ECE 477 Final Report – Spring 2008
Team 12 – The Two Wheel Deal

Team Members:

#1: Pete Dudash Signature: ____________________ Date: _________

#2: Greg Eakins Signature: ____________________ Date: _________

#3: Eric Geier Signature: ____________________ Date: _________

#4: Jeremy Gries Signature: ____________________ Date: _________

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	
		TOTAL	

Comments:

TABLE OF CONTENTS

Abstract

The Two Wheel Deal is a compact, two wheeled, self-balancing personal transportation device. It will serve as an economical and practical alternative to most forms of short range transportation, including cars, bikes, and walking. Having only two wheels gives the Two Wheel Deal a small footprint for enhanced mobility over conventional means of transportation. Additional features such as high speed motors, emission free electric drive, and a zero turning radius make the Two Wheel Deal like walking, only better.

1.0 Project Overview and Block Diagram

1.1 Design/Functionality Overview:

The idea for this project stemmed strongly from our mutual interest in digital controls theory. The Two Wheel Deal elegantly embodies the classical and difficult controls problem of the inverted pendulum in a fun and practical package. The Two Wheel Deal is designed to make life easier for the time-stressed individual by automating the strenuous and time consuming act of walking.

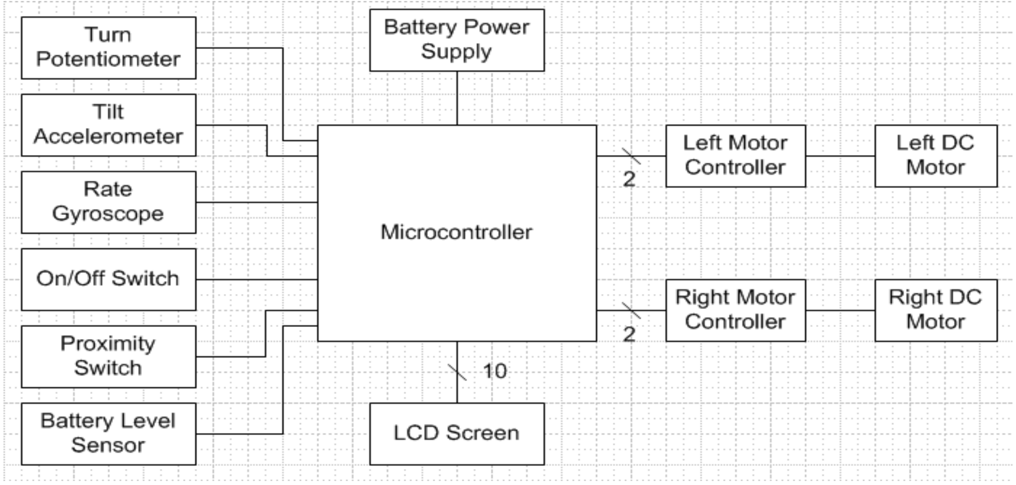
The Two Wheel Deal will be a personal transportation vehicle. The primary goal of the vehicle's electronics is to balance the vehicle. As the rider leans forward the motors turn in proportion to the angle of lean to try and keep the wheels under the center of gravity. This in turn will move the rider forward if the lean is sustained. The same process will work when the rider leans backward enabling the vehicle to go in reverse. Turning will be accomplished using a knob on the handlebars. Turns will be made by differing the speeds of the individual wheel motors. It will also be able to balance itself when a rider is not on it as well as shut down if the rider falls off. Finally it will have an LCD screen for a human interface to display important information.

Table 1.1: Cost Estimate

Component/Part	Quantity	Price	Cost
NPC-T74 DC Motors	2	$324.00	$648.00
SLA Batteries	2	$50.00	$50.00
Motor Controllers	2	$120.00	$240.00
Angular Rate Sensor	1	$100.00	$100.00
Dual Axis Accelerometer	1	$70.00	$70.00
Aluminum and Fasteners	1	$150.00	$150.00
Wheels	2	$50.00	$100.00
LCD Screen	1	$50.00	$50.00
Proximity Sensor	2	$30.00	$60.00
			Grand Total
			$1468.00

Pete with his expertise in controls theory/system modeling will help determine the controls algorithm for the controller software as well as well as determining the overall packaging design. Greg with his know-how in electromechanical control will be designing the motor controllers and interfacing to the motors. Eric with his skills in software/interfacing will design the controller model and software program as well as interface to LCD. Jeremy with his knowledge of controls theory and PCB layout will assist in determining the controls algorithm as well as designing the PCB layout.

Figure 1.2: Block Diagram



Division of Labor:

Design Component Homework		***Professional Component Homework***	
4-Packaging Design and Specs	Dudash	3-Design Constraint Analysis/Parts List	Eakins
5-Hardware Narrative and Prelim Schematic	Eakins	10-Patent Liability Analysis	Gries
6-PCB Narrative and Prelim Layout	Gries	11-Reliability and Safety Analysis	Dudash
9-Software Design Narrative	Geier	12-Social/Political/Environmental Analysis	Geier

2.0 Team Success Criteria and Fulfillment

Project-Specific Success Criteria:

1. An ability to independently control two high current electric motors.
2. An ability to shut down if no rider or low battery.
3. An ability to display sensor data to the rider on an LCD display.
4. An ability to balance a passenger autonomously.
5. An ability to move and turn through use of the navigation controls.

All of the Project-Specific Success Criteria (PSSC) were fulfilled, with some being more difficult than others. PSSC-1 was the first item to be completed. Once the PCB's were made and populated, one motor was connected and worked on the first try. This PSSC was the basis for PSSC four and five, so it was a vital step in creating the Two Wheel Deal.

PSSC-2 was fairly straightforward as it was completed by creating a voltage attenuator signal that was sent to the microcontroller. This was by far the easiest PSSC of the five, with the toughest part being the calculation of current that the microcontroller could handle.

PSSC-3 was a little bit difficult in that many timing delays needed to be met in order to display meaningful data to the LCD. This PSSC took a little time to figure out, but the datasheet was very helpful in fulfilling this PSSC.

PSSC-4 was the most difficult of the five PSSC. It took much patience and a couple of weeks to get the vehicle to a point where the control algorithm could be written and tuned. With no prior experience tuning a PD controller to an unknown system, the tuning process took a couple of weeks to get right. Eventually, the best way to tune the controller was to set the derivative gain to zero while finding a proportional gain that put the system on the edge of stability (the critical gain). Then, the derivative gain was increased until the system exhibited an acceptable transient response. This tuning method that was invented for this project is amazingly similar to the tuning method proposed by Ziegler and Nichols back in 1942 for tuning PID and PD controllers. Completing this PSSC meant that a human being could ride the Two Wheel Deal, and it was the main focus of our project.

PSSC-5 allows for the Two Wheel Deal to turn. Originally the steering gain was the same at all speeds. Next, the steering gain was set lower at higher speeds. This compensation makes turning easier and more intuitive.

3.0 Constraint Analysis and Component Selection

3.1 Introduction

The Two Wheel Deal is a compact, maneuverable, self-balancing personal transportation vehicle meant to revolutionize the way that short distance travel is conducted. Most of its design constraints are derived from the fact that it will be used as a mode of transportation to replace walking. It must be small enough to travel most places that pedestrians travel, it must be powerful enough to move a normal sized person while having enough extra power to maintain balance at normal speeds, and it must be light enough to pick up if necessary. The design of this vehicle is based heavily on the Segway PT, a commercial product built by Segway LLC. As such, many of the design constraints are based on the attributes of the production model.

3.2 Design Constraint Analysis

The most essential group of design considerations in this project involves the control system that is used to keep the vehicle upright. This means choosing sufficiently powerful motors, sensors with an appropriate measurement range, and a microcontroller capable of crunching through a precise control algorithm while interfacing to all of the vehicle's devices. The success of this project hinges on choosing all of these components appropriately.

The human interface and safety system is important for ensuring a safe and comfortable experience on the vehicle. An LCD display for displaying vital information, an easy to use steering mechanism, and battery level monitoring are among a few of the important components that must be considered in this design. Appropriate selection of these devices should allow for easy control of the vehicle and a reasonable degree of safety.

3.3 Computation Requirements

The control algorithm will be the primary responsibility of the microcontroller. This will involve reading the outputs of the accelerometers, angular rate sensor, and the steering angle sensor, then computing the required torque for each wheel and updating the PWM duty cycle. The target update rate is 100 Hz, which would be equivalent to that of the Segway [1]. The algorithm needs to use floating point arithmetic, which will require a decent amount of processing power.

The microprocessor will need to do some filtering on the sensor inputs, as the pulse width modulated motors and the potentially rough terrain will cause some noisy input signals.

The last computational responsibility of the microcontroller will be the output to the LCD screen. Every character on the screen will have to be regularly refreshed to provide up to date data to the user.

3.4 Interface Requirements

The microcontroller will first need to interface to the motor controllers using a PWM signal. This interface is optically isolated to prevent current spikes and noise from destroying the microcontroller or PLD. The microcontroller will also interface directly to the accelerometers and the angular rate sensor using ATD pins.

The main need for general I/O pins comes from the LCD. The controller chip in the LCD requires 10 pins to control. If the availability of pins becomes short, the LCD may also be controlled via the SPI by using an external shift register to free up some pins. Additional I/O might be needed at a later time to facilitate the addition of various status LEDs, vehicle lighting (headlights, tail lights, and turn signals), safety features, and other accessories.

3.5 On-Chip Peripheral Requirements

The current flowing through the motors is controlled using a pulse width modulated signal. Having precision control over this signal aids immensely in creating a stable and robust control algorithm; therefore this design requires two 16-bit PWM channels for motor control. This is the primary limitation on the microcontroller used in this application.

All of the motion sensors have analog outputs, necessitating multiple on-chip analog to digital converters. The accelerometers will need 2 ATD channels, the angular rate sensor will need 2 ATD channels, the steering mechanism will need 1 ATD channel, and the battery level monitoring circuit will need 1 ATD channel. A total of 6 ATD channels will be needed, with 10-bit resolution being preferred for precision.

3.6 Off-Chip Peripheral Requirements

The only off-chip peripheral used in this design is the controller chip that is built into the LCD. Instructions are clocked into it through a parallel interface, and then it will take care of interfacing directly to the display.

3.7 Power Constraints

The Two Wheel Deal’s power supply needs are centered about the needs of the two large motors. The power needs of all other components in this design are extremely small compared to the needs of the motors. The battery needs to supply current up to 400 amps to the motors for very short periods of time at an operating voltage of 24 volts[3]. Typical current draw is estimated to be about 50 amps, with far less being drawn during steady state conditions. The batteries also need to be rechargeable and be relatively compact, as the small nature of this vehicle limits the amount of space available for batteries. Battery life of the vehicle is highly dependent on the weight of the rider and the driver's riding style, but it is estimated to be around 3 hours on a fully charged battery. When the battery gets low, the vehicle does not shut down. Instead, it continues to work, but it is not able to balance as robustly.

3.8 Packaging Constraints

The majority of the packaging constraints stem from this vehicle's required ability to traverse areas that pedestrians would typically occupy. This means that it needs to be narrow enough to fit through a door (less than 30" wide), have enough ground clearance to pass over imperfections in the terrain without rubbing off expensive electrical components, and be strong enough to carry a normal sized passenger. The packaging section of this document covers the packaging in more detail.

3.9 Cost Constraints

The primary constraint on cost is that The Two Wheel Deal must be built such that it is competitive in price to the real Segway and not too hard on the pocketbooks of the team. Doing this is a trivial task, as a new Segway comes with a price tag of at least $4000. The target build cost of The Two Way Deal is approximately $1000, less than half the price of a new Segway. The price of the prototype version will be minimized through the aggressive pursuit of donations and product samples.

3.10 Component Selection Rationale

3.10.1 Microcontroller

The needs for the microcontroller outlined in sections 2.1, 2.2, and 2.3 are all satisfied by the two microcontrollers shown in Table 3.0.1 in Appendix C. The PIC is comparable to the Atmel in almost every attribute[4][5]. Both microcontrollers have all of the needed I/O, sufficient flash memory, and both come in a DIP package for easy development. Price is not a large factor for this part, as the microcontroller is only a very small portion of the total cost. The Atmel ATmega32 was chosen due to its high quality and readily available development tools. The AVR series is well documented and has a huge amount of support online [6]. The fact that it uses the highly optimized GCC compiler to assemble its code is another bonus, as this will help in reducing the memory used and improve performance.

3.10.2 Accelerometer

This design needs to have a dual axis high sensitivity accelerometer to detect the angle of tilt on the vehicle. A couple of accelerometers were considered, and their features are compared in Table 3.0.2 of Appendix C. The Analog Devices accelerometer was chosen for this project primarily due to its 5V operating voltage, high sensitivity, and simple pin out[7]. The additional cost of the Analog Devices chip is minor when the additional circuitry required for the Freescale chip is taken into consideration. By choosing only 5V devices for this design, the amount of overhead circuitry is minimized and the overall design simplified.

3.10.3 Angular Rate Sensor

To determine how fast the vehicle is tipping over, a sensor that detects the rate of change in angular position is vital. Theoretically, this value could be obtained in software by differentiating the reading from the accelerometers, but in reality signals have noise, and noise will quickly destroy a signal resulting from a differentiation, and any filtering would delay the response time of the system, therefore, a dedicated angular rate sensor is the best option. The selection of the angular rate sensor depended primarily on the needs of the control algorithm. An angular rate sensor of medium sensitivity is preferred so that the output of the sensor does not easily saturate during quick changes in position, yet is sensitive enough to detect a decent range of movement speeds. The gyroscopic angular rate sensors in Table 3.0.3 of Appendix C were considered. The Melexis part was chosen for this project because it provides the best compromise between cost and sensitivity of the three[8], and the Analog Devices part was not available at the time of the decision.

3.10.4 Motors

The motors were selected based on the physical needs of the system. The Two Wheel Deal should have a top speed of at least 15 mph, and have enough torque at that speed to recover from a tilt from vertical of 15°. By using some simple physics calculations, it was determined that high torque, low RPM motors were needed. Two motors made by National Power Chair were determined to be appropriate for this application. The abilities of the two motors are compared in Table 3.0.4 in Appendix C. While both motors satisfied the requirements of this project, the NPC-T74 met them by a greater margin[3]. The donation of the NPC-T74 motors solidified the decision.

3.10.5 Batteries

All of the requirements for a power supply are met by either using two small 12V lead acid batteries in series, or a string of 20 1.2V NiMH D cell batteries. The two options are compared in Table 3.0.5 in Appendix C. The sealed lead acid batteries were chosen primarily due to the huge difference in cost. The D-cell NiMH batteries would have given the vehicle tremendous battery capacity in a small volume, but they come at a high price. The lead acid batteries selected are commonly used in wheelchairs, and the selected motors are also from wheelchairs, showing that these batteries are indeed capable of driving the motors.

3.10.6 LCD

Only a few constraints guided the selection of the LCD screen. It had to use the Hitachi HD44780 LCD driver due to previous experience with this display, it needed to be large enough and customizable enough to display a few pieces of information to the rider, and it needed to be backlit for nighttime and indoor visibility. A large number of LCD screens fit these criteria. Two of them are compared in Table 3.0.6 in Appendix C. The deciding factor came down to cost, and the LCD20x4BL from Futurlec came out on top.

