ECE Final Report – Spring 2011

Team 3 – Autonomous Targeting Vehicle (ATV)

From Left to Right: **Anthony Myers, Sandunmalee Abeyratne, Daniel Barrett, Sebastian Hening**

Team Members:

#1: Daniel Barrett **Signature:** ______ **Date: 05/02/2011**

#2: Sebastian Hening **Signature:** ______ **Date: 05/02/2011**

#3: Sandunmalee Abeyratne **Signature:** ______ **Date: 05/02/2011**

#4: Anthony Myers **Signature:** ______ **Date: 05/02/2011**

CRITERION	SCORE	MPY	PTS
Technical content	0 1 2 3 4 5 6 7 8 9 10	3	
Design documentation	0 1 2 3 4 5 6 7 8 9 10	3	
Technical writing style	0 1 2 3 4 5 6 7 8 9 10	2	
Contributions	0 1 2 3 4 5 6 7 8 9 10	1	
Editing	0 1 2 3 4 5 6 7 8 9 10	1	
Comments:		**TOTAL**	

TABLE OF CONTENTS

Abstract

The Autonomous Targeting Vehicle is designed to autonomously navigate to designated GPS waypoints and to visually locate and follow targets. Our motivation for creating the ATV was to remotely track objects for surveillance purposes. The ATV uses two microcontrollers to control its speed and direction and to avoid obstacles. The on-board Intel Atom processor parses GPS and sensor data to accurately determine location of the robot. The Atom also processes images from the webcam to facilitate visually tracking a target. Integrating an Atom processor allowed us to create a hands-free, user-friendly interface to facilitate the operation of the device. The five project specific success criteria (PSSC) were successfully demonstrated at the completion of this project.

1.0 Project Overview and Block Diagram

The ATV is an autonomous wheeled vehicle that can navigate as well as visually track and follow targets. This vehicle will use GPS to determine its current location and will be able to autonomously navigate to another location using a range finder and proximity sensors to detect and avoid obstacles. To improve accuracy of its current position, this vehicle will use the Kalman filter algorithm to perform sensor fusion and "dead reckoning" using the information from the compass and wheel encoders. These features also allow the vehicle to navigate when a GPS signal cannot be received. Additionally, it will be able to visually track a target via a webcam. An Intel Atom board performs image processing on webcam data, maps the moving robot, and uses a path-finding algorithm to determine what path to follow to reach the intended destination. The Atom also allows the user to connect remotely via a wireless connection and control the robot through a GUI interface. A pair of Freescale 9S12C32 microcontrollers will communicate with the Atom board through a serial interface. They will send sensor data to the Atom board and receive from it instructions which they will use to control the camera servos and wheel motors.

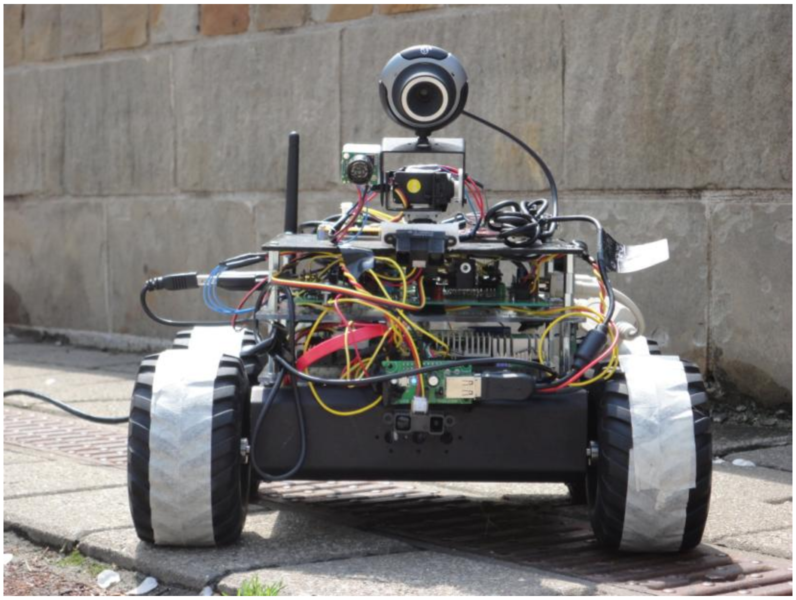

Figure 1.1: Completed Project

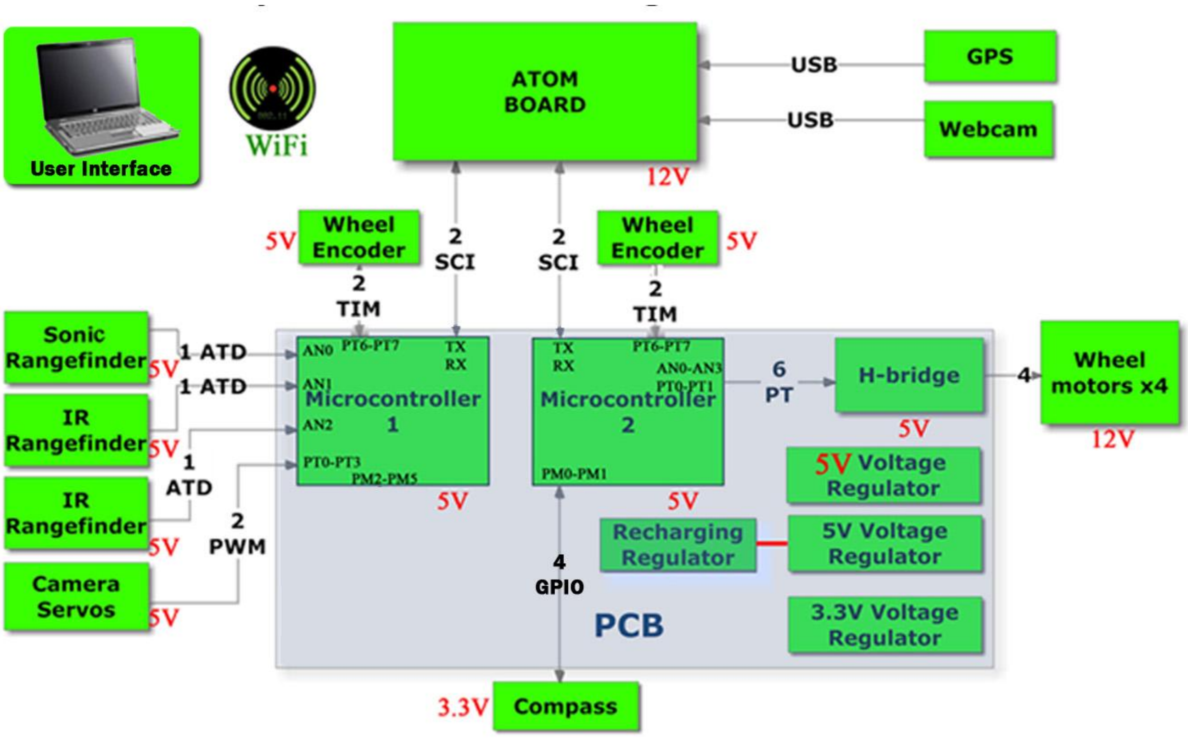

Figure 1.2: Project Block Diagram

2.0 Team Success Criteria and Fulfillment

We were able to successfully complete all five PSSCs at the completion of the project. The PSSCs along with their assessment of completion are as follows:

1. An ability to determine location within 10 meters based on GPS data.

   We were able to show the current location of the robot on a map image of the Purdue University campus.

2. An ability to control the speed and direction of the motors on each side in order to move forward, backward, turn left, and turn right.

   We demonstrated that the robot could follow a target that was selected in the webcam view, and as the target moved around, the robot was able to move in all directions in order to follow the target and stay in close proximity to it.

3. An ability to visually track and follow a target via webcam.

   When an object is clicked in the webcam view, the robot is able to control the webcam servos to stay pointed in the direction of the target and follow it as it moves.

4. An ability to detect obstacles, and determine their distance with a sonic range finder.

   The robot is able to detect obstacles in front of it as it moves toward a destination and stops when the obstacle is within a certain distance from it. Also, if the obstacle moves toward the robot, the robot is able to detect this, and compensates by moving backward away from the object.

5. An ability to determine changes in position using wheel encoders and a compass.

   We were able to show on the GUI that the direction of travel changes as the compass and wheel encoder data is retrieved by the microcontroller. The distance and direction that the robot travels can also be seen on the GUI.

3.0 Constraint Analysis and Component Selection

Our design is an autonomous wheeled vehicle that can navigate to designated way-points as well as visually track and follow targets. The constraints confronting this design are formed from the functions it is required to perform.

Computationally, the robot will need sufficient power to accomplish its tasks. The vehicle will use GPS to determine its current location within 5-10m. It will also use a compass and wheel encoders to “dead reckon” changes in position and use the Kalman filter algorithm to combine this with the GPS data and improve the precision of movement. The robot will be able to autonomously navigate to another location using ultrasonic and IR range-finders to detect and avoid obstacles. It will create a map of the detected obstacles and perform real-time path-finding around the obstacles, recalculating as new obstacles are detected. Additionally, this vehicle will be able to visually track a target using the Lucas-Kanade optical flow algorithm [1] and follow the object, maintaining a constant distance to it as it moves. Our project will use an Atom board to do the image processing, and to allow the user to connect remotely through a wireless connection to initiate tracking and navigation. Two Freescale microcontrollers will control the motors and pull data from the sensors, communicating with the Atom through serial ports.

Because it is a mobile device, its chassis and motors will need to be strong enough to carry all components, and be able to move on mildly rough terrain, such as grass. It will also be required to carry a battery, which will need to be capable of powering all the electronics without being so heavy that it compromises mobility.

Updated PSSCs:

1. *An ability to determine location within 10m based on GPS data.*
2. *An ability to control the speed and direction of the motors on each side in order to move forward, backward, turn left, and turn right.*
3. *An ability to visually track and follow a target via web-cam autonomously.*
4. *An ability to detect obstacles, and determine their distance with a sonic range finder.*
5. *An ability to determine changes in position using wheel encoders, accelerometers, and a compass.*

3.1 Design Constraint Analysis

In order for the design to possess the desired functionality, several design constraints must be overcome. The primary design constraints are computational power, electrical power, mechanical power, and sensor precision and accuracy. A large amount of processing will need to be done in order to capture sensor data, interpret it, and make decisions in real time. These computations involve creation of an obstacle map and performing path-finding to a way-point or queue of way-points. The robot needs to constantly recalculate its position and heading based on fusion of sensor data. Upon detection of an obstacle, the robot will need to first add that obstacle to the internal obstacle map, find a new path around the obstacle from the current position to the destination, and then control the wheel motors in order to follow this new path. In the target tracking mode, the video processing in particular consumes a lot of processor time, using 80% of a 1.6 GHz Intel Atom processor, as tests we have conducted have shown. To provide robust tracking of a moving object, this will require the frame rate to be roughly 20 frames per second. Because our design is a mobile robot, it will need to carry a power supply, so having a battery which can supply enough power over a reasonable period of time is necessary. Minimizing power consumption will help this goal. Being a mobile device, the chassis and motors must be strong enough to support and move the weight of all components while traversing mildly rough terrain such as bumpy grass. This means that a balance must be achieved between having a battery which supplies enough power, but also is light enough to carry. It will also be important for the sensors to have enough precision and reliability to be useful for identifying obstacles, and determining the current trajectory.

3.2 Computation Requirements

Computational tasks can be broken into two groups: those accomplished by the Atom board, and those to be accomplished by the microcontrollers. The Atom board will take responsibility for video processing, sensor fusion, obstacle mapping, path-finding, implementation of a wheel speed PID controller, and taking in user input through a Wi-Fi network. The microcontrollers will be responsible for continually pulling data from the sensors, and low level control of the motors and servos. Both devices will need to communicate via serial port. These tasks must all occur in real-time in order for the robot to track targets and navigate. The microcontrollers will poll the sensors at a rate of at least 50Hz, which should be easy to accomplish given that they run at 25MHz. This information will be sent to the Atom board, where the higher level computation will be performed. It is known that the image processing uses 80% of the Atom board CPU time, so other features will have to fit into the remaining 20%.

3.3 Interface Requirements

Because the ATV is a mobile robot, it does not have many external interfaces. It will connect to a wireless network, and accept remote-desktop connections for remote control. It also has a 19 Volt battery charging input.

The microcontroller requires a 5V power, and will be communicating with a dual H-Bridge, servos, two wheel encoders, two Infra-red range-finders, and one Sonic range-finder, all of which run on 5V. It will also communicate with a compass, which runs on 3.3V, and will require the use of a voltage translator. The H-Bridge will be interfaced to the microcontroller through six optical isolators. None of the signals driven by the microcontroller require any significant current draw.

3.4 On-Chip Peripheral Requirements

The microcontroller will control the wheel motors through two H-Bridges. This will require two 8-bit PWM pins, and four general purpose pins. The two camera servos require one 16-bit PWM each. This is because the servos are extremely sensitive to small changes in pulse width. We will be using three rangefinders, one sonic and two Infrared, all of which have analog outputs, thus requiring three ATD pins. The two wheel encoders will require two pulse accumulator pins. The microcontroller will need one serial port to communicate with the Atom board. The accelerometer and magnetometer both use I2C, requiring either an I2C module, or two general purpose pins to implement the protocol. The GPS uses a serial interface which will be connected to the Atom board.

3.5 Off-Chip Peripheral Requirements

There are no anticipated off-chip peripherals. Wireless communication will be handled by the Atom board.

3.6 Power Constraints

The robot will be powered by a Nickel-Metal-Hydride battery. This battery will need to supply enough power for the Atom board [10](2A max at 12V), microcontroller (70mA max at 5V), wheel motors (1A max x 4 motors at 12V), servos (300mA max at 5V), and sensors: magnetometer [9] (0 .8mA at 3.3V), and ultrasonic [11] and IR rangefinders [12] (3.3mA and 66mA at 5V). The current drawn by the Atom board, motors, and servos is so much more than the electronics that the power drawn by the microcontrollers and sensors is less important.

3.7 Packaging Constraints

The robot should be able to withstand its own weight and drive on mildly rough terrain at approximately walking speed, and be large enough to hold all the motors and electronics. The desired size is roughly 12 inches long by 12 inches wide by 6 inches high.

3.8 Cost Constraints

Our cost constraint is the limit on our willingness to spend money. This is approximately ~$200/person. Our device is not competing directly with other products, but slightly more robust robot development platforms with electronics included cost 3 to 5 thousand dollars, which is mostly likely due to the cost of labor in designing them and the low volume of sale.

3.9 Component Selection Rationale

The major components which needed to be chosen were the Atom board, microcontroller, chassis/wheels kit, GPS, range sensor, accelerometer, magnetometer, and batteries.

Atom board

We chose the Atom board [10] provided by the 477 lab primarily because it is free, and because it meets our minimum requirements. We have tested our OpenCV-based video processing code and an initial version of the sensor fusion algorithm, and together they use ~85% of the CPU time. Other Atom boards, such as the Zotac IONITX [13] are available with more processing power, but were not judged to be worth the cost.

Table 3.9.1: Processing Choices

	Zotac IONITX-A-U	**Zotac IONITX-G-E**	**I-Base N270**
Clock Speed	1.6Ghz	1.6Ghz	1.6Ghz
Power interface	Onboard supply with single 12V input	No onboard supply. Requires many inputs at several voltages	Onboard supply with single 12V input
Wi-Fi	Has Wi-Fi	No Wi-Fi	Has Wi-Fi
# of CPU cores	2	2	1
Cost	~$200	~$200	($300) FREE

Microcontroller

For the microcontroller, we again chose primarily based upon cost, and meeting our minimum requirements. We have chosen to use two Freescale 9S12C32 microcontrollers [2]. Together, our team possesses 4 of these devices, and already has experience with them. A single 9S12C32 comes close to meeting our requirements, but is short on pulse accumulators and PWMs. Using two of them will solve this problem, and allow the reading of sensors and motor control to happen more often by reducing the workload of each microcontroller. When compared to another microcontroller, such as a single Freescale 3s12XD256 [3], the pair of 9S12's amply meets our requirements, and draws less current.

Table 3.9.2: Microcontroller Choices

	Freescale 9s12C32	**Freescale 3s12xd256**	**2x Freescale 9s12C32**	**required**
PWMs	5	8	10	6
Pulse Accumulators	1	1	2	2
Serial Ports	1	2	2	1
I/O Pins	32	91	64	~8 General Purpose pins
A/D converters	8	16	16	3
Supply Voltage	5	3.15 to 5	5	N/A
Current Draw Max	35mA	110mA	70mA	N/A
Cost	FREE	$16.32	FREE	N/A

Wheel/Motor/Chassis Kit

The next major component chosen was the chassis and wheels kit. We found three potential items in the Lynxmotion 4WD1 [4], the DFRobot 4WD [5], and the Dagu Wild Thumper [6]. We ruled out the DFRobot because of its low payload capacity and small size, which seemed too limiting. The Dagu Wild Thumper would have been powerful enough, but perhaps too powerful. The motors in the Dagu Wild Thumper were listed as drawing 6.6A each on startup, which would have greatly increased the demands on our power supply. In the end, the Lynxmotion 4WD1 kit was chosen. It has a respectable payload capacity, while drawing significantly less power than the Dagu Wild Thumper. It also had the bonus of having wheel encoders available designed specifically for the motors.

Table 3.9.3: Wheel/Motor/Chassis Choices

	Lynxmotion 4WD1	**DFRobot 4WD**	**Dagu Wild Thumper**
Payload Capacity	5lbs	1.7lbs	5lbs+
Materials	aluminum	plastic	aluminum
Motor no load current	114mA x4	not listed	420mA x4

Motor stall current	1.2 A x 4	not listed	6.6A x4
Motor current under rated load	233mA x4	1.2A	not listed
Dimensions	10"x12.74"x4.75"	9"x7.2"x4.3"	11"x12"x5"
Encoder Availability	yes	included	no
Price	$217.00	$72.00	$250.00

GPS Module

The next important component was the GPS module. We found a number of potential GPS modules, but many did not have an antenna included. Of those with an antenna, the EM-406A SiRF III [7] and the LS20031 [8] looked promising. The LS20031 seemed to have better specifications: better accuracy, lower current, faster update rate, but it received bad reviews for having trouble getting a signal. The EM-406A SiRF III, however, received good reviews, and had a tutorial available for interfacing with it. For this reason, we chose the SiRF III.

Table 3.9.4: GPS Choices

	EM-406A SiRF III	**LS20031**
Positional Accuracy	10m	3m
current Draw	70mA	41mA
antenna included	yes	yes
update rate	1 Hz	10Hz
interface	serial	TTL serial
Reviews	excellent	mediocre: bad signal
Price	$60.00	$60.00

Another important component is the H-Bridge. We found the VNH2SP30-E [11], and the SolarBotics L298 Motor Driver Dual H-Bridge [12]. The two components are of similar price and features. The VNH2SP30-E can handle higher current, and costs less, but only has 1 channel. The L298N has 2 channels, and has an enable feature which allows the motors to coast, in addition to the standard power-on vs. brake, which was the determining factor for choosing it.

The voltage regulators were also an important choice. We require 5V and 3.3V supplies, and so will be using regulators to convert the 12V output of a Nickel-Metal-Hydride battery to the proper voltages. The 5V supply will be maintained using the OKR-T switching regulator [16], which can supply up to 3A. The 3.3V supply will be maintained by a 3.3V circuit created with the LM317 [17]. A linear regulator is sufficient for the 3.3V supply because it will be drawing very little current, and will be using the 5V supply as an input, meaning that it will also drop very little voltage.

The battery has been chosen to be a 12V Nickel-Metal-Hydride battery for its high power to weight ratio, combined with high current sourcing ability, and voltage matching the requirement of both the motors and Atom board. The most current that the design is likely to require is ~4.5A, so a battery able to supply at least 5A is desirable. A battery pack such as [18] or [19] would likely be sufficient, except for the voltage 14.8V output. Both are 14.8V Lithium-ion battery-packs, which would require an additional 12V regulator. Therefore, a 12V, Nickel-Metal-Hydride battery [20] was chosen instead. It has a 4200mAh rating and is able to source up to 40A continuously.

