpadding=20 border=0 cellspacing=0 width=100%>
<tr>
<td>
<p>
<a href="#" id="dialog_link" class="ui-state-default ui-corner-all">
<span class="ui-icon ui-icon-newwin"></span>Open Camera Control Widget
</a>
</p>
<div id="dialog" title="Camera Control Widget">
<center>
<div class="ui-widget">
<div class="ui-state-error ui-corner-all" style="padding: 0 .7em;">
<?php include_once('php/webcam.php'); ?>
</div>
</div>
<iframe src ="cameracontrol_iframe.php" width="100%" height=240 scrolling=no
frameborder=0 marginheight="0" marginwidth="0">
<p>Your browser does not support iframes, which is pretty sad</p>
</iframe>
</center>
</div>
</td>

<td><a href="#" id="dialog_link2" class="ui-state-default ui-corner-all">
<span class="ui-icon ui-icon-newwin"></span>Open GPS Widget
</a></p>
<div id="dialog2" title="GPS Module">
<center>
<div class="ui-widget">
```

```

<div class="ui-state-error ui-corner-all" style="padding: 0 .7em;">
    <iframe src ="php/livegps.php" width="100%" height=400 scrolling=no
frameborder=0 marginheight="0" marginwidth="0">
        <p>Your browser does not support iframes, which is pretty sad</p>
    </div>
</div>
</center>
</div>
<br>
<br>
<br>
</td>
</tr>
</table>

</td>
</tr>
</table>

</td>
<?php include_once("php/footer.php") ?>
</html>

```

End command.php

File: cameracontrol_iframe.php
Description:
Pulls in serial control as well as styling elements to form the controls seen below the webcam feed in the dialog box.

```

<html>
<?php include_once('php/css_javascript.php'); ?>
<?php include_once('php/serial.php'); ?>
</html>

```

End cameracontrol_iframe.php

File: webcam.php
Description:
Very simply displays a usable webcam feed linked to mjpg-streamer.

```

<?php
    // This creates an image whose source is taken from the mjpg server.
    // Modern day web browsers will animate the stream, resulting in a
    // very usable live webcam feed with very little effort or lag
    echo '';
?>

```

End webcam.php