3.10.7 Packaging

The limiting factor to the width of the vehicle is the depth of the motors, so the motors will be mounted as close together as possible so that the width of the vehicle is much smaller than a standard doorway. Ground clearance is achieved through the selection of 16” high quality plastic bike wheels made by Skyway Wheels. These wheels will give a ground clearance of 8” from the middle of the motor shaft, leaving room for the width of the motors and other large components hanging off the bottom. Replacement tires for these 16” wheels are easy to find at any store that sells children’s bikes, which greatly improves the serviceability of the vehicle. The target passenger load is 200 lbs, and will be limited only by the strength of the chassis.

3.11 Summary

Through the careful selection of all components required for this project, all of the major design constraints have been met. The microcontroller’s high resolution outputs to the motor controllers combined with the appropriately chosen accelerometer and angular rate sensors should enable a high precision control loop for balancing the vehicle. The powerful motors combined with the capable and small batteries will ensure that the vehicle has enough muscle to move the vehicle in any way that is required to maintain balance or move a person. These design decisions should ensure that The Two Wheel Deal becomes a reality.

4.0 Patent Liability Analysis

4.1 Introduction

The “Two Wheel Deal” is a personal transportation vehicle which balances itself automatically by rotating the wheels forward or backwards to keep its center of gravity. This is accomplished using angle and angular rate sensors. The angle sensor provides the relative angle to the horizontal ground. The rate sensor provides a means to tell if the angle is approaching the horizontal towards instability or approaching the vertical towards the stable axis. The “Two Wheel Deal” also consists of two independent motor controllers which provides a means for turning the transportation device.

With the device powered, the user will step onto the riding platform. If the user leans forward, the vehicle will accelerate forward to compensate for the change of the angle from the vertical. The similar intuition applies for the reverse motion. The user will also be able to monitor various data such as battery life, speed, and angles.

The “Two Wheel Deal” is susceptible to patent infringement because it mimics exactly the same functions as the popular and well-known modern day SEGWAY. The SEGWAY is a unique device in that only one company manufactures them, and there is not much competition in the marketplace.

4.2 Results of Patent and Product Search

The first patent that was found was Patent No.: US 6561294 “balancing Vehicle with Passive Pivotable Support.” This patent was filed on August 31, 1999, and it refers to the SEGWAY [9]. The summary of the device describes it as a balancing vehicle that supports a rider in such a manner as to allow the center of gravity of the vehicle to be varied by motion of the support. The creator of the device claims that the vehicle will: 1) Transport a human subject over a surface, 2) Provide local stability while the vehicle is powered up 3) Transport a human using motorized drives 4) Vary its position based on leaning from the rider.

All of the primary functions contained in the SEGWAY are mimicked in the “2 Wheel Deal.” One function the SEGWAY uses is the microcontroller as a means for processing information. Also, it uses five accelerometers and two gyroscopes for redundancy checking to determine the angle of the vehicle to rebalance itself. The SEGWAY has two independent motor controllers used for providing a pulsed-width modulated signal to the motors for control. Turning is accomplished by shifting the handlebars to the right or left.

The “Two Wheel Deal” was based off of the SEGWAY, so infringement is unavoidable. The “Two Wheel Deal” would have to be redesigned from the ground up for any chance at marketing this product. All four of the key claims are being performed substantially the same way.

The second patent that was similar was Patent No.: 2003-159546 “balancing Vehicle with Passive Pivotable Support.” This patent was filed on June 1, 2004, and it refers to a SONY “skate-way” device [10]. This invention is a two-wheeled vehicle that travels front, back, left, and right by a rider moving balance on a step-board. These key claims are somewhat similar to the previous patent. The inventor claims that this device can: 1) Be steered by movement of balance 2) Detect balance of rider by pressure sensors.

The primary functions of this device include detecting a user’s input by the means of analog pressure sensors. This input will travel to the controller which is responsible for interpreting the data and sending out a pulse-width modulated signal to the motor drives. Finally these motor drives control the two independent motors.

One of the functions that the “Two Wheel Deal” perform substantially the same way is that the motors are driven by motor drives. These drives are controlled using a pulsed electrical signal which is the same at the “Two Wheel Deal.” Also, primary function of the invention is a human transportation device, which serves the same purpose as the “Two Wheel Deal.”

The third patent that was similar was Patent No.: 5755452 “Electric Scooter” This patent was filed on January 31, 1996 and it refers to a scooter that has an electric motor attached [11]. This device is similar to the “Two Wheel Deal” because it is a balancing transportation vehicle with two wheels driven by an electric motor. This device has the frame of a scooter with an electric motor attached to the back wheel so that the front wheel may be steered by the user. Also four electric batteries are mounted underneath the frame so that they may be concealed. The key claim of this invention is that it is an electric scooter for powered movement over a ground surface. The primary function of this device is for the electric motor turn the back wheel so that the user may compensate for steering and balance. The function that the “Two wheel deal performs the same way is electric motor propulsion.

4.3 Analysis of Patent Liability

The “Two Wheel Deal” literally infringes upon the SEGWAY patent in every aspect. All of the functions of the SEGWAY were mimicked, and this was accomplished by researching how the SEGWAY worked. The “Two Wheel Deal” uses an accelerometer and gyroscope to detect rider balance, and the only difference is that the SEGWAY uses redundant detection. Also both vehicles use a digital microcontroller to control the motor drives which control the independent motors. The only function not performed substantially the same way would be the turning mechanism. The “Two Wheel Deal” uses an analog joystick to detect a change while the SEGWAY uses the vehicle’s shaft to determine a change.

With respect to the “skate-way” patent, I do not believe that there is any literal infringement. While the “skate-way” is a personal transportation device that requires skill to use, the “Two Wheel Deal” uses a completely different balance detection mechanism. The “skate-way” would require the user to determine the balance of the machine and shift his or her weight to keep the device balanced. This would not fall under the doctrine of equivalents. The only function performed substantially the same way would be a controller sending pulsed electrical signals to the motor drives controlling the motors. There is no difference in this respect between the two devices. The potential for patent liability with the electric scooter should not be a problem, because there is no literal infringement. Also, infringement by the doctrine of equivalents wouldn’t apply here because the “Two Wheel Deal” uses a completely different way of balancing the vehicle as well as driving the motor.

4.4 Action Recommended

To avoid potential infringement with the SEGWAY, it would have to be re-designed from the ground up. It would be more reasonable, more time, and cost effective to buy a license for this patent or pay the inventor royalties. With respect to the “skate-way”, the only similar function would be how the controller drives the motors. Since the use of this application is wide-spread, I don’t see any infringement, literal or by the doctrine of equivalents, so I believe nothing would have to be modified on the “Two Wheel Deal.” I do not believe that any action would have to be taken because there is no infringement on the electric scooter patent.

4.5 Summary

There are three patents which resemble the “Two Wheel Deal.” These include the SEGWAY, SONY’s “skate-way”, and the electric scooter. The “Two Wheel Deal” infringes upon every aspect of the SEGWAY, and a license would have to be purchased to sell this project. There is no literal infringement with the “skate-way”, and the doctrine of equivalents would have to be stretched to apply to the “Two Wheel Deal.” Finally the electric scooter doesn’t have a patent claim that would infringe upon the “Two Wheel Deal.” Altogether, this project would not be able to be marketed and sold “as is.”

5.0 Reliability and Safety Analysis

5.1 Introduction

The Two Wheel Deal is a device used to balance and transport a single rider on two wheels. The design uses an accelerometer and gyroscope to sense when the center of gravity is not directly over the axis of the wheels. It then drives the motors in a precise way to balance the rider safely and smoothly. The rider leans forward to move this device forward, and similarly for reverse. Obviously, the main safety concern for this project is protecting the wellbeing of the rider and the system by avoiding injury and damage from falling over. The Two Wheel Deal will only be able to protect the user to a certain degree. It is still the rider's responsibility to be a safe and smart driver. A safety feature being designed utilizes a dead-man's switch to change The Two Wheel Deal between balancing by itself and balancing with a rider. With this implementation, if the user falls off during a ride, the device will theoretically come to a stop and balance itself. Also, if the rider is brave enough to drive near full speed or full tilt, the LCD screen will alert the rider that he/she is driving at a dangerous rate and the device may no longer be able to balance correctly. In order to best identify possible failures of high and low criticality and keep the user safe, the schematics for the motor controller and brain board have been broken into functional blocks and analyzed individually.

To be a successful and commercial product, the Two Wheel Deal also needs to be reliable. A few of the most critical components that will be analyzed in this report include the two LM340T Linear Voltage Regulators [12], the HRF3205 MOSFETs [13], ATmega32 microcontroller [14], ADXL203 dual-axis accelerometer [15], and the MLX90609E2 gyroscope [16]. The failure rate and mean time to failure (MTTF) are included in the tables below for each of the previously mentioned components.

5.2 Reliability Analysis

The components mentioned above have a high risk associated with them if they fail. They have been analyzed for reliability and the results can be seen in the tables below. The voltage regulators and power MOSFETs are the main components in the Two Wheel Deal that are most likely to fail as a result of over-heating, causing the entire transport device to stop working. The microcontroller is also highly critical because it is what calculates the necessary signal to send to the motors to balance the rider. If this component fails, then the entire system cannot operate. The accelerometer and gyroscope are responsible for measuring the current tilt angle and its derivative. Without these components functioning correctly, the balancing algorithm won't be able to produce the correct signal to stabilize the system.

The next couple sections are comprised of the necessary calculations to determine the number of failures per 10^6 hours and mean time to failure (MTTF) for each of the previously mentioned components. To compute the number of failures/10^6 hours, the components use the following model from page 25 of the Military Handbook—Reliability Prediction of Electronic Equipment [17]: $\lambda_P = (C_1\pi_T + C_2\pi_E)\pi_Q\pi_L$ and $MTTF = (\lambda_P)^{-1}$. In these equations, λ_P is the number of failures per 10^6 hours, C_1 is the die complexity, π_T is the junction temperature coefficient, C_2 is the package failure rate, π_E is the environmental constant, π_Q is the quality factor, and π_L is the learning factor associated with how long the particular component has been manufactured. Assumptions have been made in order to complete the following analysis. First of all, the Two Wheel Deal's components are operating at their respective maximum temperatures (π_T). Also, the system will be ground mobile (π_E) with commercially manufactured parts (π_Q) in production for over two years (π_L). The tables below are used to derive the failure rates and MTTF using information taken from the component datasheets.

Table 5.2.1: LM340T Linear Voltage Regulators

Parameter	**Value**	**Assumptions**
C_1	0.02	Linear, between 101-300 transistors
π_T	58	Linear, max temp of 125° C
C_2	0.0012	3 pins, Nonhermetic DIP
π_E	4.0	Ground Mobile on wheeled vehicle (G_M)
π_Q	10.0	Commercially manufactured component
π_L	1.0	More than 2 years in production
λ_P	11.648	Failures/10^6 hours
MTTF	**85,851.6484 hours = 9.794 years**	

Table 5.2.2.: HRF3205 Power MOSFETs

Parameter	Value	Assumptions
C_1	0.01	Linear, between 1-100 transistors
π_T	480	Linear, max temp of 175° C
C_2	0.0012	3 pins, Nonhermetic DIP
π_E	4.0	Ground Mobile on wheeled vehicle (G_M)
π_Q	10.0	Commercially manufactured component
π_L	1.0	More than 2 years in production
λ_P	48.048	Failures/10^6 hours
MTTF	**20,812.5208 hours = 2.374 years**	

Table 5.2.3: ATmega32 Micro

Parameter	Value	Assumptions
C_1	0.14	8 bit, CMOS
π_T	58	Linear, max temp of 125° C
C_2	0.019	40 pins, Nonhermetic
π_E	4.0	Ground Mobile on wheeled vehicle (G_M)
π_Q	10.0	Commercially manufactured component
π_L	1.0	More than 2 years in production
λ_P	81.96	Failures/10^6 hours
MTTF	**12,201.074 hours = 1.393 years****	

**According to the microcontroller datasheet [14], "Reliability Qualification results show that the projected data retention failure rate is much less than 1 PPM over 20 years at 85°C or 100 years at 25°C."

Table 5.2.4: Accelerometer

Parameter	Value	Assumptions
C_1	0.01	Linear, MOS device, between 1-100 transistors
π_T	58	Linear, max temp of 125° C
C_2	0.0016	4 functional pins, Nonhermetic
π_E	4.0	Ground Mobile on wheeled vehicle (G_M)
π_Q	10.0	Commercially manufactured component
π_L	1.0	More than 2 years in production
λ_P	5.864	Failures/10^6 hours
MTTF	**170,532.06 hours = 19.467 years**	

Table 5.2.5: Gyroscope		
Parameter	**Value**	**Assumptions**
C_1	0.01	Linear, MOS device, between 1-100 transistors
π_T	21	Linear, max temp of 110° C
C_2	0.0043	10 pins, Nonhermetic
π_E	4.0	Ground Mobile on wheeled vehicle (G_M)
π_Q	10.0	Commercially manufactured component
π_L	1.0	More than 2 years in production
λ_P	2.272	Failures/10^6 hours
MTTF	**440,140.845 hours = 50.244 years**	

These results and calculations are derived for the worst case scenario for each component. To get these results, π_T is assumed to be at the maximum operating temperature before the component burns up and no longer works. That is why the MTTF is relatively low. If we operate the voltage regulators at a more realistic temperature of 65° C ($\pi_T = 2.0$), the MTTF would increase to nearly 255 years. If temperature is an issue in the future, heat sinks can be added to help dissipate that energy and increasing the lifetime and reliability of that component. That is exactly what the fan on each motor controller board does for the MOSFETs. It continuously circulates cool air directly onto the components. If these parts operated at a more realistic temperature of 35° C ($\pi_T = 0.23$), the MTTF would increase to nearly 1,608 years. If the microcontroller also operates near 35° C, then the MTTF improves to about 400 years. By using values for π_T that are within a more reasonable temperature range that our system will be operating in, the MTTF for these critical components drastically increases. Based on the operation at lower temperatures, the reliability of each of the critical components is acceptable.

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

The schematics for the Two Wheel Deal's brain board and motor controller boards have been broken down into functional blocks. This illustration can be seen in Appendix A. These blocks include Section A for the brain board's signal outputs to the motor controller boards, Section B for the microcontroller and related components, Section C for the 5V and 12V Linear Voltage Regulators, Section D for the brain board's ADC inputs from the accelerometer, gyroscope, battery detection level, and joystick for steering (not shown), Section E for the motor controller’s MOSFET H-Bridge, and Section F for the H-Bridge driver module. All the possible failure conditions for each functional block and the possible causes are included in Appendix B.

To determine the reliability of the entire system, two different degrees of criticality have been defined. Any failure that concerns the system’s main balancing and maneuvering functions is defined as being a *high* criticality with $\lambda_P < 10^{-9}$. These failures could result in system instability, damage to the device, or injury to the rider. On the other hand, failures that affect the auxiliary aspects of the system, but still allow device functionality, are defined as being *low* criticality with $\lambda_P \geq 10^{-4}$. For example, the LCD screen malfunctions or displays incorrect information, the dead-man’s switch or alarm buzzer doesn’t work, or other similar supplementary functions. For failures of low criticality, there is no definite threat of injury to the user or damage to the device. It just serves as an inconvenience to the user and may cause customer dissatisfaction.