3.10 Summary

This report contains an overview of the project, along with updated PSSCs, and block diagram, as well as a discussion of the major design constraints.

4.0 Patent Liability Analysis

The Autonomous Targeting Vehicle (ATV) is a mobile robot that is designed to autonomously travel to user specified locations, avoiding obstacles along the way. The vehicle will determine its proximity to surrounding objects using two infrared range finders, and a single sonic range finder. It will visually track objects using a webcam, allowing the user to maintain visual contact with the target. In the absence of a GPS signal, the user will click an object in the webcam view to instruct the vehicle where to travel next. Because the GPS coordinates are only accurate to 10 meters, the robot will utilize a Kalman filter for “dead reckoning” its position based on sensor data from an accelerometer, a compass, and wheel encoders. A graphical user interface will show the user the path the robot travels to get to its destination using a screenshot from Google Maps.

Several features of the ATV have the possibility of patent infringement. The features to consider are the method of autonomous movement using range finders for obstacle avoidance, the vision assisted navigation, the method of “dead reckoning” using a Kalman filter from sensor data, and the use of a Google Maps screenshot. This document will discuss a few patents that are very similar to the ATV, an analysis of how the ATV is different or similar to said patents, and recommended actions for avoiding legal issues.

4.1 Results of Patent and Product Search

Several patents were searched for operations and methods of mobile robots that are similar in functionality to the ATV. Patents of interest dealt with autonomous robotic vehicles, object avoidance capabilities, robotic systems with vision integrated performance, and robotic systems with “dead reckoning” capabilities. The issue with Google Maps copyright infringement was also researched. The following four patents/copyrights are closely related with the features mentioned above in section 4.0:

4.1.1 United States Patent 6515614, filed on October 11, 2001: Autonomous moving apparatus having obstacle avoidance function [1]. This patent concerns an autonomous moving apparatus that moves toward a destination while detecting and avoiding obstacles using a horizontal plane scanning radar device to detect a position of an obstacle, and an obstacle sensor for detecting an obstacle in a space different from the scanning plane of the radar device [1]. The patent has many claims that are relevant to the ATV. The non-scan type sensor is a set of supersonic sensors, arranged in a semi-circular shaped area and synchronized through supersonic-wave oscillation timing settings, or an optical sensor that detects objects based on light reflections [1]. When an obstacle is detected, the controller decreases traveling speed, and is only permitted to be a certain distance from the object [1]. When an obstacle is detected, the controller changes the direction of the apparatus until no obstacle is detected, and resumes movement [1]. The apparatus also utilizes a specific-configuration detecting element for detecting the presence of an object having a specific configuration, and uses the scan-type sensor to determine changes in movement of that object [1].

4.1.2 United States Patent 7817847, filed on October 5, 2005: Robot system with vision sensor [2]. This patent concerns a robot system having a vision center that obtains image data of a working environment of the robot. The major claim consists of a system that includes a section for controlling the robot, an imaging section for obtaining image data on the working environment of the robot, an image processing section, a vision controlling section to obtain, transmit, and process the data, and a communication network to which all networks are connected [2]. The robot also has a vision controlling section that makes the imaging section obtain image data at predetermined time intervals and displays the image data as a moving image [2]. The robot controlling section works to control the robot in accordance to a work program, and performs position correction based on position data retrieved from the imaging section [2].

4.1.3 United States Patent Application 20070118248, filed on October 17, 2006: Method and apparatus for reckoning position of moving robot [3]. This application concerns a method and apparatus for reckoning a position of a moving robot using dead reckoning and range sensing. As mentioned in the claims, the robot reckons its position by performing dead reckoning to determine a variation state, determining absolute position by measuring its distance from a fixed object, and predicting an optimized current position of the moving robot using the variation state and absolute position [3]. It uses an encoder and/or gyroscope to perform dead reckoning, and determines its fixed position using at least one fixed position in a charge station of the moving robot and another fixed position [3]. Radio waves are used in determining distance to the fixed locations [3]. A Kalman filter calculates the current state using the variation state and absolute position, using information from an auxiliary sensor, which comprises at least one of a nearby obstacle sensor, a laser sensor, a distance sensor, and a camera [3].

4.1.4 The last case of infringement involves Google Maps. Permission guidelines regarding specific use cases for Google Maps are as follows: "all use of Google Maps and Google Earth and Content MUST provide attribution to Google and our suppliers [4]." Content cannot be scraped or exported from Google Maps or Earth or be saved for offline use [4].

4.2 Analysis of Patent Liability

Of the three patents found for which the ATV has the possibility of infringing, one is literally infringed, one has the possibility of being infringed under the Doctrine of Equivalents, and one has no issues of infringement. For the Google Copyright, the ATV directly violates the copyright. The analysis to follow will explain how the functions performed by the ATV are similar and/or different from those of the patents/copyrights identified in section 4.1.

4.2.1 The ATV performs exactly the same function as U.S. Patent 6515614 [1], but achieves the functionality in a much different way. The mobile apparatus of this patent detects obstacles using a scanning radar device, that scans the horizontal plane for objects, and a non-scan type obstacle sensor, which is described to be an optical sensor, or a set of supersonic sensors arranged in a semi-circular shaped obstacle detection area in a horizontal plane in a traveling direction [1]. The ATV detects obstacles using two infrared proximity sensors and a single ultra-sonic range finder. None of the sensors operate as a scan-type sensor, and the collection of sensors is not arranged in the same fashion as the non-scan sensors of the apparatus described in the patent. The major similarities are with the methods of controlling the autonomous apparatus. The patent describes an apparatus that decelerates as it nears an object, and slowly turns until an object is no longer detected. The ATV operates the same way, however this method of avoiding objects appears rather obvious. The ATV does not include a specific-configuration detection method, whereas the patent does. Because of the major differences in the methods of detecting objects and the lack of a specific-configuration detection mechanism, there is no concern for infringement of this patent.

4.2.2 U.S. Patent 7817847 describes a robotic system with a vision sensor. The mechanism it uses to achieve this functionality is a control section, imaging section, image processing section, vision controlling section, and a communication network that connects all the aforementioned sections [2]. Because the patent is written in such vague context, it appears that the ATV will be in violation of this patent. The ATV includes a control section (microcontrollers / Atom board), an imaging section and image processing section (second microcontroller / Atom board), vision controlling section (Atom board), and a communication network (PCB). The function of this patent is accomplished using the same mechanism as the ATV, therefore literally infringing this patent. However, because of the vague nature of the patent, it may not hold up in court.

4.2.3 U.S. Patent Application 20070118248 describes a method and apparatus for reckoning the position of a moving robot. The method mentioned in the application includes dead reckoning using a Kalman filter to determine a variation state, determining its distance from a fixed object to determine its absolute position, and an encoder and/or gyroscope along with an auxiliary sensor to assist in the dead reckoning calculations [3]. The ATV will utilize a compass, an accelerometer, and two wheel encoders to perform the necessary calculations using a Kalman filter. The sensors used for each method are similar, however the ATV uses a compass and accelerometer, where the patent uses strictly a gyroscope. The major difference between the patent and the ATV is that the robot in the patent uses a fixed object to determine its absolute position, whereas the ATV will use a GPS signal to determine its approximate current location. If this patent is granted, the ATV will possibly infringe upon the Doctrine of Equivalents.

4.2.4 The ATV directly violates the usage terms of Content from Google Maps [4]. Google specifically states that Content from Google Maps cannot be exported or saved for offline use [4]. The graphical user interface for communicating with the mobile robot (ATV) uses a screenshot from Google Maps to display location and path information for the user to see.

4.3 Action Recommended

Of the three patents and one copyright that were found to be similar in functionality to the ATV, only two of the patents, along with the copyright, are of concern for some sort of infringement. The ATV implements the mechanism described in patent one [1] in a much different fashion, eliminating the case for infringement. Patent two [2] is worded so vaguely that it doesn't appear as if there is any way to work around the infringing function without completely removing the vision aspect from the ATV. This is not ideal, as the vision aspect is an integral part of the project. In order to eliminate the literal infringement of this patent, we would try to have the patent nullified on the basis of being obvious. If this tactic would not work, the only option would be to license the patent. As for the patent application [3], a patent lawyer would be contacted if the application were granted. The ATV would have to license the patent if necessary to avoid infringement under the Doctrine of Equivalents. Because the ATV utilizes an image (map) of the surrounding area, the only option for dealing with the copyright infringement with Google Maps [4] would be to license the image. For future expandability, a Google API could be integrated into the system and be used for displaying maps of various areas, which would completely remove the case for infringement.

4.4 Patent Analysis Summary

The main features of the Autonomous Targeting Vehicle include autonomous movement using range finders for obstacle avoidance, vision assisted navigation, a method for "dead-reckoning" that involves using a Kalman filter from sensor data, and a graphical user interface that displays a screenshot from Google Maps. Three patents and one copyright were found that were relevant to the functionality of the ATV. The ATV literally infringes upon U.S. patent 6515614, and possibly infringes upon the Doctrine of Equivalents for U.S. patent application 20070118248. U.S. patent 7817847 is of no concern for any type of patent infringement. If U.S. patent 6515614 cannot be successfully nullified, licensing the patent is the only option. If patent application 20070118248 were granted and the ATV infringed it under the Doctrine of Equivalents, it would be licensed accordingly. As for the Google Maps copyright, the image obtained from Google Maps could be licensed, or the ATV's user interface could be expanded to use one of Google's API's, eliminating the case for infringement.

5.0 Reliability and Safety Analysis

The Autonomous Targeting Vehicle (ATV) is an autonomous wheeled vehicle which can navigate to a GPS coordinate as well as track and follow targets. The robot is capable of speeds up to 1.5 m/s and collisions could result in injury to others. The battery used is capable of supplying 40A and in case of a short circuit might catch on fire or explode. Due to this the critical components will be the H-bridge, the microcontrollers, the 5V switching regulator and the battery management chip. Other components that could result in safety and reliability problems include, the voltage regulators, sensors, accelerometer, compass, GPS, webcam and wheel encoders. Passive components will also affect the safety and reliability of the product.

5.1 Reliability Analysis

Of all the components used in the design, there are four that are most likely to fail and affect the reliability and safety of the project: the L298 Dual H-Bridge, the BQ2002 battery management chip, the Murata OKR-T 5V switching regulator, and the two Freescale 9S12C32 microcontrollers.

The L298 Dual H-Bridge drives the four Lynxmotion GHM-16 motors, each drawing a constant 285 mA. The H-Bridge also has a high junction-to-ambient temperature of up to 130C. These factors increase the chance of failure.

The Freescale 9S12C32 microcontroller is the most complicated IC in the design with 32 pins. The two microcontrollers control the motors and camera servos and also acquire the compass, accelerometer, and wheel encoder's data.

The Murata OKR-T 5V switching regulator powers the camera servos, IR and Sonic sensors, and the two microcontrollers. The OKR-T voltage regulator operates at a frequency of 600 KHz which makes it very responsive to current spikes however it increases the probability of failure.

The last component is the BQ2002 battery charging IC which makes sure that the battery is being charged safely. The charging is stopped if the battery exceeds the maximum temperature or voltage, if the voltage of the battery drops, or if the maximum charging time is reached. The temperature is measured using a thermistor which is attached to the battery and the voltage drop over it is measured.

Table 5.1.1: Microcontroller 1 & 2

Parameter name	**Description**	**Value**	**Comments:** regarding choice of parameter value, especially if you had to make assumptions.
C1	Die complexity	.14	CMOS, 8-Bit
π_T	Temperature coeff.	.71	Assume linear temp of 50 C
C2	Package Failure Rate	.015	32 Pins, Nonhermetic
π_E	Environmental Factor	4.0	Ground Mobile
π_Q	Quality Factor	10	Commercially Manufactured component
π_L	Learning Factor	1	More than 2 years in production
$\boldsymbol{\lambda_P}$	Part Failure Rate	1.594	Failures /10^6 hours
MTTF	627352 Hours = 71.5 Years		

Table 5.1.2: Murata OKR-T

Parameter name	**Description**	**Value**	**Comments:** regarding choice of parameter value, especially if you had to make assumptions.
C1	Die complexity	.01	1 to 100 Linear MOS gates
π_T	Temperature coeff.	.71	Assume linear temp 50 C
C2	Package Failure Rate	.002	5 Pins, Nonhermetic
π_E	Environmental Factor	4.0	Ground Mobile
π_Q	Quality Factor	10	Commercially Manufactured component
π_L	Learning Factor	1	More than 2 years in production
$\boldsymbol{\lambda_P}$	Part Failure Rate	.151	Failures /10^6 hours
MTTF	6622516 Hours = 756 Years		

Table 5.1.3: BQ2002

Parameter name	**Description**	**Value**	**Comments:** regarding choice of parameter value, especially if you had to make assumptions.
C1	Die complexity	.02	101 to 1000 Linear MOS gates
π_T	Temperature coeff.	.71	Assume linear temp of 50 C
C2	Package Failure Rate	.0034	8 Pins, Nonhermetic
π_E	Environmental Factor	4.0	Ground Mobile

π_Q	Quality Factor	10	Commercially Manufactured component
π_L	Learning Factor	1	More than 2 years in production
λ_P	Part Failure Rate	.278	Failures /10^6 hours
MTTF	3597122 Hours = 410 Years		

Table 5.1.4: H-Bridge

Parameter name	**Description**	**Value**	**Comments:** regarding choice of parameter value, especially if you had to make assumptions.
C1	Die complexity	.01	1 to 100 Linear MOS gates
π_T	Temperature coeff.	2.8	Linear MOS at 70C
C2	Package Failure Rate	.0067	15 Pins, Nonhermetic
π_E	Environmental Factor	4.0	Ground Mobile
π_Q	Quality Factor	10	Commercially Manufactured component
π_L	Learning Factor	1	More than 2 years in production
λ_P	Part Failure Rate	.548	Failures /10^6 hours
MTTF	1824817 Hours = 208.4 Years		

Entire Design		*4.165*	*Failures /10^6 hours*
MTTF	*240096 Hours=27.4 Years*		

The calculated failure rates of the analyzed components were as expected. The two microcontrollers, having 32 pins and being the most complex ICs in the design, had the highest failure rate. The H-Bridge had a failure rate that was lower than the microcontrollers but higher than the voltage regulator and the charging IC. This is mainly because of the higher operating temperature and larger number of pins. The last two parts analyzed, the BQ2002 and the Murata OKR-T had relatively low failure rates. The design could be made more reliable if a larger microcontroller was used instead of two. For example, a change to a 64 pin microcontroller would increase the MTTF from 27 years to 37 years. Other refinements that would reduce the rate of failure would be to use heat sinks on the H-bridge and other components to reduce the operational temperature.

5.2 Failure Mode, Effects, and Criticality Analysis (FMECA)

The component failures will be categorized as low, medium and high criticality. A "High" criticality level is a failure that has the potential to injure the user or others and the failure rate should be less than 10^{-9} failures. A "Medium" criticality level represents a failure that can permanently damage components of the device and the failure rate should be less than 10^{-7} failures. A "Low" criticality level describes a failure that would change the functionality of the device or affect its performance without permanently damaging components. The acceptable failure rate for Low criticality failures is less than 10^{-5} failures.

The first functional block to be analyzed is the first microcontroller. The microcontroller is in charge of controlling the camera servos, reading the right wheel encoder, accelerometer, as well as the range sensors. The possible failures are listed in table 1.

The second functional block is the second microcontroller. It controls the H-Bridge, reads the compass data, and the wheel encoders. There are a couple of failures that can lead to injury like a failure of the PWM pins PT0-PT1 or the pins AN0-AN3. The third functional block is the 3.3V regulator and the fourth functional block is the Murata 5V regulator which powers the microcontrollers, IR sensors and servos. The only way the failure of the Murata regulator can lead to an injury is if it shorts the power and ground traces. The fifth block is the battery charging circuit which charges the NiMH battery. This block has two possible critical failures: BQ2002 fails and overcharges the battery, or the transistor that turns off the current source fails which leads to the battery being overcharged also. The last block is the voltage level translator, which enables the communication between the 5V microcontrollers and the 3.3V accelerometer and compass.

5.3 Summary

The safety and reliability analysis of the project resulted in a MTTF of 27.4 years. The most likely parts to fail are the two microcontrollers, H-Bridge, Murata 5V regulator and the BQ2002 battery charging chip. Out of all the failures a few were high criticality which could result in injury to the user but the chances of them happening are very small.

6.0 Ethical and Environmental Impact Analysis

The Autonomous Targeting Vehicle's main design goal is to navigate to a given location and to follow a target. Before the ATV can become a commercial product, the ethical and environmental impact of designing such a product must be considered. The IEEE Code of Ethics calls the engineers "to accept responsibility in making decisions consistent with the safety, health, and welfare of the public, and to disclose promptly factors that might endanger the public or the environment" [1]. Because our project contains a webcam and tracking software, relevant ethical issues consist of protecting the privacy of others. The ATV, weighing over 5 pounds, consists of a large aluminum chassis and sensors to detect obstacles. In case of sensor failure, we must also take into account the damage that the large robot can cause. Environmental damage can be caused by improper disposal of electronic components in the ATV. Lastly, steps can be taken to reduce the power utilization of the ATV during normal operations.

6.1 Ethical Impact Analysis

The safety of the operation of the ATV under normal and failure mode operating conditions must be ensured to the user. In this section, the ethical issues of the ATV under the codes of ethics of both the Association of Computing Machinery (ACM) and the Institute of Electrical and Electronics Engineers (IEEE) will be discussed.

The ACM Code of Ethics asks to "respect the privacy of others" [2]. One main feature of the ATV – following a target – could be maliciously used by hackers to disturb the privacy of others. To gain access to the controls of the robot, the user only needs a WiFi-capable laptop. Once connected via WiFi, there is nothing to prevent the user from maliciously using the robot to secretly spy on someone or do other such illegal things. Moreover, the most important software that processes images and parses GPS data are on the Atom board. The programs running on the Atom board can be easily manipulated by a virus. One way to prevent malicious access to the device is to protect the WiFi connection with a password. Another way is to replace the Atom board with a more targeted system processing chip that only processes image and GPS data and cannot be reprogrammed.

The IEEE Code of Ethics asks to "avoid injuring others, their property, reputation, or employment by false or malicious action" [1]. Currently, by walking next to the robot, we estimated it travels at the speed of about 1m/s (2.24 mi/hr). This speed is slower than the average walking speed of an adult, 1.39 m/s (3.1 mi/hr) [3], so chance of collision with a person is minimal. But the robot is quite bulky at 5+ pounds and is made of sturdy aluminum with large wheels; thus, it can easily damage a child or a small pet. The robot has been designed to detect still objects and cruise around them. Thus, before it can go on the market, it has to be tested extensively to ensure that it can respond quickly to fast-moving objects. There is a sensor onboard the ATV that detects changes in elevation, which can prevent the robot from falling down a staircase and damaging itself or someone else. In such a case, the robot would come to an immediate stop. Furthermore, testing has to be done to ensure the robot stops safely if the sensors completely fail. This would require some code that verifies the input from the sensors falls within a reasonable range.

Moreover, the current design of the ATV has exposed parts that are very sensitive and crucial to the operation of the robot. These parts include the atom board, PCB, and GPS. They can be easily damaged by spilled liquids and even during operation on a rainy day. Exposed wires are susceptible to be broken if a dog bites through them. All these sensitive components must be secured within a box cover in the final commercial product. We opted not to use a box because of the large size of our PCB and atom board. However, if we design a small PCB and use a more application specific processing chip in place of the atom board, it will be possible to secure all components within a secure box cover. Additionally, we should include a label or a message in the User Manual warning the user not to operate the device under |wet conditions or in the vicinity of animals and children.

The current location of the battery is inside the chassis under two shelves containing the PCB and the atom board. If the user of this device intends to replace the battery, he/she would have to navigate through a sea of cables such as the ones connecting the Atom with the webcam, GPS, and microcontrollers. The commercial version of this ATV should be redesigned to place the battery in a more accessible location to the user; for example, on the side or in an easily removable compartment on the underside of the robot.

6.2 Environmental Impact Analysis

Over the lifetime of the ATV, there are several potential ways that it could cause harm to the environment. Much of these are common to most electronic devices these days, such as harmful substances released during the production of the PCBs or the disposal of batteries. Some are also unique to the ATV, such as its use of power when operating. The discussion that follows below will focus on how the ATV impacts the environment and the steps that can be taken to mitigate the effects.

6.2.1 Manufacture

Several pollutants are released into the water during the fabrication process of the PCB. A common pollutant is the copper released into wastewater [4]. Pollutants are also released into the air from processes such as drilling, routing, sanding, and other board preparation processes [4]. Some pollutants are even sent to landfills [4]. Hazardous waste is generated from almost every step in the manufacturing process. While most manufacturers today have found various methods to clean the water and the air from these pollutants [4], we can always do our part to reduce the amount of pollutants in the first place. The most obvious solution is to reduce the size of the PCB. Though our robot is large and can easily accommodate a very large PCB, there are several ways to make the overall size smaller. For example, instead of using the 9S12C32 microcontroller module, we could place the microcontroller directly on the PCB, eliminating the need for large DIP sockets. Also, our PCB became large because we wanted physical distance between high noise, high voltage circuitry and sensitive analog circuitry. If we place such circuitry on its own PCB, then several small PCBs rather than a very large one would suffice, reducing the contribution to pollutants.

Next, the sturdy chassis of the ATV is made of aluminum. During production of the chassis, care must be taken to ensure that only recycled aluminum is used. This will be environmentally friendly because recycling aluminum uses only 5% of the energy required to create new aluminum and saves 95% of the green house gas emissions from new aluminum production [5].

6.2.2 Normal use

During the normal usage of the device, the main concern is its use of power. We have already taken steps towards reducing the use of power by separating the battery charging circuitry. During normal operation, none of the components on the battery charging circuitry will be powered.

As for power consumption, motors operating at maximum of 1.5A at 12V uses 18W of power. The Atom board operating with 2A at 12V uses 24W of power. Along with all the other components in our device, we can approximate the power usage to about 50W. This is typically the power used by a 19" color TV at home. Even though at first glance this might seem a less than significant use of power, at this rate, the robot can operate for only 1.2 hours with one charge of our 4200mAh battery. Therefore, taking steps to reduce this even further can prolong the life of our battery and also make our device usable for longer tasks. The best solution to improve the power consumption is to create a smaller and lighter robot. Then, we can use smaller motors that need less power to keep the robot running. One way to make the robot lighter is to replace the large Atom board with a more targeted chip, as was mentioned earlier. This will not only reduce the weight, but also reduce power consumed because the Atom board currently operates at 12V and draws 2A of current. Other steps can be taken to reduce the overall current draw and the operating voltage of the device so that a smaller, lighter battery can be used.

6.2.3 Disposal/Recycling

Being an electronic device, there are several components in the ATV that can be hazardous and cannot be disposed at a regular landfill. The best method to ensure proper disposal and/or recycling of the ATV is to have the customer return the obsolete product back to the company selling the ATV. The company can even offer monetary compensation to encourage consumer to participate in recycling and make it convenient for them to do so. This would help the environment and allow the company to either reuse parts or discard them to proper landfills in bulk. The options for recycling/reusing each major component of the ATV will be discussed below.

First component is the Nickel Metal Hydride (NiMH) rechargeable battery. Our 12V 4200mAh battery pack by Tenergy Corporation has a long life cycle of up to 500 charges [6]; however, it will eventually need to be disposed of. While most NiMH batteries are considered environmentally friendly, its main derivative – nickel – is considered semi-toxic [7]. NiMH also contains and electrolyte that, in large amounts, is hazardous to the environment [7]. A single NiMH battery can be discarded with household wastes [7]; however, the battery pack of the ATV has ten NiMH batteries. This pack must be disposed at a secure waste landfill.

Another component that needs to be disposed properly is the PCB. Printed circuit boards contain precious metals such as gold, silver, and platinum, and base metals such as copper, iron, and aluminum [8]. PCBs fabricated in the United States even contain lead because "led is permitted for electronic use" in America [9]. The best option for keeping PCBs out of landfills is to recycle them. Recovering the precious metals can be especially beneficial to the PCB manufacturers. However, there is very little use for the non-metallic components of a PCB. In 2008, innovative researchers in China recycled an entire PCB into a new useful item that was used to make park benches and fences [10].

The remaining major components are the chassis, motors, tires, camera, and sensors. All these parts can easily be reused by the company producing the ATV in their new products. The parts that no longer work can be sent to an electronics recycling factory that is able to turn these materials into new materials for use in different products. As mentioned earlier, recycling aluminum is environmentally friendly, thus the chassis can be sent to an aluminum recycling center.

The owners of the ATV should be provided with a manual of environmentally friendly usage of the ATV. The manual should include tips to reduce power consumption; for example, by turning off the device when not in use. It should also include instructions on how to dispose components properly once the ATV becomes obsolete. Owners should also be encouraged to follow federal and state guidelines on recycling of electronic parts. Ideally, the company selling ATV should implement a recycling program where owners of ATVs can send back expired ATVs for reuse of parts.

6.3 Summary

This report has discussed the ethical and environmental impacts that the ATV could have on the environment. On the ethical perspective, this includes protecting our software and hardware against malicious users so that the ATV cannot be used to invade the privacy of others. Also, the designers of the ATV are obligated to test the device extensively in failure modes to ensure it will not injure animals or other persons. On the environmental perspective, this includes ensuring that the components of the ATV are manufactured using materials and mechanisms that impact the environment the least and also disposing the ATV in a way that does not release hazardous materials into the environment.

7.0 Packaging Design Considerations

The Autonomous Targeting Vehicle is an autonomous wheeled vehicle that can navigate to targets and visually track them, all while avoiding obstacles in its path. The main aspects for the packaging of this vehicle include a chassis, motors, wheels, and room for various peripheral components. Such components include a GPS module, sonic range finder, infrared proximity sensors, digital compass, webcam, two microcontrollers, Atom board, battery, and a PCB to allow communication between all components. Two proximity sensors will be mounted at the front, one to detect objects in front of the vehicle, and one to detect differences in ground elevation. A webcam will be mounted on top of the vehicle so that it can visually lock on a target and navigate to it. The other peripherals will be housed inside the vehicle frame. The main goal of the packaging is to provide a relatively small housing that will allow the vehicle to travel in small spaces, and also be of adequate height to give the webcam a wide range of vision.

7.1 Commercial Product Packaging

Autonomous traversing vehicles are not readily available commercially, but rather are being developed by the military and science organizations for defense and exploration purposes. Two commercial products that are similar in look and functionality to our Autonomous Targeting Vehicle are the Army’s Autonomous Platform Demonstrator (APD) and NASA’s Urbie Rover.

7.1.1 Product #1

The U.S. Army's Autonomous Platform Demonstrator (APD) [1], despite weighing 9.6 tons and measuring at about 15 feet in length, operates in a very similar fashion to our Autonomous Targeting Vehicle. The Autonomous Platform Demonstrator is designed to autonomously navigate to a location using GPS waypoint technology, avoid obstacles, and do so with speeds of up to 50 miles per hour [1]. The APD is designed with a large metal body, six large all-terrain wheels, and several antennae for the sensors [1]. Although the large size of the APD allows it to travel at relatively high speeds, it does not allow the vehicle to stealthily scout a location for surveillance. Also, its large size does not allow it to fit in tight spaces, which is a highly desired feature of our project. The APD runs on six wheels and includes an advanced suspension system, which enables it to run over smaller objects and climb a one-meter step [1]. Our project only features four wheels, and does not support a suspension system. It would be a great addition for our vehicle, but with the small size limitations, there is no room for the added suspension. The object-detection sensors on the APD are located in the four-meter antenna on top of the vehicle [1], whereas our project features two IR proximity sensors on the front of the vehicle, and a single sonic range finder on the webcam. One very nice feature about this vehicle is that all its electronics are concealed inside the frame of its body, which we originally planned to adapt in the design of our project. However, because of the need for a tremendous amount of airflow to cool the Atom board and PCB, we had to settle with a more "open" package design. This also enabled us to easily troubleshoot problems and fix them without being confined to the space of a small package. The packaging feature that makes our project unique is the added webcam for visually tracking an object.

Figure 2.1: U.S. Army's APD

7.1.2 Product #2

The Urbie Rover, developed by the robotics team at NASA, is an autonomous traversing vehicle that is equipped with two Pentium processors, a GPS receiver, several sensors, a laser range finder (LIDAR), an omni-directional camera, and a binocular stereo camera pair [2]. It is designed to autonomously navigate in contaminated areas where human presence is very dangerous, and to explore the surface of Mars [2]. It is designed with a low-profile metal chassis, 360-degree revolving arms, several sensors and cameras mounted atop the chassis, and is made to be as lightweight as possible. Its small size and weight make it very portable and

Figure 2.2: NASA's Urbie Rover

able to fit into the smallest of spaces. One nice feature about the design of the "wheels" on the Urbie Rover is that it has the capability to climb stairs and climb over obstacles. The two arms at the front of the Rover rotate 360 degrees, allowing it to easily climb over objects and even turn itself upright if it flips over [2]. Our vehicle has four separate wheels with no suspension, restricting movement to flat ground only. The negative aspect about the Rover's small size is that it limits the field of vision of the robot. Our project features a webcam that sits higher up on the robot, potentially allowing it to have a greater area of vision. We planned to adapt the small "footprint" of the Urbie Rover, but add additional height for greater vision area. The distinguishing feature of our project is the four, large wheels, as opposed to the continuous tracks of the Urbie Rover.

7.2 Project Packaging Specifications

The ATV's main chassis is made of laser-cut (pre-cut) Lexan panels, with heavy-duty aluminum structural components. Its dimensions are 9.5" long by 8" wide. In order to accommodate room for the atom board, GPS, range finders, compass, webcam, and PCB, three "shelves" were installed on top of the main chassis, each separated and supported by four 1½" standoffs. The "shelves" give the vehicle an "open" packaging design, allowing greater airflow to the Atom board and PCB. It also enables easier troubleshooting by providing an open "workspace", rather than being confined to a small package. Schematics for the packaging design can be viewed in Appendix B.

A camera is mounted on the top of the vehicle, which gives it the best possible range of vision. A sonic range finder is attached to the camera to allow it to sense objects in the direction it is facing. As the vehicle navigates around an obstacle, the camera rotates around, allowing the vehicle to detect when it has traveled past the obstacle and is safe to head toward the original destination. The other two range finders are located on the front of the vehicle, one to detect obstacles from the front, and one pointing down at an angle to detect changes in ground elevation (such as a pothole). An external antenna is attached on the back of the vehicle to enhance the WiFi signal on the Atom board.

Four 200-RPM gear head motors are used to drive the vehicle. Attached to the motors are four wheels made of very durable rubber, with rims made of a sturdy nylon material. These wheels allow the robot to carry large payloads. With a total weight of just over 6 lbs., the selected high-RPM motors along with the sturdy wheels provide more than enough support for maneuverability.

7.3 PCB Footprint Layout

Because most of the peripherals utilized by the ATV are connected externally, the PCB design is rather simplistic. Two H-Bridges are used to control the speed and direction of motor movement for each side of the vehicle. A simple battery charging circuit will be used that includes a thermistor for providing overheating protection for the battery. Because two wheel encoders will be used to determine small changes in position, we require two pulse accumulators to retrieve the data. In order to accommodate for this, we used two Freescale 9S12C32 microcontrollers. Using two microcontrollers also gives us the freedom to split the workload between controlling the servos for the webcam, controlling the motors for movement, and acquiring data from the various sensors. The two microcontrollers interface with headers on the perimeter of the PCB, allowing for a clean and simple connection for all the needed peripherals. Three voltage regulators are needed: 12V, 5V, and 3.3V. The final dimensions of the PCB are 5” x 6.5”.

7.4 Summary

The Autonomous Targeting Vehicle is designed to autonomously track and navigate to targets while avoiding obstacles in its path. It will feature a small footprint to allow it to stealthily scout locations for surveillance, as with NASA’s Urbie Rover [2], yet will be tall enough to give the webcam a wide range of vision. While the original intention was to model the packaging after the U.S. Army’s APD [1] with all of the various electronics neatly concealed inside the body, we decided to implement more of an “open” package design to allow greater airflow to the Atom board and PCB. Because of the numerous “off chip” peripherals utilized by the ATV, the PCB will be designed as an interconnection network between all the sensors, motors, microcontrollers, and Atom board. It will feature several voltage regulators, along with headers to create clean connections to each of the peripherals on the vehicle.

8.0 Schematic Design Considerations

The Autonomous Targeting Vehicle (ATV) is an autonomous robot with two operation modes. In the first mode, the vehicle will use GPS to determine its current location and be able to autonomously navigate to a specified location using sensors to detect and avoid obstacles while also streaming video to the users' computer. An accelerometer, a compass, and wheel encoders will be used in addition to the GPS to provide improved precision. In the follow mode, the user will left mouse click on a target and the vehicle will use the webcam to follow the target within a specified distance.

8.1 Theory of Operation

The robot will be turned on by using a simple switch located in the rear of chassis. The switch closes the circuit to the 12 V NiMH battery, which powers all our components. The battery has a maximum discharge current of 40A which is well above our required 5A. This supply voltage was chosen because both our Atom board and the four motors require 12V unregulated voltage to operate. The maximum current drawn by the Atom Board is 2A and the max current drawn by the motors under load is 1A. This leaves 2A for all the other components and safety margin.

The first voltage regulator is a Murata OKR-T switch mode regulator which takes the 12V input and lowers it to 5V. The sonic range finder, the two microcontrollers, the two webcam servos, two IR sensors and the two wheel encoders all operate at 5V.

A LM317 linear voltage regulator is used to further drop the voltage from 5V to 3.3V. Since the current required to power the accelerometer and compass is very low the lower efficiency of the linear regulator is not an issue.

The motors will be controlled by a STMicroelectronics L298 Motor Driver Dual H-Bridge with a 4A total output current. The left two motors will be connected in parallel to one side of the H-bridge and two right motors on the other. Since the speed of each motor individually does not have to be changed individually one Dual H-Bridge is enough for our differential drive system. The speed of the motors will be controlled by sending two 100 Hz PWM signals to the enable pins of the H-bridge and adjusting the duty cycle. In order to control the direction of the motors the microcontroller will send two logic inputs for each of the pair of two motors. In order to prevent the H-Bridge from being damaged the software will ensure that the H-Bridge stays in breaking mode for at least .5 seconds before reversing direction.

The microcontroller used to control the motors, servos and acquire the data from the sensors is a Freescale MC9S12C32. It will operate at a 24 MHz frequency which will enable us to read all the sensor data, control the motors and send the sensor data on the serial port to the Atom Board at a 38400 baud rate. Two microcontrollers will be used since we need more PWMs, and timer pins. One microcontroller will handle the webcam servos, the left wheel encoder and the sonic range finder. The second microcontroller will handle the four wheel motors, the right wheel encoder as well the compass, accelerometer and IR sensors.

The project will use 3 distance sensors which will be powered by the 5V power supply. The first is a MaxBotix XL-MaxSonar-EZ3 Sonic Range finder which outputs a voltage corresponding to the distance of the obstacle. This is connected to the analog-to-digital (ADC) module of the microcontroller. This range finder will be mounted on top of the webcam and will be used to scan the surroundings in normal operation mode as well as keep a certain distance from the object being followed in follow mode. The other two distance sensors will be Sharp GP2Y0A02YK0F IR sensors . One of the IR sensors will be placed in the front of the car facing down at a 45 degree angle in order to detect holes and drops. The second IR sensor will be placed in the front of the vehicle for frontal obstacle detection. Both of the IR sensors will also interface with the microcontroller through the ADC module. Pins AN0-AN2 were chosen for this, simply for convenience.

A Honeywell HMC5843 Magnetometer will also be used. It uses I2C protocol to communicate however since the chosen microcontroller does not have I2C the protocol will be implemented in software using GPIO pins. Since the microcontroller is operating at 5V a logic level translator will be used. The choice to not put the compass on the PCB was made because of the possibility of EMI.

A Bosch BMA180 accelerometer is also used to help with determining the change in position of the robot. It interfaces with the first microcontroller using SPI and a 3.3V to 5V logic level translator.

The project will also use a GlobalSat EM-406A GPS module connected to an evaluation board in order to be able to interface it to the Atom board via USB.

The two wheel encoders operate at 5V and each will interface to one microcontroller using a Pulse accumulator pin and a timer pin.

8.2 Hardware Design Narrative

The Pulse Width Modulation, Serial Communication Interface, Timer, and Analog-toDigital Conversion peripherals of the microcontroller will be used.

The PWM peripheral will be used to control the speed of the motors by turning the H-bridge on and off and thus controlling the amount of time that the motors receive power. The two sets of two motors will require one PWM pin each which will enable us to control the speed as well as the turning speed of the robot.

In addition to controlling the motors, the PWM peripheral will also be used to control the two servos for the webcam. This would usually require only two PWM signals, however, since the servos are very sensitive in changes of the duty cycle of the PWM signal, the PWMs are operated in 16-bit mode. In order to achieve this, two PWM signals are concatenated into one which increases the PWM ports used for the servos to four. The ports PT0-PT3 were chosen for the servos on the first microcontroller and PT0-PT1 for the motors on the second microcontroller.

The motor PWM signals are calculated by the Atom board and sent over SCI to the microcontroller. In order to send the correct PWM signals, the Atom board receives the encoders, accelerometer, compass, and sensor data, from the microcontroller over SCI and together with the GPS data it calculates the correct PWM signals to avoid obstacles and reach the destination. The target frequency of the sensor data being sent to the Atom board is 20 Hz. This was chosen so that the position change of the robot is kept as precise and updated as possible.

The servo PWM signals are chosen by the microcontroller based on the pixel values of the target which are also sent over SCI from the Atom board. The reason why SCI was chosen is because it is simple to code in C and already available on our microcontroller. The image processing of the target is done on the Atom using the OpenCV library and tracks whatever the user clicks on.

The Timer peripheral will be used to interface the wheel encoders and keep track of the number of revolutions as well as the rotation direction of the wheel.

The last subsystem used is the Analog-to-Digital peripheral. This enables the microcontroller to read analog voltage input from the Sonic rangefinders, and the two IR sensors which enable the robot to detect obstacles accurate within an inch.

The robot will also feature three LEDs to display when it is powered on and in what mode it is operating. General purpose I/O pins will be used for the LEDs and the pushbuttons.

8.3 Summary

The ATV project is an autonomous vehicle that uses a variety of sensors together with a GPS to navigate to a chosen GPS coordinate while also avoiding obstacles and sending real time video to the users laptop. Using the video the user will also be able to choose a target which the vehicle will follow. This report summarizes how this will be accomplished and explains all the major peripherals, supply voltages, operating frequencies and required interfaces used in this project.

9.0 PCB Layout Design Considerations

The Autonomous Targeting Vehicle is a mobile robot platform that can navigate to targets while avoiding obstacles in its path and visually track objects. The main components of the design includes a sonic rangefinder, two IR sensors, a digital compass, two wheel encoders, an h-bridge driving four motors, two servos to control the camera, a GPS, a webcam, two microcontrollers, an atom board, and a battery pack. A majority of these components will be located externally to the PCB on the chassis. The PCB will contain the two microcontrollers, the h-bridge, and many headers to external components. Also, the PCB will contain two power supply circuits and battery charger circuit that will be discussed in Section 9.3.

This document will discuss various design considerations of the PCB. Some of the design considerations include overall placement of components, trace widths, EMI interference, and separation of digital, analog, and power circuits. A carefully designed PCB will conveniently keep all logic signals close together and ensure the correct function of the design. On the other hand, a bad design could result in hard to debug or inoperable environment.

9.1 PCB Layout Design Considerations - Overall

The board will be separated into four areas: power supply, digital circuitry for the microcontrollers, analog signals for the sensors, and the motors. Components belonging to the same functional block will be kept together to reduce trace lengths. A big concern in this design is to keep the sensitive analog signals away from the power circuitry and the motors. The switching noise from step-down voltage regulators and the motors can add noise to the analog inputs and can cause large fluctuations in important sensor data.