The completed FMECA charts are included in Appendix B. If the user falls off the Two Wheel Deal, it is assumed that it is dangerous and harmful. Also, it’s assumed that the device has been damaged or broken due to the crash. That is why this situation is defined as high criticality.

5.4 Summary

After completing the analysis, observation is crucial in identifying the failures and the resulting effects. The components that were analyzed are very important because failures in those particular functional blocks have high criticality. This report also provides a good idea about how long the components will live. The lifetime of the components can be increased by proper heat sinking, therefore significantly lowering the temperature coefficient (π_T). With regards to the FMEC analysis, the most critical failures occur in Sections B & E which contain the microcontroller circuitry and the MOSFET H-Bridge circuit, respectively. Precautions have been taken to protect these sections from failures in the design, but accidents do happen. The MOSFETs have a fan to minimize heat and the microcontroller is safe from over-voltage and noise by two linear voltage regulators. Also, the brain board and two motor controller boards are safely secured under the base plate to prevent damage from collisions with objects on the ground and crashes. Overall, the Two Wheel Deal is a safe and reliable product.

6.0 Ethical and Environmental Impact Analysis

6.1 **Introduction**

The Two Wheel Deal is a self-balancing, personal transportation device that is battery powered and driven using two high-speed motors. It works on the control theory principle of the inverted pendulum [18]. This way to accelerate or decelerate the rider simply needs to lean forward or backward respectively. It can be navigated using a joystick near the handlebars. The Two Wheel Deal is meant to serve as a practical and economical alternative to many forms of short distance transportation. Since it has a footprint no larger than a standard wheelchair it can be used practically anywhere.

As with any new or old product, there are ethical and environmental concerns that must be considered before releasing the product to the public. The first and foremost ethical concern regarding the Two Wheel Deal is the safety of the rider. Since this is a transportation vehicle that can move at speeds up to 10 mph, there are many ways that the rider could seriously injury himself/herself. Decisions to be considered are the placement of warning labels, extensive testing in a variety of environments, user training and documentation, and safety mechanisms. There are environmental concerns to consider as well. In the production stage of the product the PCBs and hardware components produce environmental pollution during their creation. During the normal use stage of the product, the batteries present a potential environmental hazard, and during the disposal stage basically all of the components have special disposal or recycling needs.

6.2 Ethical Impact Analysis

As previously mentioned there are a variety of ethical concerns that need to be considered before releasing the Two Wheel Deal to the public. A helpful guideline when considering engineering ethics is to refer to the Institute of Electrical and Electronics Engineers (IEEE) Code of Ethics [19]. The first major ethical concern is to ensure the Two Wheel Deal has been tested in a variety of environments. Since this is a transportation vehicle a variety of terrains must be considered such as grass, pavement, concrete, gravel, and dirt just to name a few. This problem would be addressed by trying the Two Wheel Deal on these different terrains and seeing how the vehicle reacted. The results and recommendations on how to traverse these environments would be gathered and placed in the user manual to allow the rider easy access to this data. It is crucial to state the vehicle cannot be driven in rain or snow since the chassis is not weatherproof. The vehicle would also be tested in a variety of temperatures. Data would be collected to inform the customer of possible premature component malfunction at varying operating temperatures, and this would be displayed in the user manual. One last consideration is to determine how great of a hill or incline the Two Wheel Deal can affectively travel over. This would be included in the user manual and possibly on a warning label on the vehicle.

This brings up a second ethical concern. This concern includes where warning labels should be placed as well as what dangers need warning labels. One warning label would be placed on the bottom of the machine near the batteries. It would warn the user of the danger that could occur if the batteries are shorted together. It would provide a picture of an electrical hazard as well as a reference to a page in the user manual that the rider could find more information on the danger. There would also be a warning label placed near the charging circuit for the batteries. It would display the potential fire hazard if the batteries were charged incorrectly. It would also have a reference to the user manual for more details on the danger. Two other warning labels would be placed on the outside edge of the frame above the small fans used to keep the motor controller PCBs cool. It would depict a moving fan warning to prevent anyone from accidentally sticking their fingers near the fans if they reach under the vehicle. A warning label would be placed on the floor of the vehicle depicting only one rider at a time on the Two Wheel Deal as well as a weight limit. This is important since the motor shafts have a maximum weight limit that can be applied, and since the warning label is on the floor of the vehicle a rider would see it each time he/she gets on it. There would also be a warning label placed on the LCD box that warned the user of the maximum angle the vehicle should be tilted to as well as the greatest incline the vehicle can safely maneuver. By placing this near the display the rider can always check the current tilt angle shown on the LCD with the maximum. Information on the tilt angle as well as incline would be included in the user manual, which is another ethical concern.

User documentation is vital because it gives the engineers and the developing company a direct way to provide the user with crucial information to protect his/her safety. The most important part of the documentation would be the user manual. This would include specific details on how to drive the vehicle, how to care for the vehicle, and how to troubleshoot problems that may arise. The manual would include information on the various environments described earlier, how the vehicle will react differently, and instructions on special actions that need to be taken during those environments. The manual would also include a list of warnings that the rider needs to be aware of when on or near the machine. It would explain in more detail some of the warning labels of the vehicle. Finally the user manual would include contact information to the Two Wheel Deal retailers as well as maintenance and disposal/recycling centers, which will be explained later.

Also included with the user manual would be a training DVD to help new users or old users wishing to refresh skills. This would provide an opportunity to see the Two Wheel Deal in action and help users become acclimated with how the vehicle can be operated. This would be the same DVD that users would watch when they buy their first Two Wheel Deal from the dealer. Once the users have watched the video they would then be able to try operating a practice Two Wheel Deal at the dealer with the aide of a trained professional. By going through this quick and easy onsite training, the company and engineers can know for sure that the new users won’t injure themselves when attempting to ride the Two Wheel Deal for the first time at home.

One final ethical concern and additional way to prevent user injury is the utilization of safety mechanisms. One safety mechanism that would be added to the Two Wheel Deal is a tone that would sound when the rider was getting close to the limit the vehicle can be tilted. This would help ensure the rider doesn’t tip the vehicle over. In addition to the warning tone a function would be added in software that caused the wheels to stop or slow if the vehicle is tilted too far in either direction. This would prevent the vehicle from running over anyone if tilted out of control. The program functionality would continue as normal once a certain angle was reached. Another safety measure that would be added would be an increased number of accelerometers and gyroscopes. These extra sensors would act as redundant circuitry that would verify the readings from the accelerometer and gyroscope. If the sensors were not then the redundant circuitry would be utilized, and a warning would be displayed on the LCD screen to let the rider know there is a sensor issue. A watchdog program would also be implemented to ensure the microcontroller continues to function correctly. Finally a function would be written to allow the vehicle to come to a slow but controlled stop if the battery voltage or any PCB voltage dropped to an unsafe level rather than simply shutting down completely. Those are some of the ethical challenges that would be faced if the Two Wheel Deal would be released to the public.

6.3 Environmental Impact Analysis

In addition to the ethical concerns addressed earlier, there are environmental concerns that need to be considered. These will be divided into three stages that occur throughout the product lifecycle. The first stage is the manufacturing stage. The greatest cause of environmental pollution during the manufacturing stage is the creation of the PCB due to the large amount of chemicals and natural resources that are needed. Typical pollution includes industrial wastewater from PCB rinsing; alkaline/acids from cleaning and etching the PCB; and potentially hazardous metals such as lead, chromium, and palladium from contacts and traces [20].

All of these potential hazards have the opportunity to be reduced using certain processes available to all PCB manufacturers. Examples include material substitutions such as using alkaline solutions rather than solvents for degreasing operations, adding multiple tanks in the rinse system to improve efficiency, and minimizing bath solution concentrations to the lower end of their operating range [21]. These concerns would be addressed by choosing a PCB manufacturing company who follows these procedures and is RoHS compliant, which is a certification restricting levels of certain hazardous materials [22]. Advanced Circuits, which made the PCB for the Two Wheel Deal, use techniques similar to the ones described above to help reduce environmental impacts caused by PCB manufacturing. The company chosen to mass produced Two Wheel Deal PCBs would also have to have one of the lowest manufacturing costs to ensure total production costs for the vehicle are at a minimum. This would ensure that environmental hazards created from the PCB would be at its lowest. One other minor concern is the choice of hardware components. Many various components include lead and other metals that can pose as potentially dangerous to the environment. Ensuring that the hardware components chosen are RoHS compliant as well would help decrease environmental impact.

The next stage is the normal use stage of the Two Wheel Deal's life cycle. Thanks to the emission-free, electric drive there are very few environmental concerns during this period. The only main concern is the use of sealed, lead-acid batteries. These batteries contain sulfuric acid and heavy metals, both of which are environmental hazards [23]. Caution must be taken when handling these batteries to ensure the plastic casing is not punctured or broken to prevent leaks from occurring since the sulfuric acid can burn through almost anything. Directions on how to handle the batteries would be included in the user manual and described in the training session provided when purchasing a Two Wheel Deal. Another caution must be taken while recharging the batteries. If caution is not taken and the battery is overcharged there is a chance it can explode which would spray sulfuric acid everywhere. There would be a warning placed on the battery as well as a reference to the user manual for helpful directions for recharging the batteries. There would also be recommendations on various chargers that work the best.

The last stage of the Two Wheel Deal life cycle is the disposal/recycling stage. This is the most important stage since many parts of the vehicle can be reused or need to be disposed of properly. Proper disposal of the vehicle components is recommended because many of the metal pieces in the frame can be recycled. The motors can be refurbished and the batteries can be recycled to help create new batteries from recycled materials. The rubber tires and plastic wheels can also be recycled easily. Finally the PCB contains lead and other dangerous metals. These metals can also be found in the hardware components and LCD.

To facilitate environmentally safe and practical disposal and recycling for the Two Wheel Deal, a combination of core charges would be added to the initial cost of the vehicle. Core charges would be created for major components such as the motors and batteries which can easily be recycled or reused if they no longer work or simply need repaired. The last core charge would be for the entire rest of the vehicle. The way the process would work is that if a customer needed a new battery or motor, he/she would take the item to a Two Wheel Deal Recycling Center and receive the core charge for bringing the item back, after verifying the customer purchased that item, to be disposed of properly. He/she could then purchase a new item if necessary. This program which gives the customer a refund helps motivate customers to recycle. The Recycling Center would then send the item to be reworked or recycled in places such as a battery recycling center or a company that refurbishes used motors. Directions, details, and contact information for this process would be specified in the user manual.

The core charge for the rest of the vehicle would work in a similar fashion. If the vehicle is no longer in operating condition or the customer decides he/she no longer wants it, then it can be returned to a Recycling Center. After verifying the customer purchased that vehicle, the core charge would be awarded depending on how much of the vehicle is returned. If there are items missing such as the PCB or the LCD, then a certain percentage of the core charge corresponding to that item would not be returned to the customer. The Recycling Center would then disassemble the vehicle to determine if items or components can be reused in new vehicles.

Anything that could not be used would be sent to appropriate recycling centers. Directions, details, and necessary contact information on this process would be included in the user manual.

6.4 Summary

This is a summary of some of the ethical and environmental concerns that would come with making the Two Wheel Deal available to the public. The ethical concerns are centered mostly on the safety of the rider as well as preparing the rider and vehicle for a variety of environments and terrains. The environmental concerns are focused mainly on limiting pollution produced from creating new PCBs as well as ensuring easy and motivating processes to ensure safe disposal or recycling of hazardous materials contained in the Two Wheel Deal.

7.0 Packaging Design Considerations

7.1 Introduction

The Two Wheel Deal is a self-balancing transporting device similar to the Segway. Both devices gather information from the accelerometers and an angular rate sensor in order to drive the wheels in a way to keep the rider's center of gravity above the wheels while displaying information on an LCD.

In order to meet our design goal, our project must be packaged in a certain way. Our finished product should weigh no more than 80 pounds to allow for easier turning and so that the user can lift it into a car or up stairs. The motors, batteries, motor controllers, microcontroller, and other circuitry are all secured under the frame which provides more passenger room. The steering post is attached to the base plate and will contain the LCD and turning mechanism.

7.2 Commercial Product Packaging

Because Segway has patented the distinctive method that permits the device to balance itself (dynamic stabilization) on two wheels using a gyroscope and accelerometer, Segway has a monopoly in this particular market. Therefore, no other commercial product is available for comparison [24].

7.2.1 Product #1 Segway i2

The Segway is a two wheeled personal transportation device that uses the inverted pendulum control problem as its basis of operation. The vehicle has a basic design. It consists of a platform to stand on, a vertical shaft that ends in handlebars, two brushless DC motors, and two wheels. The whole assembly weighs about 105 lbs and leaves a 19 inch by 25 inch footprint. The Segway uses 19 inch tires that allow for 3.5 inch ground clearance. It has adjustable vertical shaft for handle bars that leans to the left and right to allow control of turning [25].

The Segway has some positive and negative aspects of its design. First of all the overall design is very good. It is fairly small, sleek, and efficient. It is basically a platform on wheels with handlebars. It has large wheels which give plenty of ground clearance to prevent any problems that could arise such as grounding the vehicle out while driving through various terrains. It also has easily adjustable handlebars to fit various riders comfortably without much trouble. The Segway has a small display screen that shows important data to the rider but it is rather small. This can become hard to read for riders with vision problems or in dark areas. Also the Segway uses handlebars that tilt side to side to control turning. This uses a lot of movement and can become bothersome to different riders. It also has a width of 25 inches which can be a tight fit through certain smaller than average doors [25]. This can present a problem in different situations.

7.2.2 Product #2 Trevor Blackwell's Balancing Scooter [38]

As a personal project, Blackwell designed and built his own two-wheeled balancing transporter. He used electric wheelchair motors just like the ones in our design only his were a little less powerful. Instead of using 2 NiMH batteries like the Segway or 2 sealed lead acid ones like us, he used 20 NiMH battery packs from remote controlled cars. There are many similarities between his project and the Segway, but there are also many differences. The main concept of using a gyroscope, accelerometers, and microcontroller are the same, but Blackwell doesn’t have nearly as many safety features or error checking. Because of all the battery packs he’s using, he only achieves about an inch or two of ground clearance. He can use larger tires, but leaving the base plate low is safer for the rider in case of a crash. In order to steer Blackwell’s scooter, one must turn a small potentiometer. However, he does use a dead man’s switch and the program shuts off the motors once the scooter is tipped over 45 degrees.

7.2.3 Product #3 The DIY Segway at Wayland Highschool [39]

This was a group of high school students and four MIT students who built a device similar to the Segway. They used 12.5” wheels and cleared the ground by about 5 inches. Their device runs on a single 12V sealed lead acid battery from a motorcycle. Some goals they wished to achieve include weighing less than 50 pounds (battery included), less than $1000, fit through doorways, and have “lean steering” technology. They are using a PIC microcontroller, accelerometer and gyroscope, and a dead man’s switch. They use XBEE and a laptop computer to wirelessly program the micro and read the sensor information in real time.