When the PCB is placed inside the chassis, it will be surrounded by four motors from all sides. The best solution is to place the analog input connectors towards the middle of one edge of the board so that they will not be too close to either motor. The H-bridge circuitry driving the motors will be placed on the opposite edge of the PCB so that EMI from switching noise will not add noise to the analog signals of the sensors. The power supply circuitry will be placed adjacent to the motor circuitry. Refer to Figure 3 in Appendix for placement of functional blocks. To further ensure that sensitive signals will be placed well away from EMI, the team is considering placing the PCB on top of the chassis so as to keep analog inputs physically far away from the motors.

Connectors to devices off the board will be placed at the outer edge so as to keep the middle of the PCB free of clutter. The microcontrollers will be placed in a fashion that provides the most direct access to all components. The microcontrollers have their own breakout boards and will be connected to the PCB via a DIP socket. The accelerometer also has its own breakout board and will be housed on the PCB via headers.

General concerns include the size of traces, location of vias, and the size of the PCB. The power and ground traces will be in the range of 40-60 mils, the logic signals will be 25 mils and any signals that needs to be necked between pads will be 10-15 mils. Attention will be paid to avoid 90-degree or acute angle turns of traces. Vias will be placed to connect power or logic signals between the top and bottom layers of the PCB. Since each device on the PCB will need both power and logic connections, there is a high probability that these signals will cross paths. Vias will ensure that these signals will cross on different layers of the PCB. Concerns for vias include keeping them far enough away from pins to avoid shortage and to minimize vias in general. Vias were decided to have a drill size of 40 mils and annular ring of 15 mils. Adequate spacing will be left at each corner of the PCB for mounting holes. Mounting holes will have a drill size of 125 mils and an annular ring of 15 mils. Drill sizes and pads sizes for each component IC will be determined as specified in their respective datasheets. The size of the PCB is not a big constraint because the ATV has a fairly large chassis. The preliminary PCB layout fitted into a 5” x 6.5” board.

9.2 PCB Layout Design Considerations - Microcontroller

The microcontroller chosen for the ATV project is the Freescale 9S12C32, which comes with an internal oscillator as well as an external oscillator on its breakout board.

The placement of the microcontrollers is important as they should have the shortest path to the peripherals they are communicating with. Also, correct orientation will ensure that traces do not cross over each other. If crossing cannot be avoided, vias will be used to transfer signals to the bottom layer. Care will be taken to ensure that pin selections on the microcontroller are well spread out to avoid crowding of traces in one region. All the pins of the microcontroller will be connected to headers. Each microcontroller will have a decoupling capacitor between V_{CC} (pin 29) and V_{SS} (pin 4). The 9S12C32 microcontrollers will be operating at 24MHz; thus, they will be decoupled by a 0.01μF capacitor on the PCB. Both microcontrollers will also have their own reset pushbuttons placed close by on the PCB.

The reason for using two microcontrollers is to collect data from the two wheel encoders and drive the motors and servos. Wheel encoders need a pulse accumulator, but the 9S12C32 only has one pulse accumulator each. The two double precision servos will need two PWM channels each and each motor needs one PWM channel; this is a total of six PWM channels. One microcontroller has only five PWM channels. Therefore, two microcontrollers are necessary.

One microcontroller will communicate with the compass using the Port T GPIO pins. The two motors for the right and left side of the robot will be controlled using the PWM peripheral. One of the wheel encoders will be connected to the pulse accumulator on TIM port.

The other microcontroller will communicate with the sonic rangefinder and the two IR sensors using the ATD peripheral. It will control the servos for the camera using the PWM peripheral. The second wheel encoder will be connected to this microcontroller's pulse accumulator. Both microcontrollers will be communicating with the ATOM board using their SCI peripherals.

The microcontrollers will be housed on the PCB via a 40-pin 0.600" DIP socket. Since pin pitch of the DIP socket is 0.1" and recommended drill hole size is 0.035"-0.043", there will plenty of space for logic signals of trace width 25 mils. This will leave about 30 mils of clearance between each pad of the DIP socket.

Even though the microcontroller is not providing high currents to any devices, the biggest concern for the microcontroller is protection from the motors. Optical isolators will be placed for all four logic signals and two PWM signals between the microcontroller and the H-bridge circuit.

9.3 PCB Layout Design Considerations – Power Supply

The main source of power will be a 12 V NiMH rechargeable battery [8]. The battery will be connected to the PCB via a connector. Two power supply circuits on the PCB will generate regulated voltage levels for several devices. A 3.3V is needed to power the compass. A 5V supply will power two microcontrollers, sonic rangefinder, and two IR sensors. Both the motors and the atom board will be powered at 12V off the PCB. The 12V NiMH battery will be recharged with battery recharging circuit that will use the Texas Instrument bq2002 charge controller.

The 12V unregulated supply will be stepped down to regulated 5V using OKR-T/3 adjustable 3A DC/DC converter [11]. This circuit will have a 0.1µF bypass capacitor and a 100µF bulk capacitor. The 5V regulated voltage will be in turn stepped down to 3.3V using an LM317 adjustable regulator and a couple of resistors. This circuit will have 1.0µF bypass capacitor. Bypass capacitors are needed to decouple high-frequency noise and will be placed as close to the power terminals as possible. The team decided that the LM317 regulator will not need a heat sink as the digital compass draws only 0.9mA [3] and the accelerometer draws only 650µA [1]. Also, the size of the voltage regulator is 240 mils by 400 mils [2], which should be sufficient to dissipate any little heat created.

The battery charging circuit will consist of the Texas Instruments bq2002 charge controller IC, an LM317 adjustable regulator, a MOSFET, and a thermistor. The battery will be charged via a constant current source and a voltage measurement closed loop to monitor a negative change in voltage. When the bq2002 detects a negative change in voltage and a rise in battery temperature via the thermistor, it will use the MOSFET to turn off the current source. The recommended battery charger for the 12V 4200mAh NiMH [8] battery is rated at 7.2V – 12V and uses a charging current of 1.8A for battery packs above 2000mAh [9]. Since some laptop chargers are rated at 12V+ and 3.3A+, the team decided that a laptop charger would suffice as the input power supply for the battery charging circuit. The laptop charger would connect to a barrel jack on the PCB.

Each of the motors operates at 12V and draws 1 - 1.5A. They are driven by a L298 dual full-bridge driver. Schottky diodes will be placed to shunt the back EMF generated by the motor so that if the motor's power is suddenly cut off, there won't be a reverse voltage spike. The motors operate at 200 RPM [6]; a 1.0µF will be used for noise filtering. Also, the microcontroller pins driving the H-bridges will be protected by 4N33 optical isolators.

Decoupling capacitors provide the current needed by an IC by responding very quickly to changing current demands. They reduce the load on the power lines and removes unwanted glitches in the power system. The two microcontrollers will be decoupled by high-frequency, multi-layer ceramic capacitors placed as close to the microcontrollers as possible. As stated in section 3.0, the 9S12C32 microcontrollers will be operating at 24MHz; thus, they will be decoupled by a 0.01µF capacitor. A 100µF tantalum electrolytic bulk capacitor will be placed close to the power terminal to recharge the decoupling capacitors. The value of the bulk capacitor was chosen to be larger than the sum of all the decoupling capacitor values. As suggested by the Motorola application note, a small 0.1µF ceramic disk capacitor will be placed next to the bulk capacitor to decouple high frequency noise at the terminals [7].

The power and ground traces will be connected as directly as possible. The width of the power traces will be 40-60 mils as suggested in the lecture notes. The motors will draw on average 1A per motor. According to the Trace Width Calculator on the 4PCB website, to sink a 1.5A of current on a standard 2-layer 1 oz/ft^2 thick board [4, 10] at 25°C ambient temperature, the trace needs to be 20.7 mils thick in air [5]. Thus, 40-60 mils power trace will be more than sufficient. Digital and analog grounds will be only be connected at a single point close to the power terminals to reduce common impedance coupling among subsystems. Also, effort will be put into making power and ground traces parallel to each other as much as possible.

9.4 Summary

Section 9.0 focuses on the issues of ATV project on designing the PCB. The overall considerations of the PCB in Section 9.1 discussed placement of components into separate functional blocks, trace sizes, and PCB board size. Issues more directly related to the microcontroller were discussed in Section 9.2. These include the placement of the microcontrollers, the need for two microcontrollers, connections to peripherals, and connection to PCB via a DIP socket. Lastly, Section 9.3 considered issues relating to the power supply. The battery recharging circuit and the power regulation circuits were explained in detail as well as the justification of the trace widths. Details of decoupling, bypass, and bulk capacitors were also stated. With adherence to design specifications and proper PCB routing techniques, it is possible to design a fully functional PCB free of errors.

10.0 Software Design Considerations

The design is an autonomous wheeled vehicle that can navigate as well as visually track and follow targets. This vehicle will use GPS to determine its current location and will be able to autonomously navigate to another location using sensors to detect and avoid obstacles. To improve accuracy, this vehicle will use the Kalman filter algorithm to perform sensor fusion and "dead reckoning" using the information from the accelerometer, compass, and wheel encoders. This will also allow the vehicle to navigate when a GPS signal cannot be received. Additionally, it will be able to visually track a target using the Lucas-Kanade optical flow algorithm[1]. The robot will use an Intel Atom board to perform the image processing and Kalman filter algorithms. It will also allow the user to connect remotely through a wireless connection and control the robot through a GUI interface. A pair of Freescale 9S12C32 microcontrollers[2] will communicate with the Atom board through a serial interface. They will send sensor data to the Atom board and receive from it instructions which they will use to control the camera servos and wheel motors.

10.1 Software Design Considerations

The design of the project software is largely guided by the functions it must perform. The main functions which must be performed are navigation and tracking. Each of these can be broken down into several subsystems.

Navigation can be broken down into (1) a user interface for entering the desired destination, (2) a system for determining the current location and trajectory, (3) a way to detect obstacles and determine their location, (4) a system for finding a path from the current position to the destination, and (5) a system for controlling the wheels in order to follow the chosen path.

Target tracking can be broken down into (1) a user interface for choosing a target to track, (2) performing video processing to track the target in the video, and (3) a system for aiming the camera at the target.

These functions must be performed at a speed of roughly 20Hz given the processing power, memory, and data transfer limits of the hardware. Because the micro-controllers have limited RAM (2KB each) and a slower clock (up to 25 MHz) than the Atom board (1.6Ghz), the most memory and processing intensive jobs are performed on the Atom board. The micro-controllers are left with the relatively simple jobs of reading the sensor data, sending it to the Atom board, reading packets from the Atom board, and controlling the motors and servos based on the incoming packets.

Two serial interface cables are used for communication between the Atom board and the two micro-controllers. They communicate using four distinct packet types. The first microcontroller sends a 14 byte packet containing the accelerometer, left wheel encoder, and rangefinder data. The second microcontroller sends an 11 bytes packet containing compass and right wheel encoder data. The Atom board sends one type of packet to each micro-controller: a 7 byte packet to the first microcontroller, and a 6 byte packet to the second microcontroller. The first microcontroller receives the pixel location of the target in the image, which it uses to point the camera at it. The second microcontroller receives the desired wheel power and direction. At 20 Hz, this adds up to 3360 bits per second on one line, and 2720 bits per second on the other. This can be easily accommodated, since serial lines can run at speeds many times higher than this, as such 38400 baud.

Because of the simple nature of the code on the microcontrollers, and because extremely fast latency is not required, a polling loop with interrupt-driven flags is used for the microcontroller code structure. The main loop consists of reading packets from the SCI input buffer when the packet ready flag is set, followed by setting the servo or motor controls. Next, the sensors are read, and a packet is sent out on the SCI with the sensor data.

The design allows for in-circuit debugging in two ways. The first is the ability to use the BDM connectors on the microcontroller boards to perform debugging. Second, the code running on the Atom has the ability to display any information necessary on the screen while running.

The memory layout, external interfaces, and module initializations of the microcontrollers are illustrated in Appendix G.

10.2 Software Design Narrative

The user interface allows the user to view the video from the camera on the robot and choose a target in the video to track. It also displays a map of the area surrounding the robot, along with the current and past positions of the robot, as well as its current planned path, and allows the user to add waypoints by clicking on the map.

The user interface accomplishes the graphical functionality largely through using the functions provided by OpenCV [3]. OpenCV includes the ability to display and process video and images, as well as provides the ability to create a mouse clicking interrupt. When the user clicks on the video, the position of the mouse pointer in the image is recorded, and sent to the Object Tracking block.

The map display is accomplished through several steps. Beforehand, screenshots were taken of the Google Earth [4] map of the area, and the boundaries of these images were recorded. Functions were created for converting between world coordinates (latitude, longitude), and the pixel coordinates of an image, given the world coordinates of the image's boundaries and its size. Combining these functions with the Google Earth map and its boundaries, it becomes possible for the program to determine the real-world location of a pixel in the image. Thus, when a user clicks on the map, the interface can add the corresponding real world coordinates to a queue of waypoints which will be used by the Pathfinder Block to make the robot navigate to the clicked location. Similarly, it is possible to superimpose a dot, line or icon indicating the current position of the robot, its desired path, or an obstacle onto the map.

The Object Tracking block uses a function provided by OpenCV to perform the Lucas-Kanade Optical Flow algorithm [1]. This algorithm uses the changing of the pixels from frame to frame to estimate the direction and speed the image is flowing at each pixel in the image. This is used to estimate the location of a target from frame to frame given an initial position. Target locations are initialized through the user interface, and are then maintained by the tracking algorithm. Many points can be tracked at once to increase the robustness of the tracking, since while a single point may stray from the true target over time, the average of many points does a good job of tracking the object. The pixel coordinates of the average of all current points of interest is sent over the serial port to the microcontroller which controls the camera servos.

The GPS parser was obtained online [5]. It takes in an array of characters, and checks whether they form a valid GPS NMEA packet, if so, it pulls out the pertinent information, and stores it in a data structure which can be read later.

The OpenCV library contains an implementation of a Kalman filter. It is used to perform create an estimate of the state of the robot through use of a kinematic model, together with sensor fusion. A Kalman filter is an algorithm used to estimate the state of a linear system given a model which is subject to error, using measurements which are also subject to error. The Kalman filter operates in two stages. First is the prediction phase, where it uses the current estimated state of the system, combined with the inputs to the system and estimates of the standard deviation of the error of the current estimated state of the system, to make a new estimate of the system state and of the error of the system state at the next time interval. Second is the correction phase, where it uses the estimated state and estimated error, combined with measurements and the estimated standard deviation of the error of the measurements, to revise the current estimated state and estimated error. This allows many redundant measurements to be “fused” together to find a better estimate of the system state than any individual measurement could provide.

It was necessary to derive a simplified kinematic model of the robot, before it was possible to use the Kalman Filter. The Kalman filter requires several matrices as input: the state transition matrix which determines the estimate in the next time interval based solely on the previous state, the input matrix, which determines how external inputs to the system will change the next state, the process error matrix, which is an estimate of the extent to which the model fails to predict the state, the measurement matrix, which determines how the measurements are related to the state variables, and the measurement error matrix, which is the estimated error of the measurements. The variables used in the state of the system are the X and Y coordinates of the robot, the direction THETA, which the robot is facing, the current speed SL and SR of the left and right wheels, and the acceleration AL, and AR, of the left and right wheels.

In the simplified model used in the estimation phase, the robot is assumed to consist of two wheels. The rate of change of THETA is determined by the ratio of the wheel speeds and the width of the robot. The robot is assumed to move in a straight line in the direction the robot is facing during the very short time interval at a speed equal to the average of the two wheel speeds. The acceleration of the wheels is estimated as the input to the motors minus a friction constant times the wheel velocity. The X and Y position are measured through GPS, the orientation THETA is measured by a compass, and the wheel speeds are measured by the wheel encoders.

The obstacle map builder uses the position and orientation of the robot, combined with the distances of the objects as measured by the rangefinders, to mark a location as occupied by an obstacle. This information is encoded in a graph structure which is used by the path-finding block.

The path-finding block is composed of slightly modified code obtained from linux.softpedia.com [6]. It is an implementation of the D* LITE [7] path-finding algorithm. The algorithm works similarly to the Dijkstra's Shortest Path algorithm [8], except that it uses heuristics to attempt to speed up the runtime in return for no longer being guaranteed the shortest path, but only a relatively short path. It also works in reverse: finding the path from the destination to the start point. This is done because it causes a huge speedup when recalculating the path after moving because it allows the information from a previous run (which is the distance of each node from the start point of the algorithm) to be reused. This block uses the current location, the desired location, and the obstacle map to find a path to the destination which avoids obstacles. This path is stored as a queue of sub-waypoints, and sent to the PID control system.

The PID control system is designed to make the robot travel from its current location to the first sub-waypoint in the queue by minimizing the difference between the orientation of the robot and the orientation which points directly from the current position to the waypoint. The speed is also controlled in order to stop at the waypoint without overshooting it. If the current sub-waypoint is not the final destination, then it is discarded from the queue when the robot approaches it, and the control system aims at the next sub-waypoint. In this way, it is able to follow the path chosen by the pathfinder.

The Servo control block on microcontroller 1 takes the X and Y pixel coordinates, and, based upon their direction and distance from the center of the screen, controls the camera servos using the PWM. It commands them to change their position so that the target is moved toward the center of the screen. In this way, the camera is made to follow the target.

The Motor control block on microcontroller 2 simply sets the PWMs and wheel direction pins to the values determined by the Atom board PID control system.

The sensor reading blocks in the microcontrollers are very simple. They consist of either simply reading the ATD pin (in the case of the range sensors), reading the Pulse Accumulator count (in the case of the wheel encoders), or sending a read request on SPI or I2C and recording the response (in the case of the accelerometer and compass sensors, respectively).

The microcontroller reading and writing from the SCI is also quite simple. Reading consists of an interrupt routine which writes the incoming data to a buffer, and writing consists of writing the desired byte to the proper register.

10.3 Summary

The ATV robot software is designed to allow a user to initiate automated tracking of a target, or initiate automated navigation to a waypoint. The tracking and navigation software is largely implemented on an Intel Atom board, while the low level sensor communication and motor and servo control is accomplished by a pair of Freescale 9S12C32 microcontrollers.

11.0 Version 2 Changes

There are a couple of changes that we would do to improve the design. The first change would be using one bigger microcontroller instead of two. This would simplify the synchronization of the communication protocol. Replacing the two microcontrollers with a larger one would also eliminate the need to send packets from one microcontroller to the other through the Atom board which would enable faster control of the motors based on the sensor data. The second change would be using two dual H-Bridges to control the four motors. In the current design, two motors are connected in parallel to one H-Bridge. The problem with this design is that if one of the wheels gets stuck, it creates a short and draws all the current from the H-Bridge preventing the second wheel from turning. The use of two dual H-Bridges would eliminate this problem as well as allow for individual control of each motor resulting in better control of the robot. The third change made would be the packaging. We would use a more enclosed design to make it water proof in order to protect the sensitive electronics inside. We would also make it more compact by fitting components more closely together.

12.0 Summary and Conclusions

Our team has successfully designed, assembled and programmed an autonomous wheeled vehicle with target tracking and following capability, as well as obstacle detection, mapping and path-finding ability. In the process we have learned a great deal about part selection, printed circuit board design, power supply design, battery charging circuitry, motor control, sensors, embedded programming techniques, inter-device communication and synchronization, graphical displays and interfaces, and about algorithms for sensor fusion and path-finding.

13.0 References

13.3 Constraint Analysis and Component Selection References

[1] "Lucas–Kanade Optical Flow Method," 2010 [Online]. Available: http://en.wikipedia.org/wiki/Lucas%E2%80%93Kanade_Optical_Flow_Method [Accessed: 2/6/2011].

[2] "MC9S12C128V1 Datasheet," 2010 [Online]. Available: http://cache.freescale.com/files/microcontrollers/doc/data_sheet/MC9S12C128V1.pdf?pspll=1 [Accessed: 2/6/2011].

[3] "3S12XD256 Datasheet," 2010 [Online]. Available: http://www.freescale.com/files/microcontrollers/doc/app_note/AN3328.pdf [Accessed: 2/6/2011].

[4] "Lynxmotion 4WD1," 2010 [Online]. Available: http://www.lynxmotion.com/p-603-aluminum-4wd1-rover-kit.aspx [Accessed: 2/6/2011].