7.3 Project Packaging Specifications

The Two Wheel Deal will copy the basic design of the Segway. It is simple, concise, and efficient. The Two Wheel Deal will have a simple rectangular platform. We plan to mount our batteries and circuitry underneath the foot plate. This will allow for more foot room on top for comfort and potentially some luggage. It will also have a vertical shaft with handlebars but they will not be adjustable. The steering post will be fastened at the base and angle slightly outward from the rider so that the rider can relax his arms and ride comfortably. One difference from the Segway is that the Two Wheel Deal will not use a leaning handlebar shaft for steering control. Instead it will use a thumb joystick similar to that found on gaming console controllers. This will allow less movement to be necessary in order to control the vehicle. Attached to the handlebars of the Two Wheel Deal will be a 4 line by 20 character LCD display. This is larger than the small pocket display that is found on the Segway. The character LCD display will be able to present more information to the rider than on the Segway. The Two Wheel Deal will also have two independent motors similar to the Segway, but it will use brushed DC motors rather than brushless. The wheels for the Two Wheel Deal will only be 16 inches rather than the 19 inches found on Segways. The Two Wheel Deal will have more ground clearance than the Segway due to the higher placement of the rider's platform. One final difference in the basic design is that the Two Wheel Deal will have a smaller footprint than the Segway. The larger footprint of the Segway makes it harder to fit through some smaller doors which can be eventually become tedious work for the rider. Because the Two Wheel Deal has a smaller footprint, this allows greater maneuverability.

7.4 PCB Footprint Layout

For our footprint, we put the microcontroller in the center in order to easily route all the inputs and outputs. Our gyroscope and accelerometer are placed in close proximity with the microcontroller. The two motor drivers will be positioned on the outside of the PCB in order to be close to the respective motor. With this arrangement, we can separate the analog and digital portions of the circuit as much as possible. The estimated size of our PCB is approximately 11.5x3.5 inches.

7.5 Summary

Our project mimics the Segway personal transportation device that was released in December 2001 [26]. We are using a gyroscope and accelerometers like the Segway, but our motors can deliver more horsepower. Our device will include safety checks to verify that the rider is present before traveling any further forward. The estimated cost of our Two Wheel Deal will be approximately $1500 and will weigh no more than 80 pounds once completed.

8.0 Schematic Design Considerations

8.1 Introduction

Due to its duties as an upright balancing transportation device, The Two Wheel Deal requires a unique mixture of high sensitivity and high power circuitry in order to properly balance and transport a person. On the sensor side, it needs a precise sensor and microcontroller circuit to fine tune the demanded motor torque based on readings from multiple sensors and user inputs. On the power side, this vehicle needs high current motor controllers to bridge the gap between the microcontroller and the powerful DC motors on board, along with high capacity batteries to supply the required current.

8.2 Theory of Operation

8.2.1 Power Supply

All of the power for The Two Wheel Deal comes from a pair of 12V sealed lead acid wheel chair batteries wired in series to provide the circuit with 24V. The batteries are used to supply current primarily to the motor controllers, which direct the current through the 24V NPC-T74 motors. The 24V from the batteries is stepped down to 12V using a linear voltage regulator. The step down to 12V is needed to supply voltage to the FET driver, which provides the gates on the MOSFETs with the high voltages needed to turn them on. The FET driver will not be run directly off of the 12V from one battery because if the voltage on that battery dips below 12V, the FET driver may not work correctly. Though this is an inefficient means of regulating a large voltage difference, the simplicity of the circuit and the resilience to voltage ripple caused by the motor controllers far outweigh the additional current consumption which, when compared to the current consumption of the motors, is miniscule. The voltage is again stepped down from 12V to 5V using a linear voltage regulator to ensure that the microcontroller and sensor circuit receive a stable voltage. The constancy of the 5V line is critical for the proper interaction of the sensors and the microcontroller. Fluctuations in the input voltage of the sensors will result in undesirable changes on the output of the sensors. The microcontroller, sensors, PLDs, LCD, and various other support devices are powered by the 5V regulator. All components outside of the motor controllers are 5V devices to simplify the power supply circuit, as well as eliminate any need for level translators or amplification hardware.

8.2.2 Microcontroller

The ATmega32 microcontroller interfaces to all of the major components of the vehicle. It runs off of an external crystal oscillator at the microcontroller’s maximum clock speed of 16 MHz to maximize the available computing power. It uses the ATD module to read an analog signal from the accelerometers, the gyroscopic angular rate sensor, the steering joystick, and the battery level meter. Based on the incoming data, it computes the duty cycle of two outgoing PWM signals that are used to command torque from the motors. At the same time, the microcontroller interfaces to an LCD screen to give the driver feedback on the state of the vehicle.

8.2.3 Sensors

A dual axis accelerometer and a single axis angular rate sensor are used to determine the angular position and angular velocity of the vehicle about its drive axis. Both sensors have analog outputs that are fed directly to the microcontroller’s ATD channels. Both sensors have internal signal conditioning circuitry, reducing or eliminating the need for filtering circuitry. Physical orientation of each sensor is extremely important for obtaining the correct information on the vehicle’s movements [27][28]. To facilitate proper mounting of the sensors without resorting to extraneous mounting of each sensor, the entire microcontroller and sensor PCB is mounted perpendicular to the ground along the centerline of the vehicle.

Battery level monitoring is performed by using a simple high resistance voltage divider connected directly to the battery. The 24V battery voltage is divided down to 5V to allow the ATD converters to read changes in the voltage of the battery. Knowing the state of the battery’s charge will allow the controller to compensate for a low battery by increasing the duty cycle of the PWM signal going to the motor controllers, keeping the vehicle’s performance constant over the life of the battery.

Rider detection will be performed by reading the tilt angle returned by the accelerometer. This lets the system know when the rider has dismounted (either willingly or unwillingly). Logic in the software algorithm will shut the motors off in an event of a crash.

8.2.4 Motor Controller

Each motor controller is controlled using a PWM signal from the microcontroller running at a frequency of approximately 16 kHz. A lower frequency was initially proposed in order to minimize switching losses on the MOSFETs, but using a low frequency signal can cause an audible whine from the motors. The PWM signal is not fed directly to the MOSFETs, but instead it is fed to a FET driver that takes care of many of the important details of driving an H-bridge, such as applying the right voltages to the MOSFET gates and inserting a slight delay before switching from forward and reverse to prevent a circuit destroying shoot-through situation.

The FET controller controls a simple MOSFET H-bridge. Each leg of the bridge holds 3 MOSFETs rated at 100A continuously. This gives each one of The Two Wheel Deal's motor controllers a theoretical current capacity of about 300 amps, though given the passive cooling system, 150A is a more realistic value [29].

Even though the FET drivers help prevent shoot-through by inserting a slight delay before changing directions [30], special circuitry has been added to the H-bridges to compensate for the large MOSFETs used in this design. The large gate capacitances (on the order of a few µF) cause a sizable delay in the turn-off time of each FET, and if the opposing set of FETs turn on before the first set turns off, shoot through will occur. Resistors have been added in series with the gate of each transistor to slow down the turn on time, and a Schottky diode wired in parallel with the resistor aid in speeding up the turn off time of each FET by pulling the charge off of the transistor's gate.

Each motor controller requires that the incoming PWM signal be switched between two separate pins to switch the motor controller between forward and reverse. Since the microcontroller only has enough PWM channels to supply 1 PWM signal to each motor controller, additional logic must be added to allow for a single PWM signal to command both forward and reverse. Though this logic can be implemented using a few discreet logic gates, a single PLD is used to simplify this circuit and allow for future expansion.

The rapid switching of high current inductive loads makes the motor controller an electrically chaotic part of the circuit. To provide a measure of safety to the microcontroller and logic circuits, the motor controller circuits are separated from the microcontroller circuit by running each PWM signal through an optical isolator.

8.2.5 Human Interface

An LCD screen is used to provide the user with vehicle information such as approximate vehicle speed and battery life. The microcontroller interfaces to the LCD screen directly using a parallel interface. Steering of the vehicle is implemented by reading the voltage output from a variable resistance joystick circuit. This value is sent directly to an ATD channel on the microcontroller.

8.3 Hardware Design Narrative

The PWM module of the ATmega32 is used to send a torque demand to the motors. The microcontroller has only two 16-bit PWM outputs, and both will be utilized. These outputs correspond to pins OC1A and OC1B. General I/O pins PD0 and PD1 have been chosen to control the direction of the motors primarily due to their close proximity to the PWM output pins.

The ATD module is used extensively for reading the values coming from the sensors. All of the sensors have a 0-5V output range, so they can be directly connected to the ATD pins. The dual axis accelerometer's two outputs uses ATD pins PA1 and PA2, the output of the angular rate sensor uses PA3, the 2.5V reference voltage of the angular rate sensor uses PA4, the steering joystick uses PA5, and the battery level circuit is connected to PA0.

The LCD screen will take up the most I/O pins. While the parallel interface of the LCD could be using the SPI and a shift register to save microcontroller pins, a high number of free I/O pins make this an unnecessary addition to the circuit. The LCD screen requires 10 pins to operate. The 8 data bits will use pins PC0-PC7. These pins are controlled internally by a single data register. Putting all of the LCD data pins together simplifies the software algorithm for writing data to the LCD. The screen needs a register select (RS) and a clock signal (E) as well, and these are assigned to pins PA7 and PD7 respectively. Placing these signals in close proximity simplifies the physical routing on the PCB. A summary of the microcontroller pin assignments can be found in Table 3.1.

Table 8.3 – Summary of Microcontroller Pin Usage

Signal	Pin Name	Pin Number
PWM Motor Controller Torque Demand	OC1A, OC1B	19, 18
Motor Controller Direction Control	PD0, PD1	14, 15
2-axis Accelerometer	PA1, PA2	39, 38
Angular Rate Sensor	PA3,PA4	37,36
Steering Joystick	PA5	35
Battery Level Monitor	PA0	40
LCD Data	PC0-PC7	22-29
LCD Register Select (RS)	PD6	20
LCD Clock (E)	PD7	21

8.4 Summary

The Two Wheel Deal’s circuitry has been carefully designed around creating a circuit that is both sensitive enough to properly detect and react to small changes in the vehicle’s dynamics, yet powerful enough to move a person. The lead acid batteries supply ample current to the motors, while the linear voltage regulators provide a stable power supply to the voltage sensitive digital components. A microcontroller sits at the center of the design, interfacing directly to all of the sensors and user inputs while interfacing through isolation circuits to the motor controllers to prevent damage to the delicate circuitry surrounding the microcontroller. The end result is simple yet effective realization of a circuit that performs all of the calculations, sensing, and power delivery required for The Two Wheel Deal.

9.0 PCB Layout Design Considerations

9.1 Introduction

The “Two Wheel Deal” is a personal transportation vehicle which balances itself automatically using angle and angular rate sensors. The “Two Wheel Deal” PCB will consist of three individual sections: two high current h-bridge motor controllers and a controller circuit. The controller circuit will contain the microcontroller and various other inputs to the microcontroller. This circuit will also contain the sensors used to balance the vehicle. The h-bridge PCB will contain all of the transistors for the voltage switching as well as an IC to time the switching as to not short circuit the voltage supply. These two h-bridges are used to control each motor individually.

9.2 PCB Layout Design Considerations - Overall

The “Two Wheel Deal” PCB has a couple design considerations. The PCB will be split into three physically separate sections which are the left motor controller, right motor controller, and controller circuit. Since the motors are similar, the right and left motor controllers will be similar, and the individual PCBs will be place near the respective motor on the vehicle. The controller PCB will be mounted under the rider platform because it contains sensors that are sensitive to the orientation of the vehicle.

First, the motors that were selected to drive the vehicle require a large amount of current, up to 210 Amps for max torque. This amount of current creates two major considerations which are heat and trace size for the motor controllers.

To deal with the heat of the transistors due to high current and switching, the transistors have been physically arranged in a square-like configuration. The plan is to bend the transistors off of the board and secure them directly to a heat sink. Some of the transistors will have to be insulated as to not ground the wrong transistors. The heat sink would possibly be made of aluminum or some material that has high heat transfer characteristics. There are two purposes that this design serves. This configuration allows a way to secure the PCB to the vehicle, and allows a way to dissipate heat that is generated by the transistors.

The final design did not place the transistors around the perimeter for mounting the PCB. The heat issue was addressed by the use of a simple computer fan. The fan was added on top of the FETs to cool them with ambient air temperature. This method should provide enough cooling to keep the FETs from overheating.

To deal with the high current, a large trace size is needed. This trace is confined to the outer areas of the board. This design will isolate the high current from the more sensitive part motor controller that is positioned in the middle of the PCB. It is determined that the trace size needs to be 500 mils to deal with the maximum possible current. 100 mil traces will be tapped off of the 500 mil trace to provide the current to the transistors. The design included three transistors in parallel so that the current could be split between them. Once the board arrives, another conductor will be added to the trace to increase the thickness. This conductor will consist of more copper wire or a solder trace.

The controller had few project specific considerations because the control algorithm is fairly simple. The power and ground traces were placed around the outer part of the PCB. The microcontroller is the central part of the circuit and traces split into many directions, therefore the microcontroller was placed close to the center.

9.3 PCB Layout Design Considerations - Microcontroller

The microcontroller and all of the other components that are on the PCB are all run at 5V, therefore there is a single supply rail. The microcontroller that was chosen for out project was the Atmel ATMega32. This is a 40 pin DIP package as seen in the datasheet. [32] Before considering trace paths, an effort was taken to determine which pins could be grouped together for the controllers I/O. Pins that had a common purpose were grouped together which made the trace routing considerably easier.

First, the microcontroller contained enough pins so that a parallel communication to an LCD is possible. The LCD requires eight data pins and four power and ground pins. One port, of the controller is used for data transmission and the other four pins are routed to the power and ground pins of the controller. The LCD header was placed near the port that it uses. Another input header was added to the same part of the microcontroller because it will use the rest of the input pins on the same said as the LCD. Next, the microcontroller requires an external oscillator circuit. The oscillator was placed close to the microcontroller so that the traces were short and will not pick up much noise before they connect to the microcontroller. Then, the two input sensors were placed in close proximity to each other so that the input data from both sensors would be accurate, and they were wired up according to the datasheets. [33],[34] Unreliable input from these sensors could cause an unstable system. These sensors were also placed close to the microcontroller. A PLD is required to interface the microcontroller to the h-bridge controller signals. The PLD, optical isolators, and motor controller connectors were placed close together to make routing the traces easier. Finally, a header was added so that the microcontroller could be programmed easily. A rough draft the microcontroller PCB can be viewed in Appendix B.

9.4 PCB Layout Design Considerations - Power Supply

The power supply for the "Two Wheel Deal" will consist of two – twelve volt batteries tied in series to provide the 24V dc power for the PCB. The unregulated 24V will be fed into the left and right motor controllers. A 12V regulator on each motor controller PCB will be added because the motor controller driver chip runs at 12V. One of these 12V regulated signals will be sent to the microcontroller board to a 5V regulator. All of the parts on the microcontroller board run at 5V, so only 1 regulator is needed. Also there will be an attenuated unregulated 24V signal sent to the microcontroller straight from the battery. This signal will be used to determine the battery level. This is a low current application, and should not affect any other signals on the microcontroller board. 500 mil traces will be placed on the motor controller board to supply enough current for the motors. Again, once the board is made, an additional conductor will be added to the trace to increase its volume to provide enough current capacity. This trace will be routed around the outside of the board to try to keep it away from the motor driver and power the transistors A rough draft of the PCB layout can be seen in Appendix A.

9.5 Summary

In conclusion, the PCB will have two breakout boards for each motor controller and a PCB for the microcontroller. Each motor controller will have switching transistors that will provide the h-bridge functionality that is needed. These transistors will be switching high current, therefore a 300 mil trace will be used to supply this current. This trace will later be thickened so that it doesn't overheat and become unusable. The h-bridge transistors will also be a means to secure the PCB to the vehicle.

The controller PCB contains all of the sensitive digital circuitry, and it will be isolated from the motor controllers. The microcontroller will be clocked using an external oscillator to achieve the necessary clocking speed. It will contain the angular inputs, and it will provide outputs to LCD screen as well as the motor controller driver.

Overall parts were placed close to the other parts that they would be interfacing with to reduce trace routing and trace lengths, which in turn reduced the amount of EMI. A careful planning of placement for the parts makes the PCB layout process run much smoother and take considerably less time.

The initial design had the power and ground traces routed around the perimeter of the PCB. This design was rethought after some advice. Routing the power and ground through the middle of the board results in less interference than a circular design. This new design also made the PCB smaller and more compact. The smaller PCBs fit better underneath the chassis of the Two Wheel Deal, where area is scarce due to the batteries and brainboard. The new design can be seen in Appendix D.

10.0 Software Design Considerations

10.1 Introduction

The Two Wheel Deal is a personal transportation vehicle based on the control theory principle of the inverted pendulum [35]. The inverted pendulum is an inherently unstable system that consists of a mass atop a rod that is hinged to a cart. In order to stay upright a force is applied to the cart in a linear direction so that the cart is always under the system's center of gravity. This is usually accomplished using some sort of feedback control. With this principle in mind, there are a few major software design considerations.

The first and most important consideration is to ensure that there is a constant, correct, and efficient reading of analog outputs from the accelerometer, gyroscope, joystick, and battery. This data determines how the system needs to respond to keep the vehicle stable. Another major design consideration is to accurately determine the angle of tilt from vertical that the vehicle has sustained. This way the proper motor speed is obtained to keep the base of the vehicle under the center of gravity. One final major design consideration is to use a quick and efficient method of control that will not allow a lot of overshoot of the vertical angle yet will be fast enough to keep the vehicle upright. One other design consideration is to display important information to the rider such as battery life and other operating information.

10.2 Software Design Considerations

10.2.1 Memory Mapping

The microcontroller [36] that was chosen has 32 kB of self-programmable Flash memory. The Flash memory is addressed from $0000 to $3FFF. All static data which includes the Two Wheel Deal program as well as any constants will be stored in Flash memory. The current size of the program is about 8 kB and the final program size should be about 10 kB. This will leave plenty of space left to add additional features or functions. The microcontroller has 2 kB of SRAM and it is located from $0000 to $085F. The general purpose working registers make up the first 32 addresses in the SRAM space. The next 64 addresses are for the I/O registers and will be where the variables are kept. The stack will start at the bottom address $085F, and it will work its way up as items are added or back down as items are removed. No heap space will be used since all variables are declared when compiled.

10.2.2 Mapping of External Interfaces

There are a large number of components that are interfaced to the microcontroller with every port being used. Information such as the address [37], the function, and the peripheral, if applicable, of every port that is used can be found in Table 1 of Appendix C.

10.2.3 Utilization of Integrated Peripherals

The Two Wheel Deal utilizes the following peripherals: ADC, PWM, and TIM. The interrupt driven ADC is used to read the values from the sensors at a constant time interval. The 16-bit PWM is used to provide a signal to the motor controller that corresponds to the appropriate output torque of the each motor. The duty cycle of the PWM is based on the angle of the vehicle from vertical. The timer is used to create an accurate time base in which to run the main program.

The analog to digital register ADMUX is set so that the ADC uses AREF which is set equal to 5 V, the results are left adjusted so that the result can be read in one byte, and the ADC is initially set to start on channel 0. The register ADCSRA is set so that the ADC is enabled, there is no auto triggering, ADC interrupts are enabled, there is a clock prescalar of 128, and the ADC conversions are started. The PWM register TCCR1A is set so that on compare match OC1A/OC1B are cleared and force output compare on A/B disabled. The register TCCR1B is set so the input noise canceller is disabled, a falling edge is used for input capture, fast PWM mode is used with a top value of ICR1, and there is no clock prescaling. The register ICR1 is set so that the PWM frequency is 2 kHz. The timer register TCCR0 was set so that there was no force output compare, it used Clear Timer on Compare (CTC) mode, there was normal port operation, and a clock prescalar of 64 making the timer run at 250 kHz. The register OCR0 was set equal to 250 so that the timer interrupt service routine would run every 1 ms. Finally register TIMSK was set to allow output compare interrupts but not overflow interrupts. These register values can be viewed in Table 2 located in Appendix C.

10.2.4 Overall Organization of Application Code

The overall organization of the application code is a hybrid or flag driven program. This means that the main loop of the program is a simple “round robin” loop that checks to see if certain flags are set. Those flags are set by interrupt service routines that occur at a fixed and known rate. The rationale behind this is since the rider will most likely be in constant motion, it must be ensured that the controller will check the tilt angle and update the motor torque at a fast and constant rate. This is best performed using an interrupt that occurs at a known interval and increases a counter each time it is executed. Once enough interrupts have occurred a flag is then set in the "round robin" main loop. Since this main loop is small and only checking three flags, it can be assured that the balancing algorithm function will be run at a precise rate.

10.2.5 Debugging Provisions

There is nothing specific that has been included in the code that deals only with debugging. The microcontroller is programmed using the SPI and an inline programmer. The LCD has been used as a terminal to help with debugging code. Since it was the first interface that was programmed, it has been used to display register, variable, and port information to help locate sources of error in the software.

10.3 Software Design Narrative

The following sections include a detailed narrative of each module used throughout the Two Wheel Deal software. Each module is linked to allow the actual code to be viewed online. Also, a hierarchical arrangement of the code modules including the functions included in them can be viewed in Appendix B.

10.3.1 Main Module (main.c)

This module contains the "round robin" polling loop as well as the ADC and TIM interrupt service routines. The main loop serves as an endless polling loop that checks if the tens (LCD), huns (balance), and ones (battery) flags are set. If one of them is then it calls that function otherwise is continues looping. This module has been written, programmed, and verified to work correctly. The flowchart illustrating the activity of the main loop can be viewed in Figure 1 of Appendix A.

The first operation of the main loop is to call the function ioinit() which sets the data direction registers for Ports A, C, and D. It also calls varinit() which initializes the battery filter array so that the battery does not read 0 V when powered up. It then enables interrupts and calls the function PERinit() which initializes all the peripherals and interfaces. This includes the TIM, PWM, ADC and the LCD. In that function it calls lcdcch() which loads custom characters into the LCD driver. Next it prints an introduction greeting on the LCD for 2 seconds using print_greeting() and delay_ms() and then clears the display. Finally it enters the endless while loop where it checks the huns and tens flags repeatedly.

If the huns flag is set then 10 ms has passed and the balancing algorithm function is called. If the tens flag is set then 100 ms has passed and the LCD update function is called. Finally if the ones flag is set then 1 second has passed and the battery filtering function is executed. Each of these functions is described in detail later.

The ADC ISR is used to continually check and update the outputs from the accelerometer in the X and Y directions, the gyroscope, the joystick, and the battery voltage divider. The flowchart describing the routine activity can be found in Figure 2 in Appendix A. It works by waiting until the ADC interrupt bit is set which means a conversion has been completed, and then it reads the ADC data register result from the completed conversion and stores it in the corresponding spot in a data array. It then increments the ADC channel counter, resets the ADMUX register for that channel, starts the ADC conversion, and then leaves the routine.

The TIM ISR is used to keep a continuous time base for the system and the flowchart can be viewed in Figure 3 of Appendix A. It works by waiting until the OCIF bit is set indicating that the timer counter equals the compare register. It then sets the 1 ms flag and increments the hundredths, tenths, and ones counters by one. Then if the hundredths counter equals 10 the huns flag is set and the hundredths counter reset. The same occurs with the tenths counter except it must equal 100, the tens flag is set, and the tenths counter reset. Finally if the ones counter equals 1000 then the ones flag is set and the counter is reset. The routine is then exited.

10.3.2 LCD Update Module (lcd.c)

This module contains the lcd_update() function which is called by the main function to update information displayed on the LCD every tenth of a second. This module has been written, programmed, and verified to work correctly. The flowchart for this function can be viewed in Figure 4 of Appendix A. The first operation performed by lcd_update() is to read the filtered battery level variable and determine how many full and empty blocks are necessary for updating the custom battery symbol.

Then the cursor is positioned at the first character of the first line using lcd_rc(rows,columns). This function uses the inputted rows and columns values and sets the corresponding cursor address using the lcd_addr() function which simply writes a command to the LCD driver. Next the strings “Angle:” and “Battery:” are written to the LCD using lcdstr(). This uses a character array and a pointer to display a string character by character on the LCD until a null character is reached. Then the cursor is repositioned on the second line and the angle is displayed using lcdnum(). This function is used to display an inputted float number to two significant digits. This is done by casting the original float to an int. The int is then displayed using lcdrite() which simply writes data be displayed on the LCD. The float and int are then multiplied by 100 and subtracted from each other. This remainder is the decimal portion that is then displayed on the LCD using lcdrite().

Next the cursor is repositioned on the third row and displays the left side of the custom battery symbol. Then using the number of full and empty blocks found earlier displays the filled and empty parts of the battery then displays the right side of the battery. This is done again on the fourth row to complete the symbol. Finally the function is exited to return to the main loop.

10.3.3 Battery Filter Module (bat.c)

This module contains the battery_alg() function which is called by the main function to average the values read by the ADC from the battery voltage divisor to determine the voltage left in the batteries. The flowchart is located in Figure 5 of Appendix A. It is called every second to read a new value from the ADC and determine the new average which is used when updating the LCD. The reason a filtered battery value is used to display the voltage left on the batteries is because if there are power spikes or dips due to increasing or decreasing the motor speed this will not be displayed to the rider. This module has been written, tested, and verified to work correctly. The first operation the function performs is to read value from the ADC and store it in the battery filter array. It then increments the array counter and uses a while loop to sum the array. It then averages the value and converts to volts to be used in the LCD update function.

10.3.4 Balance Update Module (bal.c)

This module contains the balance_alg() function which is called by the main function to determine the angle of tilt from vertical and update the motor controller PWM signals and therefore the motor torque accordingly. The function executes every hundredth of a second. This module has not been tested but is partially written and is laid out in pseudocode. The logic and activity of this function was based on open source pseudocode provided by Trevor Blackwell [38] as well as open source code provided by a group of MIT/Wayland High School students [39] while designing a similar transportation vehicle. The flowchart can be viewed in Figure 6 of Appendix A. The first operation performed is to read the ADC values for x axis accelerometer, y axis accelerometer, gyroscope, and joystick. Then a constant is subtracted from the x and y axis values to account for the positioning of the accelerometer. The angle in degrees is found by taking the arctangent of the y axis divided by the x axis then converted from radians.

The balancing torque is then found using a Proportional-Derivative Controller. The angle is multiplied by an appropriate constant and added to the rate multiplied by an appropriate constant. These two values represent the proportional and derivative values respectively. The balancing torque value is then set as the base motor value for the rest of the algorithm. Turning is then accounted for by first reading the value from the joystick and subtracting a constant from it to make a left turn a negative number and a right turn a positive number. This is then subtracted from the base motor value so that if a left hand turn is needed the left motor slows and vice versa for a right hand turn. Finally the motor value for the left and right side motors are written to the left and right PWM duty registers respectively, and the function is exited.

10.4 Summary

Those are the major design considerations for the Two Wheel Deal as well as the software design narrative describing the overall code functionality. One final note is that as of right now there are no safety functions or considerations taken into account. These are currently out of the scope of the project, but if there is time after the main project specific success criteria are completed, then they will be added as needed.

11.0 Version 2 Changes

Version 1 of the Two wheel deal works fine, but many improvements could be made with more time and money. Here are some of the improvements for Version 2 of the Two Wheel Deal. First, the brushed dc motors that currently provide the torque would be replaced by brushless dc motors. The brushless motors will mean less maintenance as no brushes will every need to be replaced. Also the brushless dc motors will be somewhat quieter than the brushed motors. The downfall to brushless dc motors would be that they run on 3 phase AC current, so different motor controllers would need to be designed. More research would determine if this would be economically feasible. Also the brushless dc motors could provide a smaller footprint than the current 28 inches. This would make travel through doorways or other tight spaces much easier. Currently, driving the Two Wheel Deal through a doorway takes some skill. The brainboard PCB could be redesigned to be smaller which would help in the smaller footprint. Also, the ISP header on the PCB would be placed in a more accessible position as it is difficult to reach without removing a bolt.

Currently, motor hubs needed to be designed to mechanically interface the rotor to the wheels. It is very difficult to remove the wheels to maintenance the rotor if bolts come loose. This problem could be resolved by finding wheels that connect directly to the rotor, or finding a motor that works with the wheels. Another solution is to redesign the hubs so that a wrench is more accessible to the bolts needed to disconnect the wheel.

Next, the current motors are rated for 24V, but they can be run safely at 36V. Currently, there is no room left to add another battery. Our group would make the new design expandable or add more room so that new items such as batteries could be added if needed in the future. Also, nickel cadmium batteries would replace the current sealed lead acid batteries. Nickel cadmium batteries are smaller and more power dense, which would allow for the smaller footprint. The drawback to nickel cadmium batteries is that they are much more expensive.

Subsequently an easier recharge circuit would be nice. Currently, wires need to be disconnected and reconnected to recharge the batteries. This leaves room for error to hook up the charger the wrong way or damage the batteries. A system of relays could be designed so that the user could hook up the charger, and the system would automatically know to charge the batteries.

With all of the expensive and sensitive electronics on the bottom of the Two Wheel Deal, a waterproof cover needs to be added to keep dust, water, and debris away. Also, a cover could be added to the LCD screen as it can be damaged rather easily in a crash.

Presently, the linear regulators are heating up and heat sinks were added to dissipate the extra heat. These linear regulators would be replaced with stepping regulators which are more efficient and don't heat as fast. Another solution would be to position the current linear regulators underneath the cooling fans that are already in place.

Another problem with the PCBs were that some footprints were wrong. This meant that traces needed to be cut and rewired. These revisions were made so that nothing has to be rewired. Also, the accelerometer has a current rating of 1.7g. The old accelerometer would be replaced with a 1.2g which provides better resolution of the angle. In addition, more accelerometers and gyroscopes would be added for redundancy and error checking. Presently, if the accelerometer or gyroscope malfunctions, the vehicle can't be ridden and unsafe. This redundancy will add safety was well as better resolution for balancing the vehicle. Error checking would be added to software to determine if a part was bad, and the reading from the bad part would be discarded. The new software would change the switching frequency of the FETs to audible to let the rider know that a part went bad or something else went wrong.