[5] "DFRobot 4WD," 2010 [Online]. Available:http://www.robotshop.com/dfrobot-4wd-arduino-compatible-platform-encoders-2.html [Accessed: 2/6/2011].

[6] "Dagu Wild Thumper," 2010 [Online]. Available: http://www.pololu.com/catalog/product/1566 [Accessed: 2/6/2011].

[7] "20 Channel EM-406A SiRF III Receiver," 2010 [Online]. Available: http://www.sparkfun.com/products/465 [Accessed: 2/6/2011].

[8] "66 Channel LS20031 GPS," 2010 [Online]. Available: http://www.sparkfun.com/products/9891 [Accessed: 2/6/2011].

[9] "Triple Axis Magnetometer Breakout," 2010 [Online]. Available: http://www.sparkfun.com/products/9371 [Accessed: 2/6/2011].

[10] "I-Base N270" 2010 [Online]. Available: http://www.ibase.com.tw/2009/Datasheet/IB887.pdf [Accessed: 2/6/2011].

[11] "Ultrasonic Range Finder - XL-Maxsonar EZ3," 2010 [Online]. Available: http://www.sparkfun.com/products/9494 [Accessed: 2/6/2011].

[12] "Infrared Proximity Sensor Long Range - Sharp GP2Y0A02YK0F," 2010 [Online]. Available: http://www.sparkfun.com/products/8958 [Accessed: 2/6/2011].

[13] "ZOTAC IONITX-G-E Intel Atom 330," 2010 [Online]. Available: http://www.newegg.com/Product/Product.aspx?Item=N82E16813500037 [Accessed: 2/6/2011].

[14] "Digikey AP1501A," 2010 [Online]. Available: http://search.digikey.com/scripts/DkSearch/dksus.dllsite=us&lang=en&mpart=AP1501A-12T5L-U [Accessed: 2/6/2011].

[15] "Digikey AP1509," 2010 [Online]. Available: http://search.digikey.com/scripts/DkSearch/dksus.dll?Detail&name=AP1509-12SGDIDKR-ND [Accessed: 2/6/2011].

[16] "OKR-T/3 Series," 2010 [Online]. Available: http://www.murata-ps.com/data/power/okr-t3-w12.pdf [Accessed: 2/6/2011]. [Accessed: 2/6/2011].

[17] "Digikey LM317," 2010 [Online]. Available: http://www.national.com/ds/LM/LM117.pdf [Accessed: 2/6/2011].

[18] "Only Batteries," 2010 [Online]. Available: http://www.onlybatteries.com/showitem.asp?ItemID=15994.128&cat1=27&uid=2071 [Accessed: 2/6/2011].

[19] "All Battery Li-Ion 18650," [Online]. Available: http://www.all-battery.com/li-ion18650148v4400mahpcbpreotectedrechargeablebatterywithbareleads.aspx [Accessed: 2/6/2011].

[20] "4200mAh Expandable NiMH Battery Pack" [Online]. Available: http://www.all-battery.com/sidebyside12v4200mahexpandablenimhbatterypackmodules.aspx [Accessed: 2/6/2011].

13.4 Patent Liability Analysis References

[1] T. Sakai, et al. "Autonomous moving apparatus having obstacle avoidance function." U.S. Patent 6515614. October 11, 2001. [Online]. Available: http://www.freepatentsonline.com/6515614.html

[2] Hashimoto, et al. "Robot system with vision sensor." U.S. Patent 7817847. October 5, 2005. [Online]. Available: http://www.freepatentsonline.com/7817847.html

[3] Hyoung-ki, et al. "Method and apparatus for reckoning position of moving robot." U.S. Patent Application 20070118248. October 17, 2006. [Online]. Available: http://www.freepatentsonline.com/y2007/0118248.html

[4] Google, Inc. (2011) Permission Guidelines for Google Maps and Google Earth. [Online]. Available: http://www.google.com/permissions/geoguidelines.html

13.5 Reliability and Safety Analysis References

[1] "MC9S12C128V1 Datasheet," [Online]. Available: http://cache.freescale.com/files/microcontrollers/doc/data_sheet/MC9S12C128V1.pdf?pspll=1 [Accessed: 4/6/2011].

[2] "Murata OKR-T Datasheet", [Online]. Available: http://www.murata-ps.com/data/power/okr-t3-w12.pdf [Accessed: 4/6/2011]

[3] “L298 Dual H-Bridge” [Online]. Available: http://www.st.com/stonline/books/pdf/docs/1773.pdf [Accessed 4/6/2011].

[4] “LM317 Data Sheet” [Online]. Avaiable: http://www.national.com/ds/LM/LM117.pdf [Accessed 4/6/2011].

[5] “BQ2002 Data Sheet”[Online]. Available: http://focus.ti.com/lit/ds/symlink/bq2002.pdf [Accessed 4/6/2011].

[6] MIL-HDBK-217F Military Handbook—Reliability Prediction of Electronic Equipment. [Online]. Available: https://engineering.purdue.edu/ece477/Homework/CommonRefs/Mil-Hdbk-217F.pdf. [Accessed Apr. 2011].

13.6 Ethical and Environmental Analysis References

[1] IEEE.org. “IEEE Code of Ethics. Internet: http://www.ieee.org/portal/cms_docs/about/CoE_poster.pdf [Accessed April 10, 2011].

[2] ACM.org. (1992, October 16). “Code of Ethics.” Internet: http://www.acm.org/about/code-of-ethics [Accessed April 10, 2011].

[3] “Walking.” Internet: http://en.wikipedia.org/wiki/Walking [Accessed April 10, 2011].

[4] International Network for Environmental Compliance and Enforcement. (1998, February 16). “Printed Circuit Board Manufacturing.” Internet: http://www.inece.org/mmcourse/chapt7.pdf [Accessed April 6, 2011]. – renamed to [11]

[5] “Aluminum Recycling”. Internet: http://en.wikipedia.org/wiki/Aluminium_recycling [Accessed April 14, 2011].

[6] All-Battery.com. “Side by Side 12V 4200mAh Expandable NiMH Battery Pack Modules with Standard and Mini Tamiya Connectors.” Internet: http://www.all-battery.com/sidebyside12v4200mahexpandablenimhbatterypackmodules.aspx [Accessed April 9, 2011].

[7] I. Buchmann. Cadex Electronics. (2001, April). “Recycling your Battery.” Internet: http://www.buchmann.ca/article16-page1.asp [Accessed April 9, 2011].

[8] "Electronic Waste." Internet: http://en.wikipedia.org/wiki/Electronic_waste [Accessed April 9, 2011].

[9] Multicircuits.com. "Technologies." Internet: http://www.multicircuits.com/pcb/tech/lead_free.html [Accessed April 29, 2011].

[10] J. Qiu. (2008, August 20). "Recycling: From computers to park benches." Internet: http://www.nature.com/nchina/2008/080820/full/nchina.2008.194.html [Accessed April 13, 2011].

13.7 Packaging Design Considerations References

[1] A. Cameron. (2010, June 23). Precision Guidance: U.S. Army Testing Rugged, Autonomous Robot Vehicle [Online]. Available: http://www.gpsworld.com/defense/precision-guidance/news/us-army-testing-rugged-autonomous-robot-vehicle-10101

[2] B. Boen and B. Dunbar. (2001, June 26). Urbie, the Urban Robot [Online]. Available: http://www.nasa.gov/vision/earth/technologies/urbie.html

13.8 Schematic Design Considerations References

[1] "MC9S12C128V1 Datasheet," 2010 [Online]. Available: http://cache.freescale.com/files/microcontrollers/doc/data_sheet/MC9S12C128V1.pdf?pspll=1 [Accessed: 2/6/2011].

[2] "Lynxmotion 4WD1," 2010 [Online]. Available: http://www.lynxmotion.com/p-603-aluminum-4wd1-rover-kit.aspx [Accessed: 2/6/2011].

[3] "20 Channel EM-406A SiRF III Receiver," 2010 [Online]. Available: http://www.sparkfun.com/products/465 [Accessed: 2/6/2011].

[4] "Triple Axis Accelerometer Breakout," 2010 [Online]. Available: http://www.sparkfun.com/products/252 [Accessed: 2/6/2011].

[5] "Triple Axis Magnetometer Breakout," 2010 [Online]. Available: http://www.sparkfun.com/products/9371 [Accessed: 2/6/2011].

[6] "Ultrasonic Range Finder - XL-Maxsonar EZ3," 2010 [Online]. Available: http://www.sparkfun.com/products/9494 [Accessed: 2/6/2011].

[7] "Infrared Proximity Sensor Long Range - Sharp GP2Y0A02YK0F," 2010 [Online]. Available: http://www.sparkfun.com/products/8958 [Accessed: 2/6/2011].

[8] "Digikey LM2675," 2010 [Online]. Available: http://search.digikey.com/scripts/DkSearch/dksus.dll?Detail&name=LM2675M-5.0-ND [Accessed: 2/6/2011]. [Accessed: 2/6/2011].

[9] "Digikey LM2675," 2010 [Online]. Available:http://search.digikey.com/scripts/DkSearch/dksus.dll?Detail&name=LM2675M-3.3-ND [Accessed: 2/6/2011].

[10] IBase Technology Inc., "Intel® Atom™ N270 3.5-inch Disk-Size SBC Intel® 945GSE Chipset – IB887", IBase Technology Inc. [Online] Available: http://www.ibase.com.tw/2009/Datasheet/IB887.pdf . [Accessed: Feb 16, 2011].

[11] "AT: 12V 4200 mAh NiMH Battery Pack"[Online].Available : http://www.all-battery.com/sidebyside12v4200mahexpandable nimhbatterypackmodules.aspx [Accessed: 2/17/2011]

13.9 PCB Layout Design Considerations References

[1] Bosch Sensortec. (2009, December 10). "BMA180 Digital, triaxial acceleration sensor Data Sheet." Internet: http://www.sparkfun.com/datasheets/Sensors/Accelerometer/BST-BMA180-DS000-03.pdf [Accessed, February 19, 2011].

[2] National Semiconductor. (2010, March 1). "LM117/LM317A/LM317 3-Terminal Adjustable Regulator." Internet: http://www.national.com/ds/LM/LM117.pdf [Accessed, February 17, 2011].

[3] Honeywell. (2009, February). "3-Axis Digital Compass IC HMC5843." Internet: http://www.sparkfun.com/datasheets/Sensors/Magneto/HMC5843.pdf [Accessed, February 19, 2011].

[4] Advanced Circuits. (2007). "Manufacturing and Technical Questions." Internet: http://www.4pcb.com/pcb-faqs-manufacturing-technical/ [Accessed, February 19, 2011].

[5] B. Suppanz. Advanced Circuits. (2007) "Trace Width Calculator." Advanced Circuits. Internet: http://www.4pcb.com/index.php?load=content&page_id=95. [Accessed, February 19, 2011].

[6] Lynxmotion. (2010). "Gear Head Motor - 7.2vdc 50:1 175rpm (6mm shaft)." Internet: http://www.lynxmotion.com/p-96-gear-head-motor-72vdc-501-175rpm-6mm-shaft.aspx [Accessed, February 19, 2011].

[7] M. Glenewinke. (1995). "AN1259." Freescale Semiconductor. Internet: http://cache.freescale.com/files/microcontrollers/doc/app_note/AN1259.pdf [Accessed, February 19, 2011].

[8] All-Battery.com. "Side by Side 12V 4200mAh Expandable NiMH Battery Pack Modules with Stadard and Mini Tamiya Connectors." Internet: http://www.all-battery.com/sidebyside12v4200mahexpandablenimhbatterypackmodules.aspx [Accessed, February 19, 2011].

[9] All-Battery.com. "Tenergy Smart Universal Charger for 7.2v – 12v NiMH/NiCd Battery Packs (Automatic Charging and Temperature Sensor)." Internet: http://www.all-battery.com/TenergySmartUniversalChargerforNiMHandNiCdBatteryPacks-01005.aspx [Accessed, February 19, 2011].

[10] Advanced Circuits. (2007). "PCB Tolerances." Internet: http://www.4pcb.com/pcb-design-specifications/ [Accessed, February 19, 2011].

[11] Murata Power Solutions. (2010, February 25). "OKR-T/3 Series." Internet: http://www.murata-ps.com/data/power/okr-t3-w12.pdf [Accessed, April 24, 2011].

13.10 Software Design Considerations References

[1] "Lucas–Kanade Optical Flow Method," 2010 [Online]. Available: http://en.wikipedia.org/wiki/Lucas%E2%80%93Kanade_Optical_Flow_Method [Accessed: 2/6/2011].

[2] "Freescale 9S12C32 Reference Documents" 2011 [Online]. Available: https://engineering.purdue.edu/ece362/Refs/9S12C_Refs/index.html [Accessed: 3/25/2011].

[3] "OpenCV" 2011 [Online]. Available: http://opencv.willowgarage.com/wiki/ [Accessed: 3/25/2011].

[4] "Google Earth" 2011 [Online]. Available: http://www.google.com/earth/index.html [Accessed: 3/25/2011].

[5] "C++ NMEA Sentence Parser" 2011 [Online]. Available: http://sourceforge.net/projects/cppnmeaparser/ [Accessed: 3/25/2011].

[6] "Download Dstar Lite 1.0 for Linux" 2011 [Online]. Available: http://linux.softpedia.com/get/Science-and-Engineering/Mathematics/Dstar-Lite-32304.shtml [Accessed: 3/25/2011]

[7] "D*" 2011 [Online]. Available: http://en.wikipedia.org/wiki/D* [Accessed 3/25/2011]

[8] "Dijkstra's algorithm" 2011 [Online]. Available:http://en.wikipedia.org/wiki/Dijkstra's_algorithm [Accessed: 3/25/2011]

Appendix A: Individual Contributions

A.1 Contributions of Dan Barrett:

Dan played a role in almost all sections of the project, but was primarily responsible for the software design and implementation. His contributions ranged from conceptual to hardware design, construction and implementation, to software design and implementation. He played a large role in the initial idea of the project, and the design of the initial hardware and software block diagrams based on analysis of the design constraints. He was responsible for the design of the battery charging circuitry, and played a role in the choice of parts, the layout of the printed circuit board, and the assembly and debugging of the printed circuit board. He was in charge of the software design and implementation. He also contributed to the final synthesis and debugging of the project as a whole.

Dan played a large role in coming up with the idea for the project, and deciding on the set of desired features. He created the preliminary versions of the hardware and software block diagrams. He was largely responsible for choosing parts which would be suitable for fulfilling the goals while meeting design constraints. These included the chassis and motor assembly, the battery, and the sensors.

Dan was responsible for the design of the battery charging circuit. He first had to research batteries, and decide on one which would fulfill the projects needs. After deciding on a Nickel-Metal-Hydride battery, he then set out to learn what must be done to safely and reliably charge one. He researched parts which are designed for that purpose, and found one which was suitable, the bq2002, which uses temperature and voltage to determine when the battery is charged. The bq2002 assumes an existing constant current source, which can be switched on and off with a voltage. Therefore, Dan designed and prototyped a voltage switchable current source, and then tested it in conjunction with the bq2002. After it proved successful, the design was incorporated into the schematic and PCB design.

Dan also helped with prototyping test circuits for the H-bridge, sonic and infrared range finders, accelerometer (which worked, but somehow broke, and was therefore not incorporated in the final design), compass, and logic level translators. He wrote a custom SPI and I2C implementations using general purpose I/O pins after it was determined that the built in SPI module was not sufficient.

Dan also helped with the design of the PCB, particularly in the layout phase, organizing the parts on the board in a way that would reduce clutter and keep sensitive parts separated from high voltages and noise. This included the initial part layout, and rearranging parts and traces in order to allow the power and ground traces to fit while being the desired width. He also researched serial ports and routed the serial port wires to ensure that our board is compatible with standard serial connectors.

Dan also helped with the population of the printed circuit board with parts. He helped with the placement and hardware debugging of the whole circuit piece by piece, and soldered many of the resistors and most of the battery charging circuit.

Dan was the one primarily responsible for the software design. After creating the initial system software block diagram, he set out to learn about the algorithms necessary to accomplish the goals of the project. The most involved were the Kalman filter, and the D* Lite path-finding algorithm. He was able to find implementations of both online. After learning the basics, he created a kinematic model of the robot as part of the Kalman filter implementation. He then created a simulation of the robot on which to test the software. He generated fake noisy measurements from the simulation in order to test the Kalman filter's ability to perform sensor fusion on multiple noisy measurements. He then implemented an initial wheel control system in order to make the simulated robot travel to a designated waypoint. Next, he added in simulated obstacle detection, and wrote the code necessary to add a detected obstacle to a graph data structure representing an obstacle map. After this, he incorporated the path-finding algorithm based on the obstacle map created with the simulated obstacle detection. Next, he found a GPS packet parser, and used it to extract latitude and longitude from the GPS module. He then created functions to convert coordinates between three coordinate systems: GPS latitude and longitude, pixel coordinates on the display, and local coordinates in meters. After making these functions, he took images from Google Earth, and was able to display real GPS data points on the map images in their real location. After getting this to work, he added the ability for the user to click on the screen, converting the mouse location to the real position on the map to which it corresponds. This position was added to the waypoint queue of the simulated robot, which would then travel to that location, finding a path around simulated obstacles. Dan then took time to clean up the code, and reorganize it to be object oriented, so that it would be easier to deal with when adding in communication with the microcontrollers. Next, he worked to create a packet protocol for use between the two microcontrollers and the Atom board, and then created the code necessary for the Atom board to simultaneously communicate with both microcontrollers using this protocol. After debugging the communication, he then began altering the code to send instructions to the microcontrollers, and to use the real sensor data from the packets in place of the fake simulated sensor data. Next, he worked with the rest of the team to fix a latency problem with the communication between the Atom board and microcontrollers. Once this was taken care of, he then combined the camera targeting code he and Sebastian had created in ECE362 with the rest of the code, and used it to make the robot follow a target. He then worked with Sebastian to create a new motor speed controller in order to improve the quality of turning. Next, he created functions to use real range-finder data to implement the obstacle mapping, and made the robot scan back and forth with the camera and rangefinder turret when not visually tracking a target. Finally, he incorporated the path-finding system with the motor speed controller in order to make the robot follow the path it finds to the destination added through the user interface.

A.2 Contributions of Sebastian Hening:

In the beginning of the semester Sebastian took part in the initial design of the project proposed the use of an Atom board for the image processing and path calculation. Sebastian also participated in creating the final proposal and researched parts for the robot with a focus on chaises, motors, servos, H-Bridges and batteries.

Sebastian also tested the image processing software on an Atom board to see if it is powerful enough. In addition he tested the viability of using remote desktop connection to stream the webcam video from to another computer over wifi.

Sebastian also chose the H-Bridge and helped make a prototype circuit to test it. He also helped with testing the power supplies and sonic range finder. After making sure that all the chosen components worked as intended, Sebastian also worked on designing the PCB, specifically the H-Bridge schematic, voltage regulators and microcontrollers. Sebastian also helped with checking and fixing errors in the layout of the PCB and making sure the decals fit the actual components. When the PCB was delivered, Sebastian soldered most of the components and made sure that each section of the PCB worked as intended.

In terms of software Sebastian used the functions made by Anthony to implement the code on the microcontroller interfacing with the camera servos, right wheel encoder and range finders. He also worked on the target tracking code on the atom and made it communicate with the Microcontroller over serial port. Sebastian also helped with getting the SPI implementation on the microcontroller to work and communicate with the accelerometer. Next Sebastian also wrote a PID control for the motors which takes as an input a desired speed and controls the signals to the motors in order to keep the robot moving at the specified speed. He tested and configured the PID control system on different surfaces to make sure that the turning is smooth enough so that the target being tracked by the webcam is not lost. Sebastian also spent many hours calibrating the control system based on the sensor data in order to have the robot travel to waypoints and avoid obstacles. Sebastian also worked with Dan on merging the target tracking code and his path finding code.

Packaging was another area Sebastian contributed in. He worked taking apart the Atom board and mounting it on the robot. He also worked on attaching the sensors, GPS, and PCB on the robot and connected all the components together. Sebastian's most significant contributions were made to the project concept and hardware design and assembly, however he tried to be involved in all aspects of the project.