There are not many safety features on the Two Wheel Deal, and an on/off switch would make it safer. When powering up nowadays, it takes two people connect the wires and set the vehicle upright. Another safety feature would be a "Dead Man's Switch" This would be a switch, mechanical or electrical, that would tell the microcontroller if a rider fell off. This way the vehicle would shut off and not run over the user.

Next, a better handlebar shaft would be added. The handlebar shaft right now is flimsy which makes the "self balancing mode" oscillate some. A stiffer handlebar shaft would help with the "self balancing mode" as well give a user more control.

A more robust steering mechanism could be designed. Right now a little joystick controls the steering, but it could be made easier. Another sensor could be added to the handlebars to let the user keep both hands on the handlebars while steering.

Finally, the Two Wheel Deal could use more aesthetics. Paint or at plastic covering with designs would make the vehicle more pleasing to the eye.

12.0 Summary and Conclusions

The Two Wheel Deal was a huge learning experience. It was the largest project that our team has been a part of from start to finish. Picking out and ordering parts was a learning experience. This was a very open ended project. Our team had to agree on which parts we were going to need to accomplish our project. Also, we had to be conscious of how much the parts were going to cost because the money was coming from each of the team members.

Learning the PCB design software was a huge challenge that all of the team members faced. Cadence Layout took a long time to learn, but it is really useful software. Reading datasheets of suggestions on how to design the PCB was new to our team also.

Learning a new microcontroller was another experience. While many commands are similar, there are subtle differences that needed to be learned. Also learning initialization code took some time. Once the software was downloaded to the microcontroller, the debugging started. Our team learned that debugging code was much different than debugging the PCB. When something goes wrong, the problem could be in code as well as somewhere on the PCB. Usually, finding the problem is much harder than solving it.

Another accomplishment of this project was getting the Two Wheel Deal to balance itself as well as with a rider. Learning how to tune a PD controller took common sense along with the knowledge learned from the classroom. Our team learned that just guessing gain constants is a bad way to start tuning. A person should set the derivate gain to zero to see how the response of the proportional gain reacts. This is a way to uncouple both gains to determine how each affects the balance of the system.

The mechanical part of the Two Wheel Deal was also a big learning experience. Our team learned that there is a turnaround time for work to be done. At times, we were waiting on mechanical parts to be fabricated when that part of the project could have been done weeks ahead of time. Looking ahead and time management is a skill that everyone can improve on. Also, our team learned that aluminum is a soft metal. Our team tried to make our wheel hubs out of aluminum. After an hour of riding the Two Wheel Deal, the aluminum had sheared and the wheel hubs were useless. Eventually, the hubs were made out of steel. It is better to do something right the first time. Our team only lost a couple days of work, but it could have been worse. Finally, our group accomplished what we set out to do. We completed the Two Wheel Deal, and it is a good feeling to see the looks on people’s faces when they ride it.

13.0 References

List of References

[1] Segway, “The Science of Segway”, [Online Document], [cited 31 January 2008], http://www.segway.com/about-segway/science-of-segway.php

[2] BetaNews, “Segway Offers Financing to Push Sales”, [Online Document], May 2006, [cited 31 January 2008], http://www.betanews.com/article/1149106883

[3] Robotmarketplace, “NPC-T74”, [Online Document], [cited 31 January 2008], http://www.robotmarketplace.com/products/NPC-T74.html

[4] Atmel, “Atmel ATmega32 Description”, [Online Document], [cited 31 January 2008], http://www.atmel.com/dyn/resources/prod_documents/2503S.pdf

[5] Microchip, “PIC16F87XA Datasheet”, [Online Document], [cited 31 January 2008], http://ww1.microchip.com/downloads/en/DeviceDoc/39582b.pdf

[6] AVR Freaks, “AVR Freaks homepage”, [Online Document], [cited 31 January 2008], http://www.avrfreaks.net/

[7] Analog Devices, “Analog Devices ADXL103/ADXL203 Data Sheet”, [Online Document], [cited 31 January 2008], http://www.analog.com/UploadedFiles/Data_Sheets/ADXL103_203.pdf

[8] Melexis, “MLX90609-E2 Angular Rate Sensor Datasheet”, [Online Document], [cited 31 January 2008], http://www.melexis.com/prodfiles/0005095_MLX90609E2_web_datasheet.pdf

[9] Free Patents Online Patent: 6561294 Inventor: Dean L Kamen Date Filed: Aug 31, 1999. Available http://www.freepatentsonline.com/6561294.pdf

[10] Free Patents Online Patent: 5775452 Inventor: Steven J Patmont. Date Filed: Jan. 31, 1996. Available http://www.freepatentsonline.com/5775452.pdf

[11] US Patent and Trademark Office Patent: 2003-159546 Inventor: Gousuke Nishikawa Date Filed: June 1, 2004. Available http://appft1.uspto.gov/netacgi/nph-arser?Sect1=PTO1&Sect2=HITOFF&d=PG01&p=1&u=%2Fnetahtml%2FPTO%2Fsrchnum.html&r=1&f=G&l=50&s1=%2220060260862%22.PGNR.&OS=DN/20060260862&RS=DN/20060260862

[12] National Semiconductor, Linear Voltage Regulators, LM340T series Datasheet http://www.national.com/ds/LM/LM340.pdf

[13] Fairchild Semiconductor HRF3205 100A, 55V N-Channel MOSFET Datasheet http://www.tranzistoare.ro/datasheets3/fairchild/HRF3205.pdf

[14] Atmel ATmega32 Datasheet http://www.atmel.com/dyn/resources/prod_documents/doc2503.pdf

[15] Analog Devices ADXL203 ±1.7g Dual-Axis Accelerometer http://www.analog.com/UploadedFiles/Data_Sheets/ADXL103_203.pdf

[16] Melexis ±150 °/s MLX90609E2 Angular Rate Sensor (Gyroscope) http://www.melexis.com/prodfiles/0005359_MLX90609_standard_datasheet.pdf

[17] MIL-HDBK-217F Military Handbook—Reliability Prediction of Electronic Equipment http://cobweb.ecn.purdue.edu/~dsml/ece477/Homework/CommonRefs/Mil-Hdbk-217F.pdf

[18] Engineering at University of Michigan, “Modeling an Inverted Pendulum”. [Online]. Available: http://www.engin.umich.edu/group/ctm/examples/pend/invpen.html. [Accessed: Apr. 6, 2008].

[19] Institute of Electrical and Electronics Engineers, “IEEE Code of Ethics”. [Online]. Available: http://www.ieee.org/portal/pages/iportals/aboutus/ethics/code.html. [Accessed: Apr. 6, 2008].

[20] American International Group, “Printed Wiring Board Manufacturing”. [Online]. Available: http://www.aigenvironmental.com/environmental/public/envindustries/0,1340,63-11-333,00.html. [Accessed: Apr. 7, 2008].

[21] Virginia Department of Environmental Quality, “P2 in Printed Circuit Board Manufacturing”. [Online]. Available: http://www.p2pays.org/ref/10/09835.htm. [Accessed: Apr. 7, 2008].

[22] National Weights and Measure Laboratory, “RoHS”. [Online]. Available: http://www.rohs.gov.uk/. [Accessed: Apr. 7, 2008].

[23] U.S. Environmental Protection Agency, “Batteries”. [Online]. Available: http://www.epa.gov/garbage/battery.htm. [Accessed: Apr. 7, 2008].

[24] Segway, “The Science of Segway”, [Online Document], [cited 6 January 2008], http://www.segway.com/about-segway/science-of-segway.php

[25] Segway, “Segway i2 Specifications”,[Online Document],[cited 6 February 2008], http://www.segway.com/downloads/pdfs/i2-specs.pdf

[26] Wikipedia, “Segway PT”, [Online Article], [cited 6 February 2008], http://en.wikipedia.org/wiki/Segway

[27] Analog Devices, “Analog Devices ADXL103/ADXL203 Data Sheet”, [Online Document], [cited 9 February 2008], http://www.analog.com/UploadedFiles/Data_Sheets/ADXL103_203.pdf

[28] Melexis, “MLX90609-E2 Angular Rate Sensor Data Sheet”, [Online Document], [cited 9 February 2008], http://www.melexis.com/prodfiles/0005095_MLX90609E2_web_datasheet.pdf

[29] Fairchild Semiconductor, “HRF3205 Power MOSFET Data Sheet”, [Online Document], [cited 9 February 2008], http://www.fairchildsemi.com/ds/HR/HRF3205.pdf

[30] Intersil, “HIP4081A 80V High Frequency H-Bridge Driver”, [Online Document], [cited 9 February 2008], http://www.intersil.com/data/an/an9405.pdf

[31] Atmel, “ATmega32 Datasheet”, [Online Document], [cited 9 February 2008], http://www.atmel.com/dyn/resources/prod_documents/doc2503.pdf

[32] Atmel (2007) ATMega32 datasheet Available:

http://www.atmel.com/dyn/resources/prod_documents/2503S.pdf

[33] Melexis (2008) Angular Rate Sensor datasheet Available:

http://www.melexis.com/prodfiles/0005359_MLX90609_standard_datasheet.pdf

[34] Analog Devices (2008) Precision +- 1.7g Single-/Dual-Axis iMEMS Accelerometer

Available: http://www.analog.com/UploadedFiles/Data_Sheets/ADXL103_203.pdf

[35] Engineering at University of Michigan, “Modeling an Inverted Pendulum”. [Online]. Available: http://www.engin.umich.edu/group/ctm/examples/pend/invpen.html. [Accessed: Mar. 19, 2008].

[36] Atmel Corporation, “ATmega32/32L Microcontroller”. [Online]. Available: http://www.atmel.com/dyn/resources/prod_documents/doc2503.pdf. [Accessed: Mar. 19, 2008].

[37] Atmel Corporation, “AVR505: Migration between ATmega16/32 and Atmega164P/324P/644P”. [Online]. Available: http://www.atmel.com/dyn/resources/prod_documents/doc8001.pdf. [Accessed: Mar. 19, 2008].

[38] Trevor Blackwell, “Building a Balancing Scooter”. [Online]. Available: http://tlb.org/scooter.html. [Accessed: Mar. 19, 2008].

[39] Massachusetts Institute of Technology, “The DIY Segway”. [Online]. Available: http://web.mit.edu/first/segway/. [Accessed: Mar. 19, 2008].

Appendix A: Individual Contributions

A.1 Contributions of Pete Dudash:

For this project, I helped set up and organize our group's homepage and links for easy navigation. I also completed the design component homework about the Packaging Design and Specifications and the professional component homework about the Reliability and Safety Analysis of our project. In addition to those, I completed PowerPoint presentations for each of those homework assignments to present to the rest of the senior design students and staff. I helped design and draw the circuit schematic for our microcontroller board. Once the time came, I then helped layout the PCB from the motor controller schematic. However, the entire PCB layout for the motor controller boards was drastically changed thanks to Karl's advice about avoiding power rings. While we were waiting for the PCBs to return and after our microcontroller samples arrived, I began writing code for a heartbeat program to familiarize myself with the new AVR Studio software and the microcontroller. Once I had a good understanding of that, I initialized all the peripherals that we needed to use for this project. Those peripherals include the ADC conversion sequences, the timer module, pulse-width modulation, and some code to help us use the LCD screen more easily. Once we got those peripherals and the LCD screen working properly, I worked with the accelerometer and software in order to get the correct angle to show up on the LCD in degrees. I also participated in writing the balancing algorithm. Once the PCBs returned, I worked on populating the microcontroller board and testing the components at frequent intervals to eliminate problems as they occur. One major problem I discovered was that the traces connecting the accelerometer were incorrect. I had to cut some traces and fly-wire some other ones in order to get the component to operate properly. Some other minor fly-wiring was required to correct the ADC reference voltage. After we completed the PCBs, I worked on arranging the batteries, motors, and circuit boards under the base plate to maximize the space while keeping the system as mechanically stable/balanced as possible. I also designed the brackets out of aluminum to secure the batteries and PCBs in the appropriate places. For convenience and safety, I designed a relay circuit that switches the batteries between parallel for charging and series for operating along with a dead man's switch that changes between balancing a rider and balancing itself in case the rider falls off. After we fastened everything under the base plate, I helped tune the Proportional-Derivative controller and the steering constants by using potentiometers. Once everything was completed, I helped test the functionality and durability by spending a lot of time riding the Two Wheel Deal.

A.2 Contributions of Greg Eakins:

I provided the team with the initial (and at the time, crazy) idea of building a Segway-like vehicle. It took some time to convince the rest of the team that this was not impossible. Throughout the first month of the project, I did a huge amount of research into what it would take to make this project happen. I researched all of the parts necessary, such as the accelerometers, gyroscopic angular rate sensor, microcontroller, motors, and motor controllers. I did some of the basic calculations to determine the physical needs of the project. I then started requesting samples from many manufacturers (at least 3 different companies for the angular rate sensors alone), and placed Digikey, Newark, and PartsExpress orders to get some parts in hand as soon as possible so that the final decisions on the PCB and circuit could be known. Research into the motor controllers needed to control the high current motors revealed that there was not an affordable commercial product available, so I decided to build our own motor controllers. I did the research and design of the motor controllers, as well as the PCB population and hardware testing of the two motor controller boards. I designed the second revision of the motor controller PCB from scratch due to feedback from the course staff during the midterm design review. I participated in the design of the microcontroller board PCBs as well. I then did all of the final PCB output and submission.

As the team leader of this project, I ensured that all vital decisions were made with the coherency and success of the project in mind. I meticulously inspected the design several times, referencing data sheets carefully to determine if things would work correctly when implemented, and I tried to think up better ways of doing things. I made sure everyone was on the same page with regards to the design as a whole, as well as incremental design changes that might affect other parts of the project, and I set attainable project development goals for the group each week and made sure that we stayed on track and on schedule.

I developed the website from scratch, writing all of the CSS and figuring out all of the little features on it such as the random JavaScript quote on the front page, and the embedded Flash videos on the videos page. I also worked on the cross-browser compatibility. I did all of the graphics, photography, and video editing. The results of this can be seen on the Pictures and Videos pages of the website.

I contacted an MET student to help with the design and manufacture of the initial aluminum wheel hubs. I helped with the building and machining of several parts, including the aluminum hubs and the LCD box. I also took part in most of the chassis build up, and chassis repairs, making many trips to the hardware store.

I wrote some of the software such as the interrupt driven ATD algorithm, some of the functions for controlling the PWM outputs, and some of the balancing and steering algorithm. I also took part in all of the tuning and refining to the algorithms during the later weeks of the project.

A.3 Contributions of Eric Geier:

When class began, I began looking for possible donations for the motors which were the most important and most expensive component needed for the project. I was able to obtain a donator for the motors as well as the metal used in the main chassis frame. It was originally going to be aluminum but while discussing the idea with the donator, he advised using steel which would hold up better under the intense strain.