A.3 Contributions of Sandunmalee Abeyratne:

Sandunmalee’s main contribution to the completion of this project was designing the PCB. She also assisted in the creation of the schematics as it was a process closely related with the PCB. Sandunmalee chose the GPS unit used by our robot and the thermistor used in the battery charging circuit. She also assisted in writing the code for reading the sonic range finder and in parsing the data from the compass.

Sandunmalee started early in the semester with the designing of the PCB as it is a crucial component of the project. As she had little prior knowledge of PCB design, she began by reading the tutorials that were posted on the course website. With the help of the tutorials, she learned about traces, pads, vias, and common measurements techniques such as “mils.” She followed all three step-by-step PADS tutorials on the course website to familiarize herself with the design process of a two-layer PCB. Once she was familiar with PADS Logic and PADS Layout, she made libraries for the decals that would be created for the components of the PCB. Then, one by one, she created decals for several components, such as the DIP socket for the microcontroller, pushbuttons, voltage regulators, h-bridge, and battery charging IC. In the process of creating these decals, Sandunmalee visited the datasheet for each part and made decisions on the sizes of pads, annular rings, and drill sizes. She custom created most of the decals and used the ready-made decals only for regular components such as headers, capacitors, and diodes. Once the decals were created, she assisted in connecting them to the schematic components used in PADS Logic. Because the PCB board for this project was quite large and had complex connections, she divided the schematic into several pages to facilitate easy editing and readability. During the process of laying out the components on the PCB board, she ensured that no traces had 90-degree or acute angles. She also ensured that all annular rings were sufficient in size and that traces did not run too close together. She decided to make the power supply traces 40mils in width to provide for ample current draw.

In addition to the PCB, Sandunmalee chose the GPS used in the robot. In order to find a suitable GPS for the project’s needs, she read online tutorials to understand the characteristics of different GPS units. After looking at several GPS units available in the market, she chose the 20-channel, EM-406A GPS receiver. She found this particular GPS to have much online support due to many projects using it. It operates at a convenient 5V and had a development board that could be bought separately. Sandunmalee also chose a thermistor for the battery charging circuitry. The battery charging IC – BQ2002/F – detects that the battery is fully charged when it detects a negative voltage change and the warm temperature of the battery via the thermistor. The BQ2002/F datasheet recommends a NTC thermistor, which is a resistor that rises in conductivity with increasing temperature. The datasheet did not specify the value of the thermistor, so she decided to buy several thermistors in the ranges of 10kΩ, 4.7kΩ, and 3.3kΩ, keeping in mind that they should be easily securable to the battery.

When Sandunmalee was not working with hardware, she assisted in some code writing, mainly for the sonic range finder and compass. She developed a routine that can parse the hex values output by the ATD conversion register to determine the distance of the object that the sonic rangefinder is detecting. She also assisted in parsing the values from the compass. The compass outputs a value that has both the strength and the angle of direction, from which the angle must be extracted using the arctangent function. She spent some time studying the arctangent function and comparing that to the values output by the compass to develop an expression that could correctly yield the direction which the robot is facing.

Towards the end of the semester, Sandunmalee brought her camera and started videotaping the functionality of the robot. She uploaded these video to the team's YouTube account. This video footage was also taken to create the PSSC demonstration video for the team. She also wrote the User Manual for the robot for homework 13.

A.4 Contributions of Anthony Myers:

Anthony's main role in the development of this project was to write code for the microcontrollers for sensor data acquisition and motor control algorithms. He was also in charge of obtaining the necessary materials for the packaging design, and assisted in debugging the issues the group was facing with the proper functioning of the microcontrollers when connected to the PCB. He also developed a rather simplistic solution for allowing the two microcontrollers to communicate with the Atom board without experiencing synchronization issues on the receiving end of the Atom board.

Early on in the semester, Anthony began writing the code for the microcontrollers. He studied various references on Embedded C, and started coding several simple test programs to get acquainted with the method of setting register values, setting the values of output pins, and reading the values on input pins. He then wrote Analog-to-Digital conversion routines for interpreting the voltages supplied by each of the range finders. The two infrared proximity sensors were a bit unique, as their voltage outputs weren't linear. He created a linear interpolation routine in order to translate the supplied voltages to distances as accurately as possible. The group was having several issues with getting the I^2C compass working, so he invested a large quantity of time into researching the different digital compasses the group had available to choose from. He ordered several different versions, and experimented with each. The best working compass was the HM55B compass from Hitachi, for which he created a custom bit-banging routine to match its own specified communication protocol. His largest endeavor was writing the motor control algorithms. The job of one of the microcontrollers was to receive motor direction and PWM commands for each of the motors from the Atom board, for which it was supposed to carry out the commands. In the motor control code, he implemented safety functions that would not allow the motors to change from rotating forwards to rotating backwards (or vise-versa) too quickly. He used several timers that worked in conjunction to ensure that proper care was taken with controlling the motor directions. In order to receive commands from the Atom board, he created an SCI interrupt routine that would place incoming data bytes starting with a specified start byte and ending with a specified stop byte into a receive buffer, allowing the microcontroller to parse the packets.

For the design of the packaging, it was soon realized that having a small enclosure that neatly covers and conceals all the electronics would be infeasible, as the project required a few serial to USB converters, which extended quite far beyond the limits of the chassis. The group also felt that it would be wise to allow the Atom board as much air as possible, to prevent it from overheating. In order to accommodate for these issues, Anthony came up with the idea for an "open" packaging system, which also allowed the group the ease of working on it and debugging, since the cables were easily accessible without completely dismantling the package each and every time something needed to be changed. He obtained the sheets of acrylic that were needed to assemble the packaging, and drilled the holes for assembling the initial version of the packaging design.

During the stages of testing, the group had fried four microcontrollers, and was having the most difficult time diagnosing the problem. After hours and hours of headaches from debugging, Anthony discovered that when shutting off the power to the robot, a 10V spike occurred, which slowly damaged the microcontrollers over time. After realizing that power was being shorted to ground when the switch was flipped to the off position, the switch was rewired and no more microcontrollers were fried.

Another major source of issues came from the fact that the two microcontrollers were sending data to the Atom board at different rates, forcing the Atom board to empty its input buffers each time before receiving a message. This was a very slow process, which resulted in huge delays in receiving "real time" messages from the microcontrollers. Anthony suggested that the microcontrollers should only send a packet of data to the Atom board when they received a packet of data. This method worked perfectly, and solved the synchronization issues with reading packets from each of the micros.

Near the end of the project, Anthony worked with Sandunmalee to finish most of the final documentation for the project, which included the final report and the senior design report. He also completed most of the video editing needed for compiling the final PSSC demonstration video.

Appendix B: Packaging

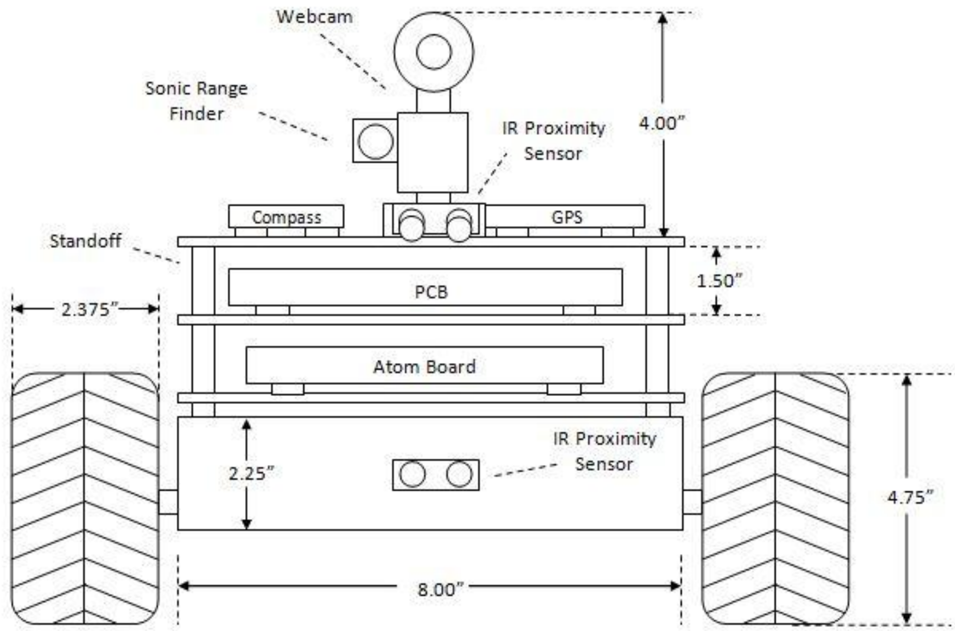

Figure B-1: Front View of Packaging

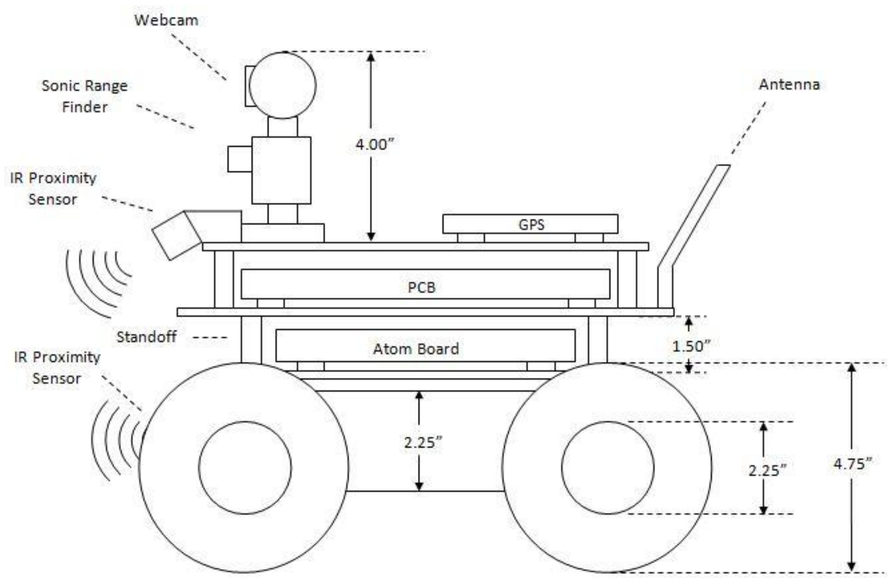

Figure B-2: Side View of Packaging

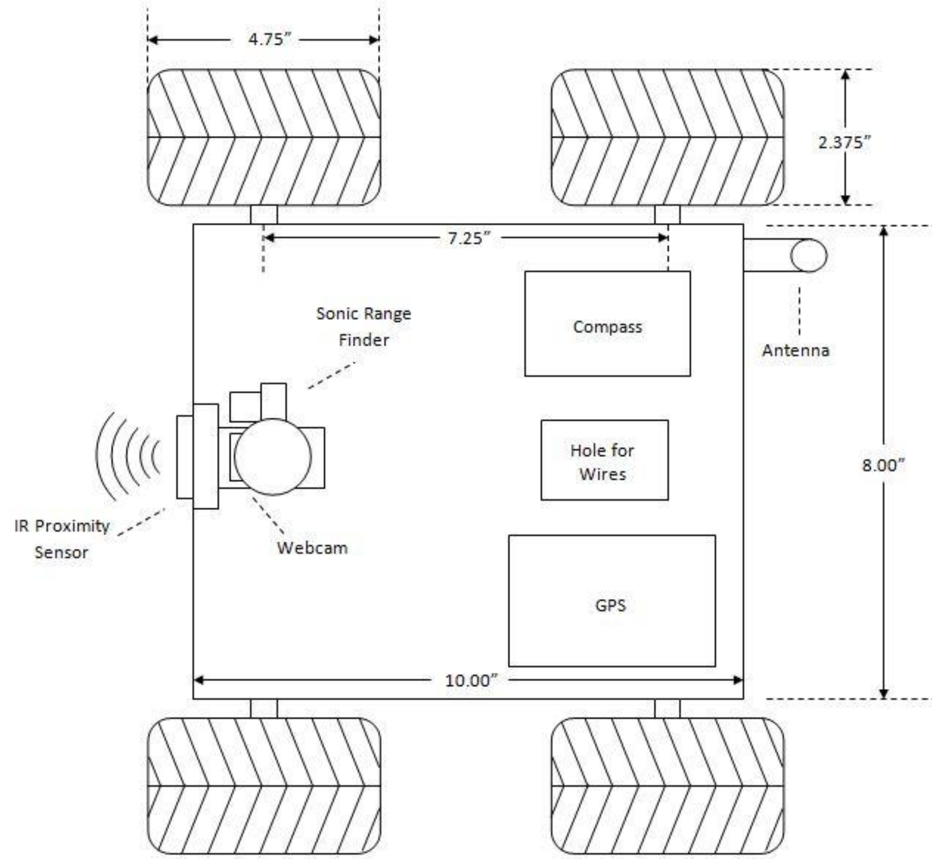

Figure B-3: Top View of Packaging

Appendix C: Schematic

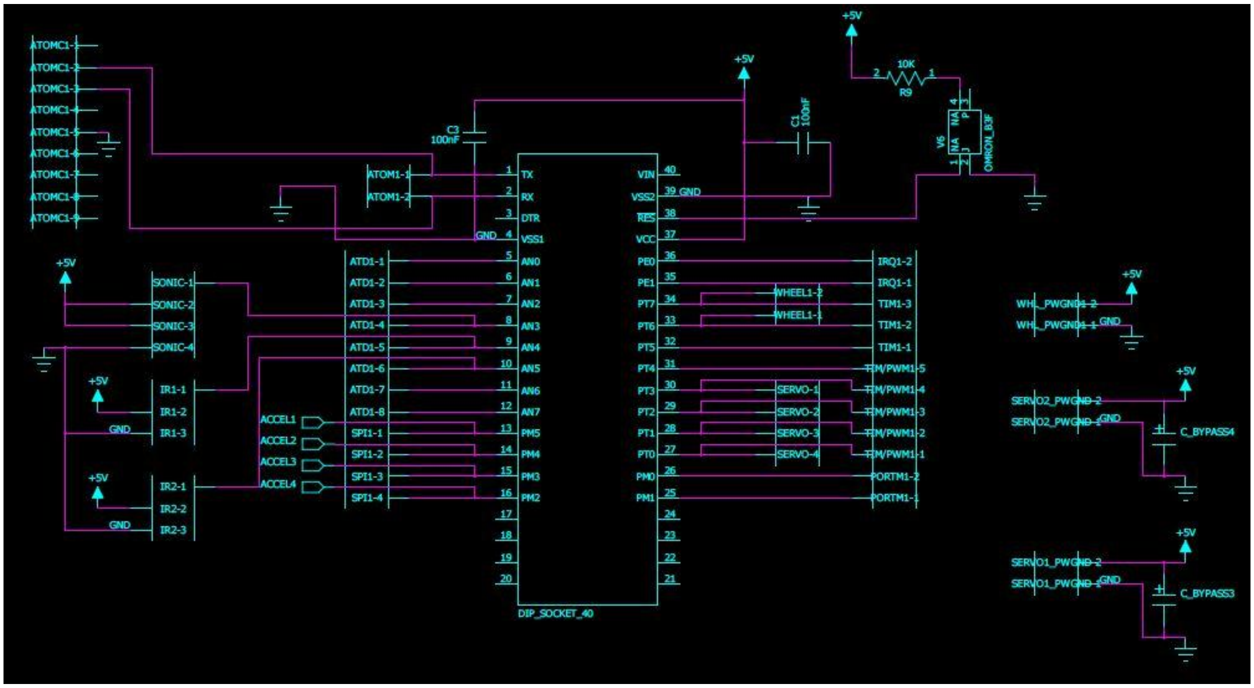

Figure C-1: Microcontroller #1 Circuit

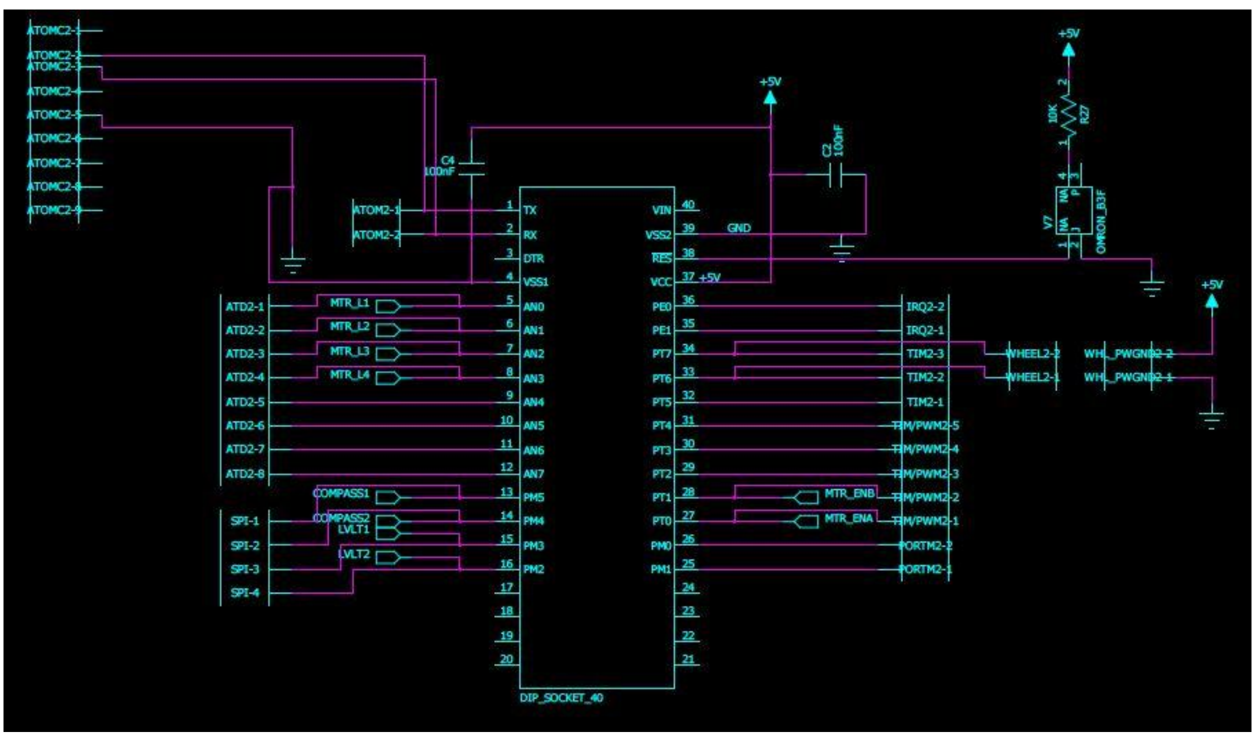

Figure C-2: Microcontroller #2 Circuit

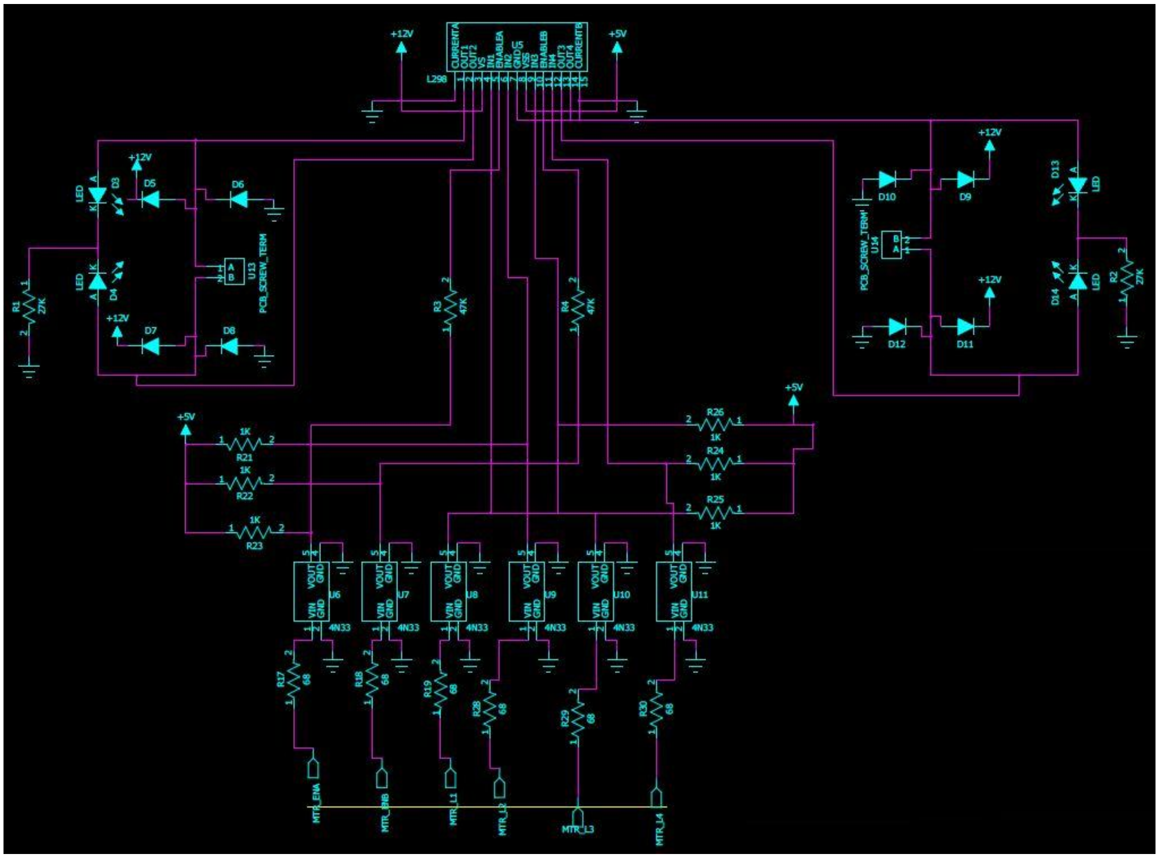

Figure C-3: H-Bridge Circuit

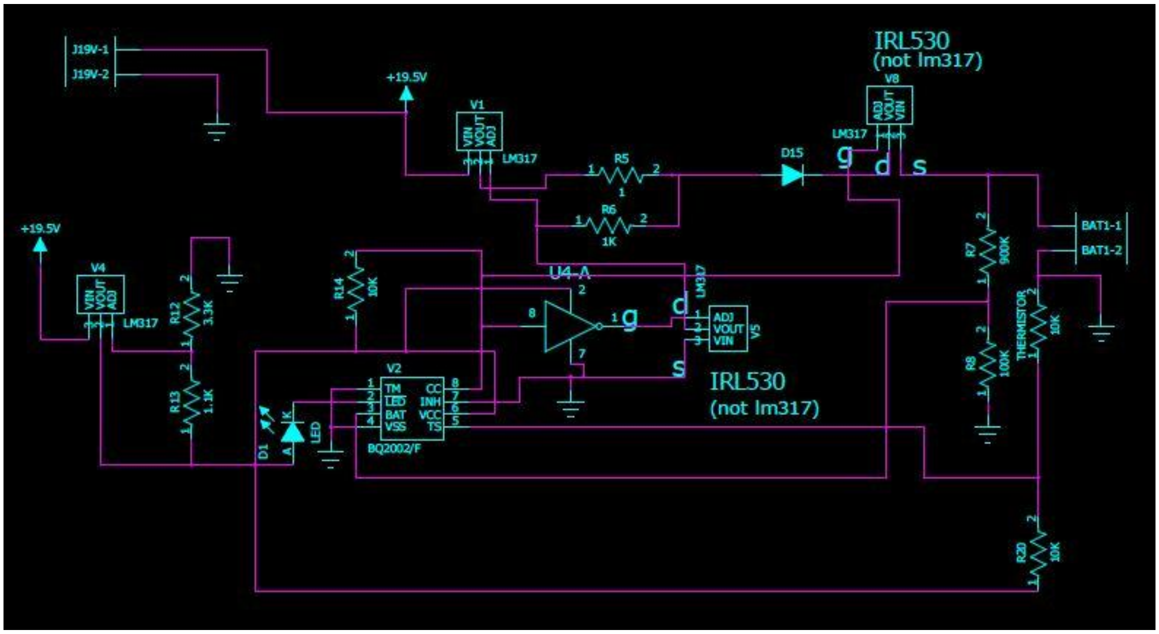

Figure C-4: Battery Charging Circuit

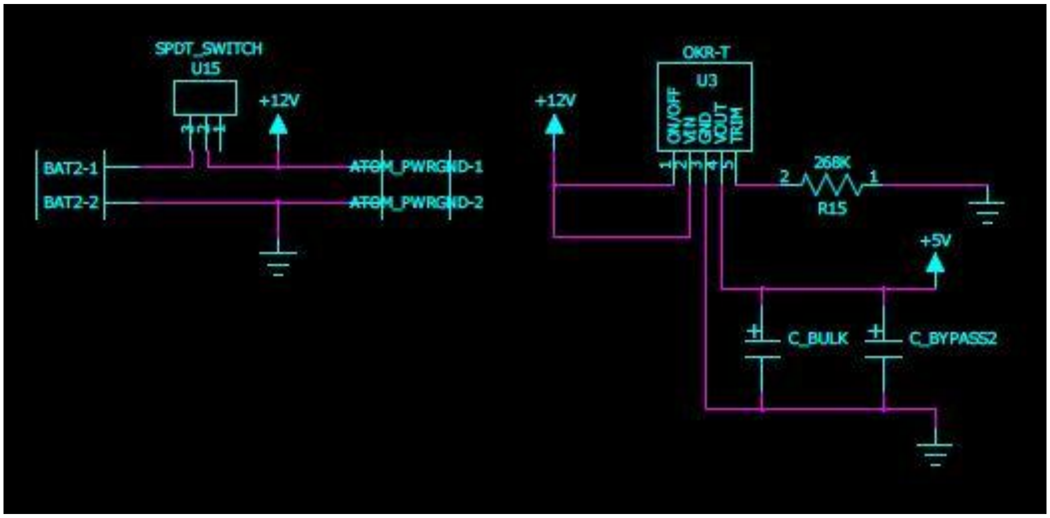

Figure C-5: Reset Switch Circuit

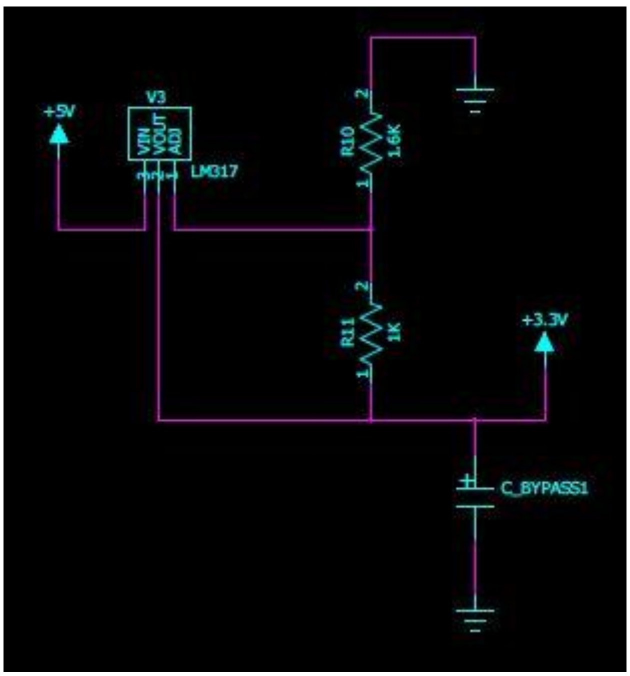

Figure C-6: 3.3V Regulator Circuit

Appendix D: PCB Layout Top and Bottom Copper

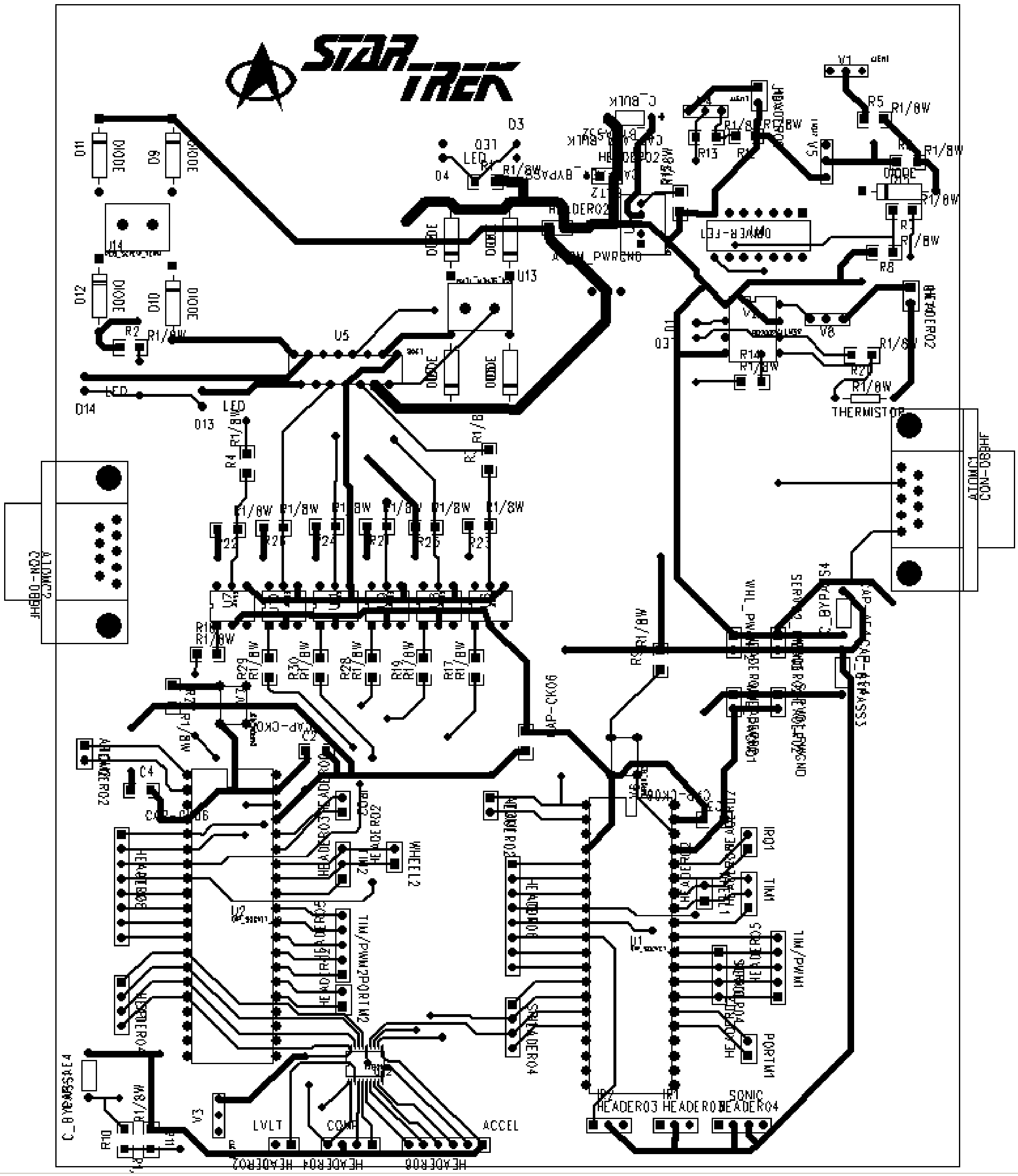

Figure D-1: PCB Top Copper (with Silkscreen)

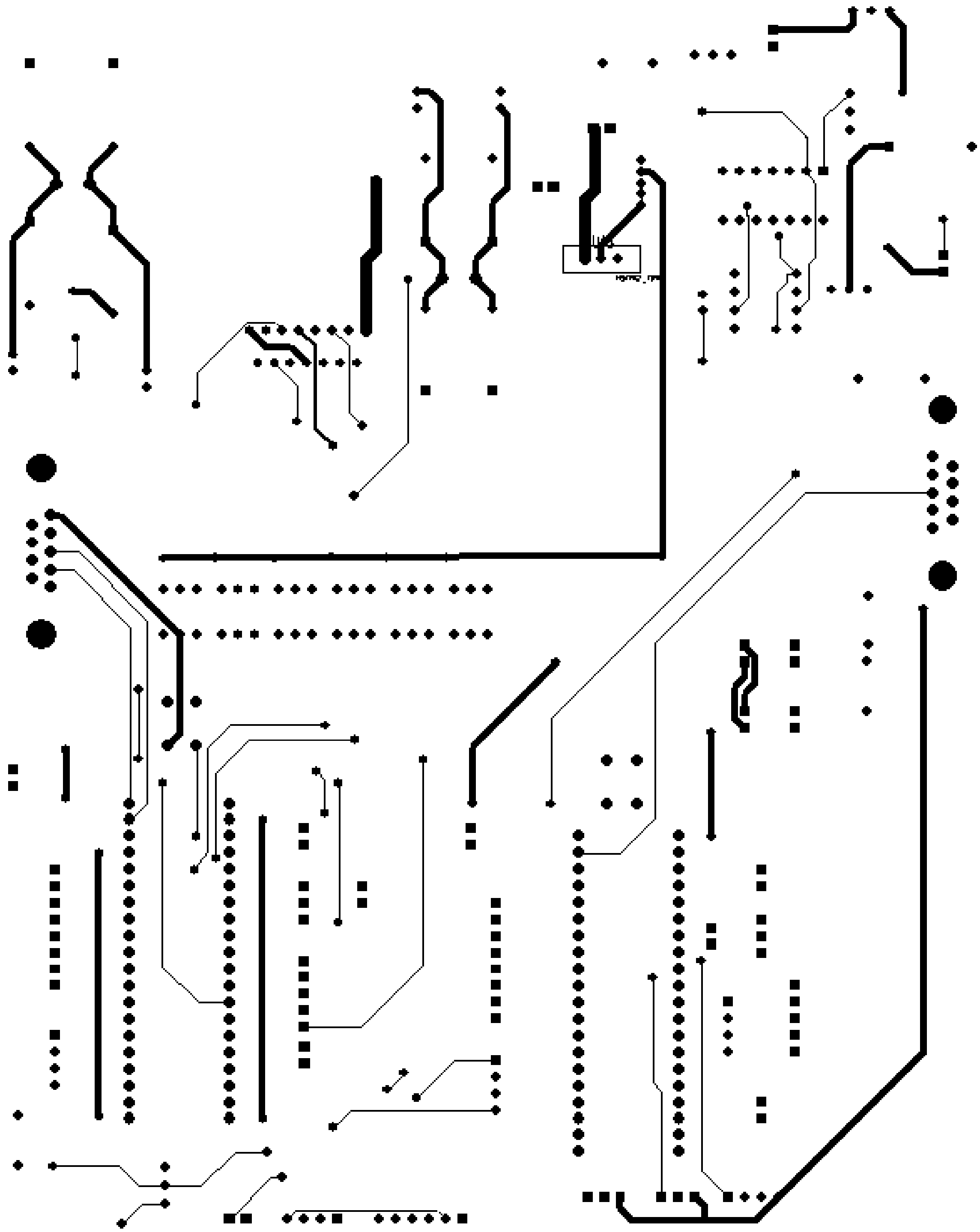

Figure D-2: PCB Bottom Copper (with Silkscreen)

Appendix E: Parts List Spreadsheet

Table E-1: Parts List Spreadsheet

Item	Unit Price ($)	Quantity	Total Price ($)
Chassis			
Robot Chassis	82.15	1	82.15
Gear Head Motor	21.95	4	87.80
Dual H-Bridge (L298)	19.99	1	19.99
Quadrature Motor Encoder	25.95	2	51.90
Off Road Robot Tires	25.00	2	50.00
Sensors			
GPS (20 Channel)	59.95	1	59.95
GPS Evaluation Board	39.95	1	39.95
Ultrasonic Range Finder	49.95	1	49.95
Infrared Sensors	14.95	2	29.90
Accelerometer	29.95	1	29.95
Magnetometer (HMC5843)	49.95	1	49.95
Magnetometer (Hitachi HM55B)	29.95	1	29.95
Magnetometer (Dinsmore 1490)	15.00	2	30.00
Webcam	0.00	1	0.00
Power Supplies			
Battery	56.12	1	56.12
Battery Charging IC (TI BQ2002)	2.64	4	10.56
Thermistor	1.50	5	7.50
5V DC/DC Converter (OKR-T/3)	6.39	1	6.39
Processors			
Atom Board	0.00	1	0.00
Atom Board Antenna	4.00	1	4.00
Freescale MC9S12C32 (replacements)	10.00	2	20.00
Freescale MC9S12C32	0.00	4	0.00
Miscellaneous Hardware			
Bulk Capacitor	0.00	1	0.00
Resistors	0.00	20	0.00
LEDs (blue, green)	0.00	10	0.00
40-Pin IC Sockets	1.50	2	3.00
PCB Headers	0.00	5	0.00
Diodes	0.00	9	0.00
Optical Isolators (4N33)	0.00	6	0.00
Cables			
USB (Type A to B)	0.00	1	0.00
Serial to USB Converters	0.00	2	0.00
PE Micro USB Multilink Interface	0.00	1	0.00
	Project Total		719.01

Appendix F: Component Memory Layout

Table F-1: Microcontroller 1 Memory Layout

Microcontroller 1

Name	Starting Address	Ending Address	Size
Registers	0x0000	0x03FF	1KB
RAM	**0x0800**	**0x0FFF**	**2KB**
Variables	0x0800	0x0900	256 Bytes
Sensor Data	0x0800	0x0810	16 bytes
unused	0x0810	0x0820	16 bytes
Serial Buffers	0x0820	0x0870	80 bytes
unused	0x0870	0x0880	16 bytes
Servo Control	0x0880	0x0890	16 byes
unused	0x0900	0x0D00	1KB
Stack	0x0D00	0x0FFF	767 bytes
FLASH	**0x4000**	**0x7FFF**	**16KB**
Code	0x4000	0x4FFF	4KB

Table F-2: Microcontroller 2 Memory Layout

Microcontroller 2

Name	Starting Address	Ending Address	Size
Registers	0x0000	0x03FF	1KB
RAM	0x0800	0x0FFF	2KB
Variables	0x0800	0x0900	256 Bytes
Sensor Data	0x0800	0x0810	16 bytes
unused	0x0810	0x0820	16 bytes
Serial Buffers	0x0820	0x0870	80 bytes
unused	0x0870	0x0880	16 bytes
Motor Control	0x0880	0x0890	16 byes
unused	0x0900	0x0D00	1KB
Stack	0x0D00	0x0FFF	767 bytes
FLASH	0x4000	0x7FFF	16KB
Code	0x4000	0x4FFF	4KB

Table F-3: External Interfaces

External interfaces			
Micro-controller 1		Micro-controller 2	
SCI		SCI	
	TX		TX
	RX		RX
Servo PWM		Wheel PWM	
	PWM0		PWM0
	PWM3		PWM2
SONIC Range.		Wheel direction control	
	AN0		AN0
IR Range 1			AN1
	AN1		AN2
IR Range 2			AN3
	AN2		
Accelerometer		Compass	
	PM0		PM0
	PM1		PM1
	PM2		PM2
	PM3		PM3
Pulse accumulator		Pulse accumulator	
	PT7		PT7
Wheel Direction Detection		Wheel Direction Detection	
	PT6		PT6

Table F-4: Microcontroller 1 Initializations

Register initializations

Microcontroller 1			
SCI			
	SCIBDH	0x00	9600 baud
	SCIBDL	0x9C	9600 baud
	SCICR1	0x00	
	SCICR2	0x2C	enable receiver/transmitter and enable interrupts
	DDRB	0x10	PB4 output mode
	PORTB	0x10	assert DTR
ATD			
	ATDCTL2	0x80	
	ATDCTL3	0x18	3 conversions
	ATDCTL4	0x05	
	ATDCTL5	0x00	single sequence, start at channel 0
PLL			
	SYNR	0x02	24mhz
watchdog	COPCTL	0x40	turn of watchdog
PWM			
	PWME	0x03F	
	PWMPOL	0xFF	
	PWMCLK	0x03	
	PWMPRCLK	0x00	
	PWMCAE	0x00	
	PWMCTL	0x30	concatenate ch0 and ch1 and also 2 and 3
	PWMSCLA	0x0A	
	PWMSCB	0x0A	
	MODRR	0xFF	
	PWMPER0	0b01011101	
	PWMPER1	0b11000000	
	PWMPER2	0b01011101	
	PWMPER3	0b11000000	
I2C (emulated)			
	DDRM	0b00111010	
	PTM	0b00111000	
Pulse accumulator			
	PACTL	0b10100010	enable pulse accumulator and set to rising edge and enable overflow interrupt