I helped determine the five Project Specific Success Criteria which determined exactly what the vehicle needed to accomplish as well as write the first and second homework assignments. These included the overall project description as well as the block diagram and PSSCs. I created the initial block diagram for the system. I also started continually updating my lab notebook at this time since the website was started. Then I helped design the overall chassis for the vehicle as well as including an LCD and using an ATMEL microcontroller. I picked up the metal and motors and determined where holes needed to be placed to mount the motors to the metal. Then I started researching how to model the motors and the vehicle to begin designing the controller and model. I started to determine parameters for the DC motor and develop a system model with Jeremy. I also dropped the steel bars to Chuck H. to be drilled and welded.

I helped to write Homework 4 and started to create the microcontroller schematic. I created a unique part in Orcad for the microcontroller. I completed most of the microcontroller schematic. Then I tried importing the schematic to the PCB software to start the microcontroller board PCB. I worked with Greg to continue the initial microcontroller PCB layout and do some changes to the motor controller PCB layout. I found some initial software to help get the microprocessor up and running and worked on the presentation for the Design Review.

Then I worked to get the microcontroller up and running using the starter software and setting up registers for the peripherals. This was done on a prototyping board that was created. I wrote functions and code to get the LCD working. This was done to allow debugging to be used on the LCD. I then helped determine what the registers needed to be to get the ATD and TIM peripherals up and running. I wrote a delay function to ensure that the main loop for the code was executed at a constant rate using interrupts. Then I worked with Pete on writing the steering and balancing algorithm. I also helped prototype the accelerometer.

I added more LCD functions that helped display information in a more software friendly way. I also created a custom battery character for the display and wrote the logic that would determine how much of the battery needed to display depending on the voltage left in the batteries. I completed Homework 9. I helped mount the vertical stick as well as obtain a box that would house the joystick, LCD, and handlebars. Pete and I made drawings for the battery holders and dropped off the metal to be tooled. Then I worked with the team in a small tool shop to drill many holes in the floor plate to mount the batteries and PCBs. Then assembled the vehicle and I tested the vehicle and it balance me as well as drove forward and backward.

Jeremy and I then worked on the joystick and managed to get it mounted to the LCD box. I helped solder headers on the cable running to the LCD and the joystick and was then connected to the microcontroller. The program was changed to include the steering and I tested the vehicle and it worked perfect. I also managed to find someone to make some new wheel hubs out of steel for free since the aluminum hubs were too weak to withstand the torque from the motors. I completed Homework 12, worked on the User Manual, and wrote the ECE Senior Design Report. I helped tune the constants for the PD controller which testing the vehicle. I also helped with the variety of testing that occurred on the Two Wheel Deal throughout the last couple of weeks of the semester. Finally I helped add LockTight to the bolts used throughout the hubs to ensure the wheels do not come loose after a mishap occurred after a weekend of riding.

A.4 Contributions of Jeremy Gries:

When the class began, I helped in setting up the webpage. I have been updating my web notebook and infrequently adding posts to the homepage. I helped write some of the papers that were done in this course. One paper included the first homework, and I also helped come up the initial PSSCs. I wrote homework six and ten, which were the PCB design layout and patent liability, respectively. Also, I created the PowerPoint slides that corresponded with the respective homework papers and presented them to the class during the TCSP sessions. When the PCB design started, I created the initial design of the PCB layout for the brainboard. The motor controllers design was later changed. I helped pick out some of the components that were used on the PCBs. I also helped pick out the motors. When the Atmel microcontroller came in, I helped write the heartbeat program that got some lights to blink. Additionally, I figured out how to initialize the ATD, PWM, and Timer peripherals. This process was very time consuming because nobody in our group had worked with Atmel microcontrollers before. I helped write other small subroutines that were used for the display. When the PCBs were received, I helped populate the board by soldering on some of the components. Also I helped debug the circuit and determine if all of the parts were working correctly. A few of the footprints were incorrect, so I helped determine how to rewire the components that needed to be. I was able to obtain materials that were needed for the construction of the Two Wheel Deal. I donated the steel handlebar shaft and the base plate used to stand on for the vehicle. Also I took the steel chassis to the tool room and explained how the metal was to be machined. I drilled holes and assembled the smaller parts that didn't need to be machined. I also brought in my tools so that our team could assemble the vehicle in the lab. I went to the store to purchase bolts and other various parts that were used on the Two Wheel Deal. Once the vehicle was assembled, I helped tune the proportional and derivative gains used in the microcontroller. I did a lot of research on the inverted pendulum problem, and the best way to tune the PD controller. In the beginning of the semester, I attempted to run simulations on the system using a Matlab program called Simulink. Eventually, this was done by using potentiometers, so that the microcontroller didn't need to be reprogrammed each time. I also helped in the testing of the Segway. When we were determining the maximum speed of the vehicle, I was the volunteer. In the final weeks of the project, I compiled all of the revised homework papers into the final report. Finally, I went back to make the revision to the sections based on the comments provided by the TA's.

Appendix B: Packaging

Figure B.1: Project Packaging Illustrations

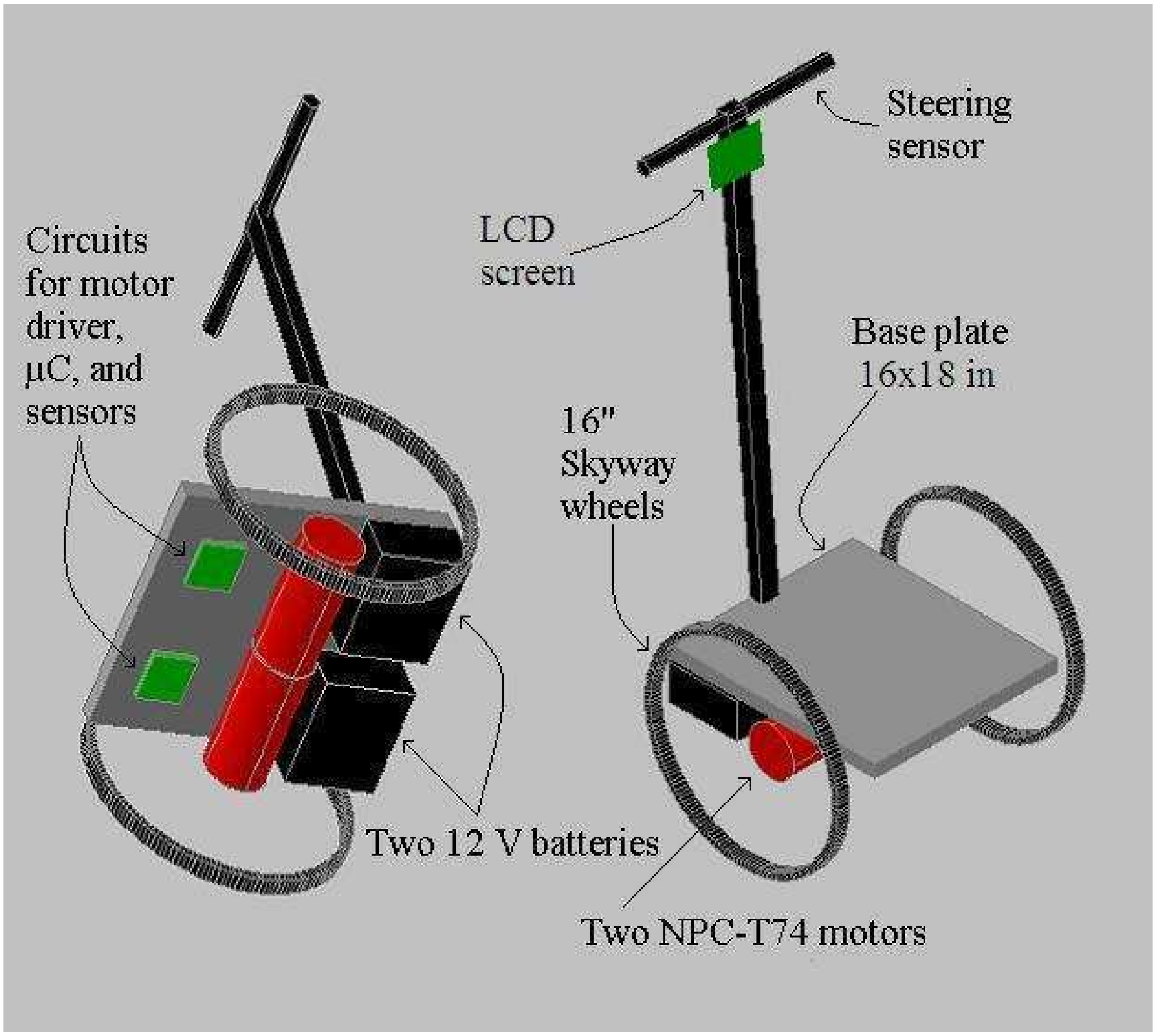


Figure B.2: PCB Footprint Layout

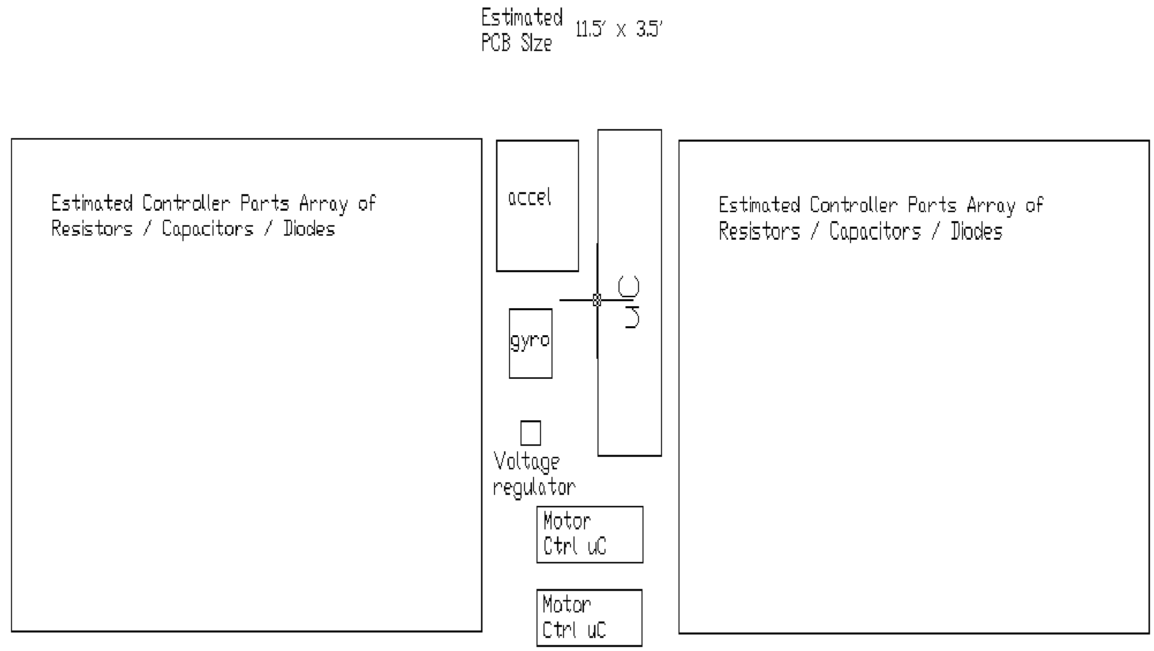
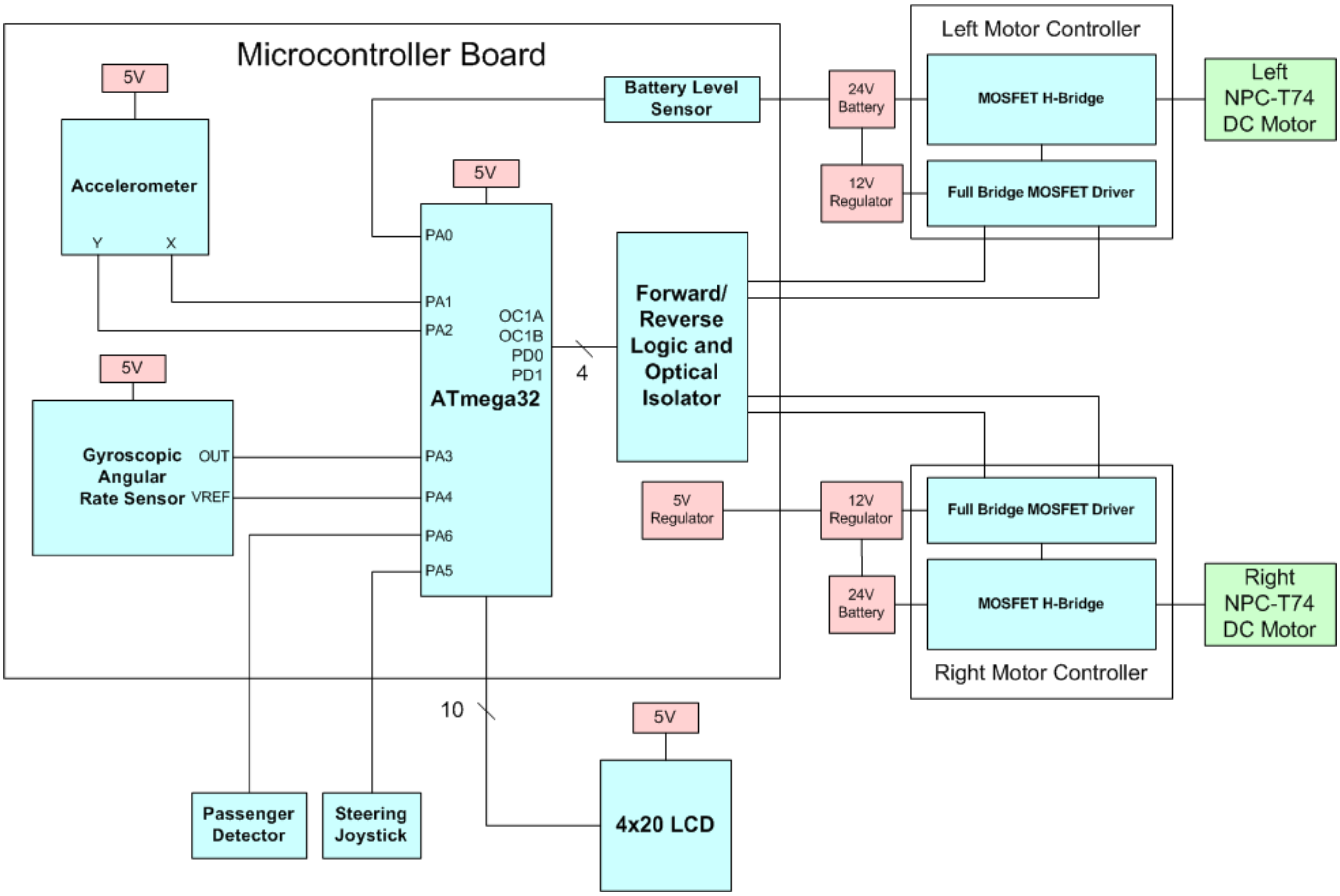

Estimated
PCB SIze
11.5' x 3.5'
Estimated Controller Parts Array of
Resistors / Capacitors / Diodes
accel
gyro
Voltage
regulator
Motor
Ctrl uC
Motor
Ctrl uC
Estimated Controller Parts Array of
Resistors / Capacitors / Diodes