Table F-5: Microcontroller 2 Initializations

Register initializations			
Microcontroller 2			
SCI			
	SCIBDH	0x00	9600 baud
	SCIBDL	0x9C	9600 baud
	SCICR1	0x00	
	SCICR2	0x2C	enable receiver/transmitter and enable interrupts
	DDRB	0x10	PB4 output mode
	PORTB	0x10	assert DTR
PLL			
	SYNR	0x02	24mhz
watchdog	COPCTL	0x40	turn of watchdog
PWM			
	PWME	0x03	
	PWMPOL	0xFF	
	PWMCLK	0x03	
	PWMPRCLK	0x03	
	PWMCAE	0x00	
	PWMCTL	0x00	no concatenate
	PWMSCLA	0x3C	
	MODRR	0x03	use pt0 and pt1
	PWMPER0	0xFF	
	PWMDTY	0xFF	0% active low
	PWMPER1	0xFF	
	PWMDTY1	0xFF	0% active low
SPI (emulated)			
	DDRM	0b00111010	
	PTM	0b00111000	
Pulse accumulator			
	PACTL	0b10100010	enable pulse accumulator and set to rising edge and enable overflow interrupt
AN			
	DDRAD	0xFF	set pins to output

Appendix G: Software Flowcharts

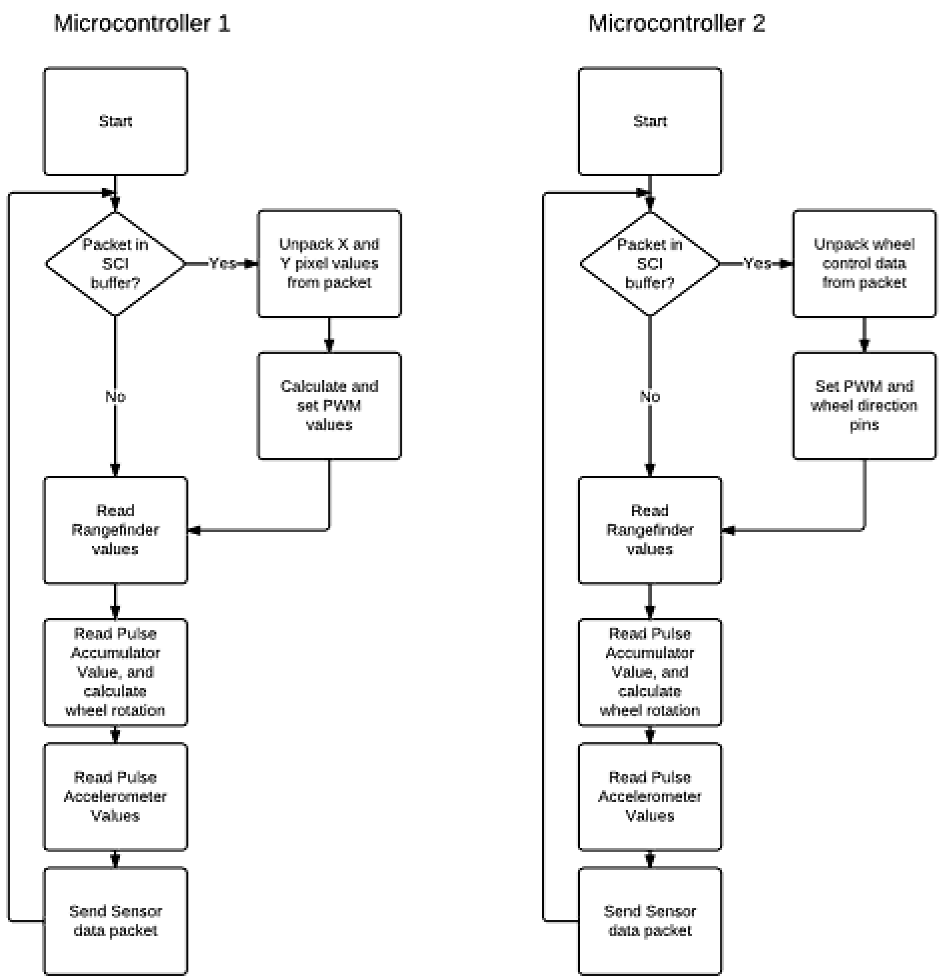

Figure G-1: Microcontroller Flow charts

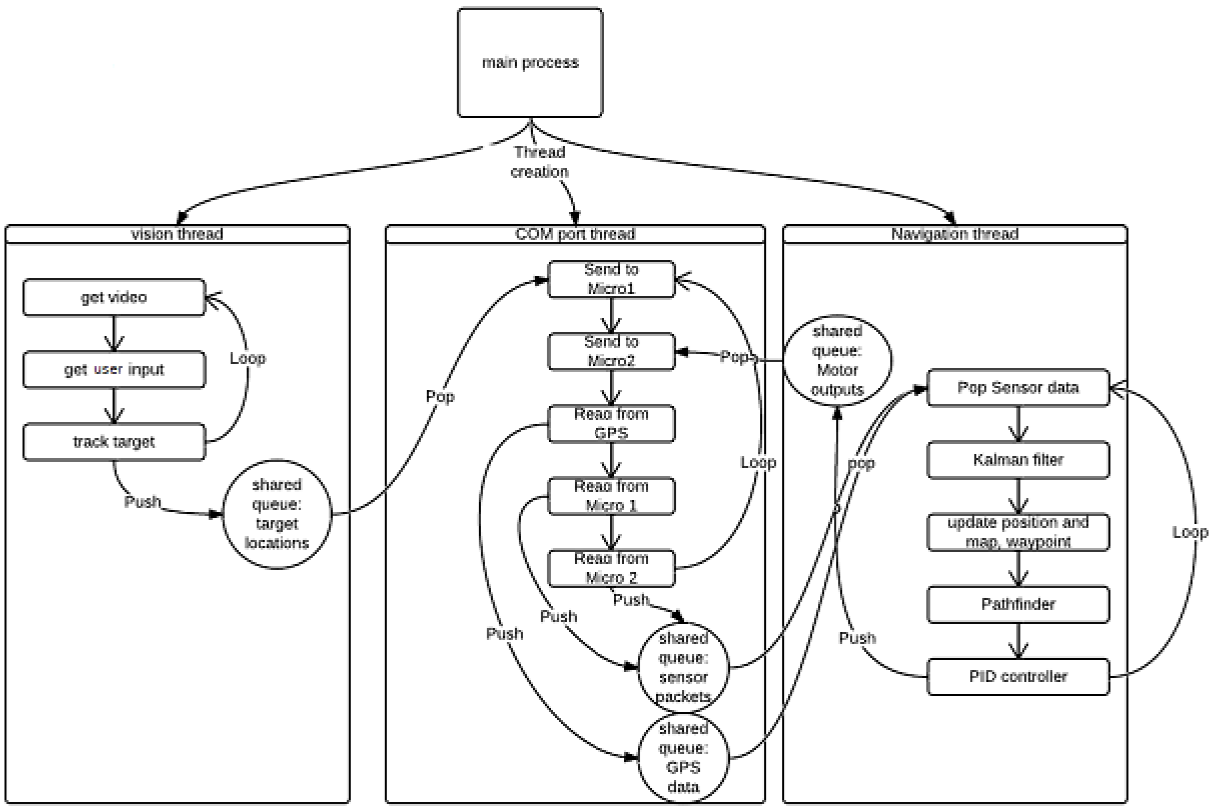

Figure G-2: Multi-Threaded Atom Board Diagram

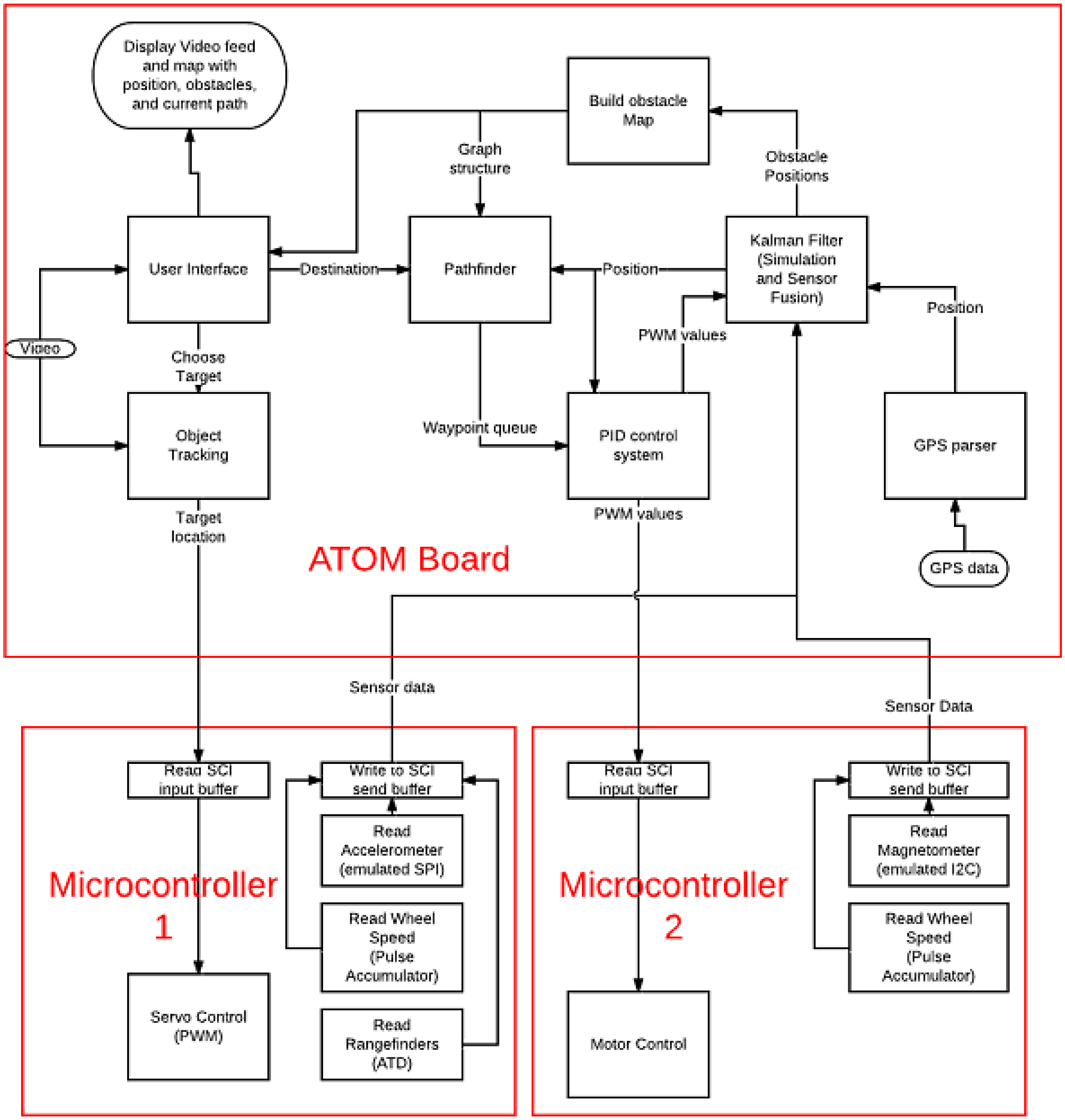

Figure G-3: Software Block Diagram

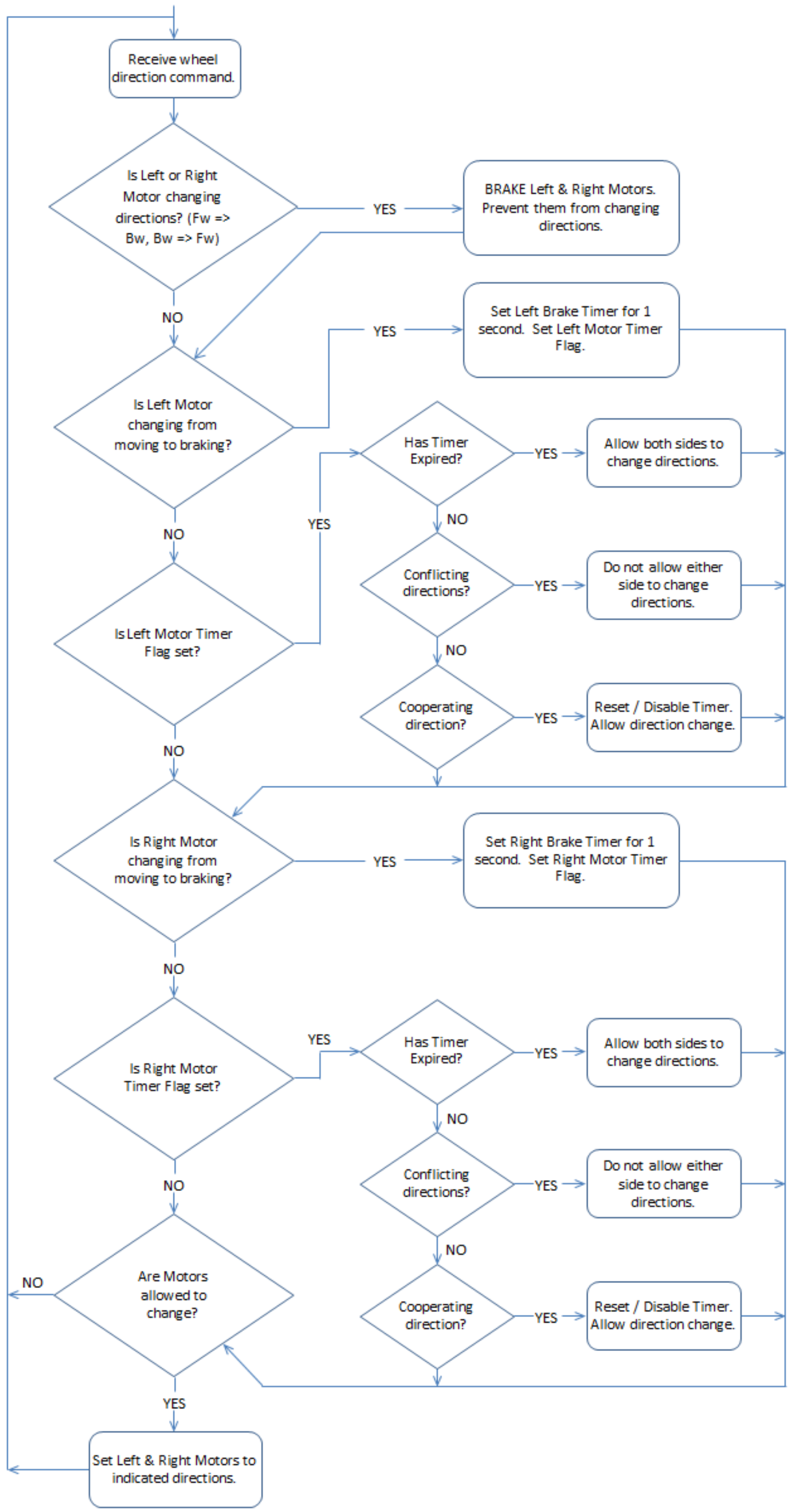

Figure G-4: Motor Control Algorithm

Appendix H: FMECA Worksheet

Table H-1: Microcontroller 1 Table

Failure No.	Failure Mode	Possible Causes	Failure Effects	Method of Detection	Criticality
A1	PWM signal failure	Failure of HC9S12C32 pins PT0-PT3.	The camera servos will not work.	Observation	Low
A2	SCI failure	Failure of the HC9S12C32 pins TX and RX.	Unable to communicate to atom board.	Observation	Low
A3	Failure of range sensors	Failure of the HC9S12C32 ATD pins AN3-AN5	Unable to detect obstacles. Can result in injury of others.	Observation	High
A4	Failure to Reset or run	Failure of R9 or Pushbutton	The microcontroller is unable to reset or is being constantly reset	Observation	Medium
A5	Failure of SPI	Failure of HC9S12C32 SPI pins PM2-PM5	Unable to read the accelerometer data. Will make the calculation of the current location less accurate	Observation	Low
A6	Failure of Timer	Failure of HC9S12C32 timer pins PT6-PT7	Unable to determine the speed and direction of the robot	Observation	Low

Table H-2: Microcontroller 1 Table

Failure No.	Failure Mode	Possible Causes	Failure Effects	Method of Detection	Criticality
B1	Failure of Motor logic signals	Failure of the HC9S12C32 AN0-AN3 pins	Unable to control the movement of the robot.	Observation	High
B2	SCI failure	Failure of the HC9S12C32 pins TX and RX.	Unable to communicate to atom board.	Observation	Low

B3	SPI Failure	Failure of the HC9S12C32 pins PM2-PM5 or failure of Compass	Unable to determine the direction of movement of the robot	Observation	Low
B4	Failure to Reset or run	Failure of R9 or Pushbutton	The microcontroller is unable to reset or is being constantly reset	Observation	Low
B5	Failure of PWM signal	Failure of the HC9S12C32 pins PT0-PT1	The motors are not able to stop	Observation	High

Table H-3: 3.3V Linear Regulator

Failure No.	Failure Mode	Possible Causes	Failure Effects	Method of Detection	Criticality
C1	Vout = 0	Failure of the LM317 or no Vin	Unable to power the accelerometer and compass	Observation	Low
C2	Vout = 1.25 V	short circuit R11 failure or open circuit R10 failure	Unable to operate the accelerometer and compass	Observation	Low
C3	Vout >3.3V	Open circuit R11 failure or short circuit R10 failure	Compass and accelerometer would be damaged	Observation	Medium

Table H-4: Murata 5V Regulator

Failure No.	Failure Mode	Possible Causes	Failure Effects	Method of Detection	Criticality
D1	Vout < 5V	R15 fails and causes an open circuit	The servos, sensors and microcontrollers do not function	Observation	Low
D2	Vout >5V	R15 fails and shorts	The sensors and microcontroller might be damaged	Observation	Medium

D3	Vout = 0	Failure of the Murata OKR-T creating a short	The batter might heat up and explode or traces might be destroyed.	Observation	High

Table H-5: H-Bridge

Failure No.	Failure Mode	Possible Causes	Failure Effects	Method of Detection	Criticality
E1	Unable to control motors	One of the Resistors R21-R26 fails or one of the six 4N33 optical isolators fails	Motor behave randomly and cannot be controlled	Observation	High
E2	H-bridge failure	Failure of the L298 H-bridge	Unpredictable. Robot might drive into others and cause injury.	Observation	High

Table H-6: Battery Charging Circuit

Failure No.	Failure Mode	Possible Causes	Failure Effects	Method of Detection	Criticality
F1	Current source failure	LM-317 failure	The battery does not charge	Observation	Low
F2	Failure of charging IC	Failure of BQ2002/F chip	Failure to detect when battery is charged which can lead to destruction of battery. Batter can also explode.	Observation	High
F3	Failure of Q1	Failure of the Q1 transistor	Unpredictable. Unable to start charging or stop charging the battery. Battery can explode or ignite.	Observation	High
F4	Failure of Q2	Transistor Q2 fails	This is in place only in case other components fail. Its failure might in the worst case make the	Observation	Low

			battery unable to be charged.		
F5	Current source > 1 A	Resistors R5 or R6 fail.	LM317 current source breaks	Observation	Medium
F6	Current source <1A	Resistor R5 or R6 failure	The battery doesn't fully charge	Observation	Low

Table H-7: Voltage Level Translator

Failure No.	Failure Mode	Possible Causes	Failure Effects	Method of Detection	Criticality
G1	No data transaction to and from compass/accelerometer	Failure of TXB0108	Unable to read accelerometer and compass data on the microcontroller. Compass and accelerometer might get destroyed.	Observation	Medium