Table B.3: Project Packaging Specifications

Item	Dimensions	Weight	Cost	Tooling Req.
Steel Frame	16 x 18 in	14 lbs	$50	Welding, drilling
Steel Footplate	15 x 17 in	9 lbs	$20	Cut to size
Steering Post	1 x 1 x 42 in	5 lbs	$10	Cut to size
Handle Bar	1 x 1 x 12 in	1 lb	$5	Cut to size
	Total	**29 lbs**	**$85**	

Appendix C: Schematic

Figure C.1: Final Block Diagram Schematic



Appendix D: PCB Layout Top and Bottom Copper

Figure D.1: Brainboard PCB Layout

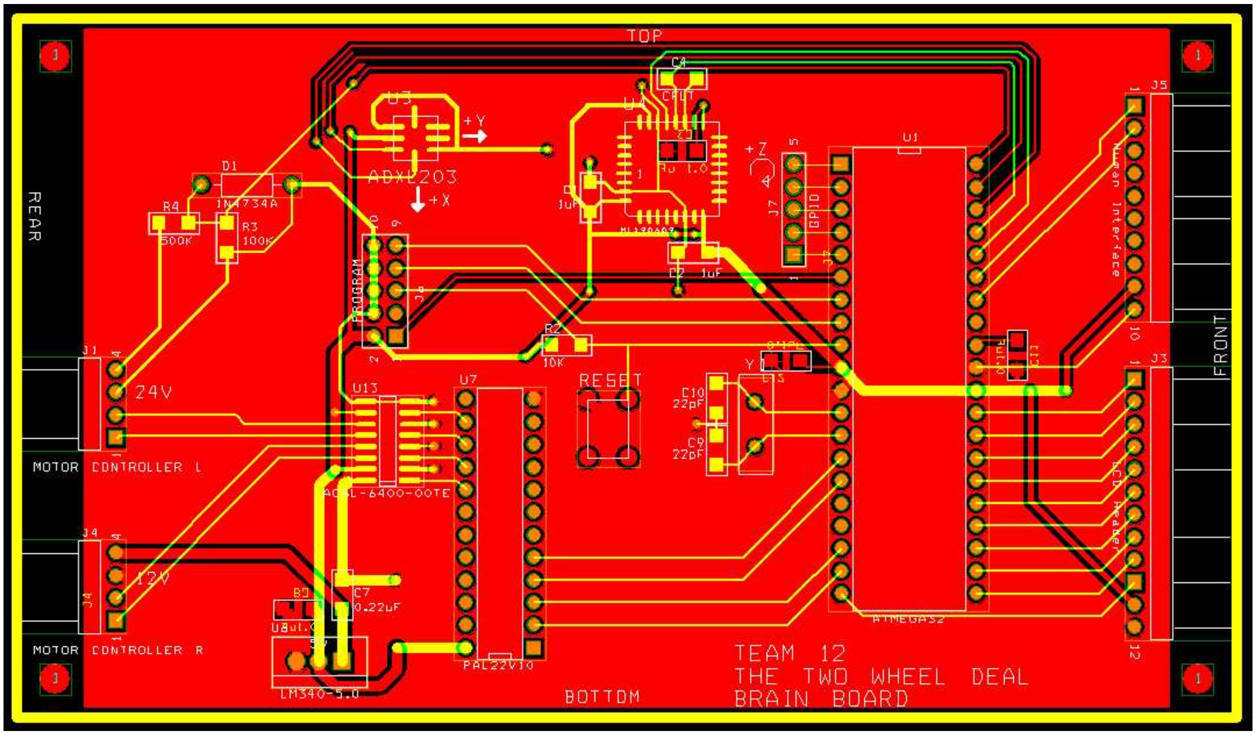
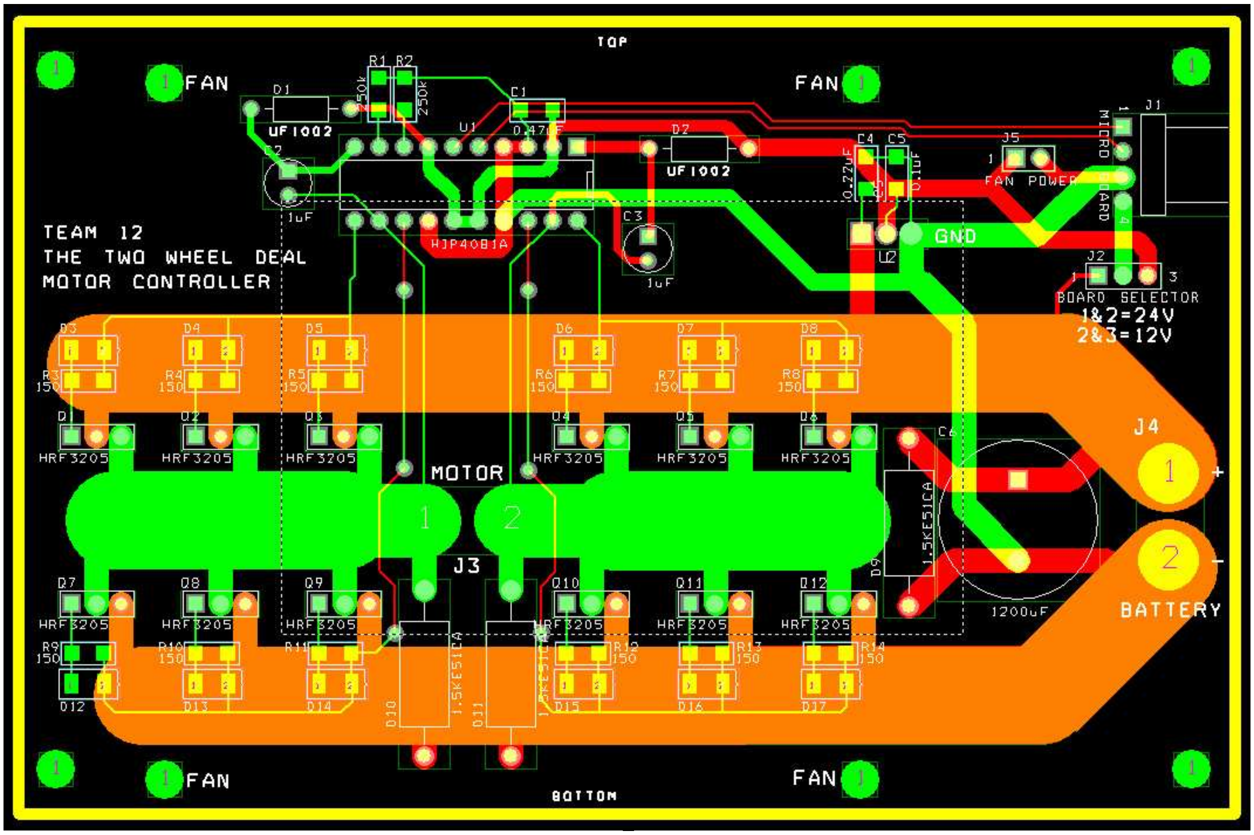


Figure D.1: Motor Controller PCB Layout



Appendix E: Parts List Spreadsheet

Vendor	Manufacturer	Part No.	Description	Unit Cost	Qty	Total Cost
Atmel	Atmel	ATmega32	8-bit microcontroller	$8.00	1	$8.00
NPC-T74	National Power Chair	NPC-T74	Wheelchair Motor	$324.00	2	$648.00
eBay	Generic	none	Sealed Lead Acid Battery	$44.99	2	$89.98
Melexis	Melexis	MLX90609-E2	Gyroscopic Angular Rate Sensor	$39.95	1	$39.95
Analog Devices	Analog Devices	ADXL203	Dual Axis Accelerometer	$12.00	1	$12.00
Futurlec	Futurlec	LCD20X4BL	20x4 LCD	$20.90	1	$20.90
Skyway Wheel	Skyway Wheel	WHL4015	20” Tuff Wheel	$19.10	2	$38.20
Digikey	Fairchild Semiconductor	HRF3205-ND	100A MOSFET	$1.46	24	$35.04
Intersil	Intersil	HIP4081AIP	Full bridge FET driver	$6.24	2	$12.48
Digikey	Diodes Incorporated	SD101AW-FDICT-ND	Schottky Diode	$0.31	24	$7.44
Digikey	National Semiconductor	LM2574HVN-12-ND	Switching Voltage Regulator	$4.85	1	$4.85
				TOTAL		**$916.84**

Appendix F: Software Listing

```
/*
Two Wheel Deal
Completed:
April 11, 2008

By:
Pete Dudash
Greg Eakins
Eric Geier
Jeremy Gries
*/

/*====================================================================================
This is the main code file used for this project. It is programmed for an ATMEL
ATmega32 8-bit microcontroller and programmed using AVR Studio.

This code includes functions to initialize the peripherals and timer interrupt
routines. It has a main polling loop that keeps track of time. At every 1, 10, 100,
and 1000 milliseconds, an interrupt is generated to complete its assigned task before
returning to the main polling loop. Due to this fact, the overall program is
considered a hybrid.

Every thousandth of a second, this program reads the ADC results from the parameters
and filters them to produce a more stable average for use in the balancing algorithm.

Every hundredth of a second, this program runs the balancing algorithm which computes
the tilt angle, steer demand, and then outputs the correct PWM signal to each motor.

Every tenth of a second, this program updates the LCD screen with the battery voltage,
tilt angle, and approximate speed. This interrupt also calculates the approximate
speed by using the PWM duty cycles, tilt angle, and steer demand.

Every second, this program updates the battery algorithm which filters out all the
spikes in voltage due to the switching of the motors and overcoming static inertia.
====================================================================================*/

#include <avr/io.h>
#include <avr/interrupt.h>
#include <math.h>
#include <stdio.h>
#include <string.h>

//Define functions
//======================
void ioinit(void);                                 //Initializes IO
void perinit(void);                                //Initialize LCD,ADC,PWM,TIM
void varinit(void);                                //Initialize variables
void print_greeting(void);                         //Prints welcome screen
void delay_ms(int);                                //General purpose delay
void lcd_rc(int,int);                              //Locates cursor position on LCD
void lcdcch(int,int,int,int,int,int,int,int,int);  //Creates Custom LCD character
void lcdaddr(int);                                 //Writes command to LCD Input HEX
void lcdrite(int);                                 //Writes data to LCD INPUT HEX
void lcdstr(char *);                               //Writes string to LCD
void lcdnum(float);                                //Writes number to LCD
void balance_alg(void);                            //Derives motor PWM
void lcd_update(void);                             //Updates LCD
void battery_alg(void);                            //Filters battery readings
void parameter_filter(void);                       //Filters acceleration and rate
void left_motor(float);           //Used to update Left Motor PWM and direction
void right_motor(float);          //Used to update Right Motor PWM and direction
void calculate_speed();           //Calculates current vehicle speed
//======================

//Global Constants
//======================
#define KP 0.75           // proportional
#define KD 1.20           // derivative
#define KS 5.25           // steering
#define KV 0.4            // Steering Velocity Compensation
#define PI 3.14159265     // constant for pi

//Define Variables
//======================
int thou, tens, huns, ones, tcnt, ocnt, hcnt,delflag,time = 0;              //Timer
Variables tracking 10ths,100ths
int planb = 1;                                              //Software Dead Man Switch

//Balance Algorithm variables
float acx_adc, acy_adc, gyr_adc, gyr_adc_ref, pot_adc, pot_KP = 0, pot_KD = 0;

//Algorithm ADC values
float gyro, accx, accy;
float steer_demand, steer_zero;
float vehicle_speed;
float tilt_angle = 0, rest_angle;                  //Vehicle Angles
signed int turn, l_motor, r_motor;
unsigned int adc_raw[8], currentADCChannel;

//Filtering variables
unsigned int batteryfilter[60], bfindex;
unsigned int accxfilter[120], afxindex;
unsigned int accyfilter[120], afyindex;
unsigned int gyrofilter[120], gfindex;
unsigned int steerfilter[240], sfindex;
float battery_filtered;
float accx_filtered;
float accy_filtered;
float gyro_filtered;
float steer_filtered;
float lmotor_torque, rmotor_torque; //contains duty cycle of PWM output
int frequency;
int balance_mode;

int main(void)
{
        lmotor_torque = 0;
        rmotor_torque = 0;
        vehicle_speed = 0;
        rest_angle = -4;
        frequency = 1000;
        sei();                                     //Enables Interrupts
        ioinit();                                  //Setup IO pins and defaults
        perinit();                                 //Initialize Peripherals
        print_greeting();                          //Prints Greeting
        lcdaddr(0x01);                             //Clears LCD
        varinit();                                 //Initialize variables
        left_motor(lmotor_torque);
        right_motor(rmotor_torque);

        while(planb)
        {
                if (adc_raw[6] > 128)  //checks PB0 by masking rest of bits
                {
                        rest_angle = -4; //balance rider
                }
                else
                {
                        rest_angle = -16.25;       //balance itself
                }
                if (thou == 1)
                {
                        parameter_filter();
                        thou = 0;
                }
                if(huns == 1)                      //Checks if 10 ms passed
                {
                        balance_alg();
                        huns = 0;
                }
                if(tens == 1)                      //Checks if 100 ms passed
                {
                        lcd_update();
                        calculate_speed();
                        tens = 0;
                }
                if (ones == 1)
                {
                        battery_alg();
                        ones = 0;
                }
        }
        return(0);
}

// *********************************************************
//
//  TIMER INTERRUPT SERVICE ROUTINE
//      Occurs every 1ms
//
// *********************************************************
SIGNAL(SIG_OUTPUT_COMPARE0)
{
        delflag = 1;                               //Sets 1 ms flag
        hcnt++;
        tcnt++;
        ocnt++;
        thou = 1;
        if (hcnt == 10)                          //Checks if 10 ms passed
        {
                huns = 1;
                hcnt = 0;
        }
        if (tcnt == 100)                         //Checks if 100 ms passed
        {
                tens = 1;
                tcnt = 0;
        }
        if (ocnt == 1000)
        {
                ones = 1;
                ocnt = 0;
                time += 1;
        }
}

// ***************************************************************************
// *
// *      A D C _ I S R
// *
// ***************************************************************************
//
// ADC interrupt service routine
// with auto input scanning
ISR(ADC_vect)
{
//  Read the AD conversion result and store in ATD array.
//  adc_raw[0] = Battery Meter
//  adc_raw[1] = Accelerometer X
//  adc_raw[2] = Accelerometer Y
//  adc_raw[3] = Gyroscope Output
//  adc_raw[4] = Gyroscope Reference
//  adc_raw[5] = Human Interface 1 - Steering
//  adc_raw[6] = KS Pot
//  adc_raw[7] = KV Pot
        adc_raw[currentADCChannel]=ADCH;

// Select next ADC input
        if (++currentADCChannel >= 8)
                currentADCChannel = 0;

   ADMUX = 0x20+currentADCChannel;

// Start the AD conversion
   ADCSRA |= 0x40;
}
```

```
// ********************************************************
//
//  User Defined Functions
//
// ********************************************************
void balance_alg(void)
{
        gyr_adc_ref = adc_raw[4];

        //Subtract Off Offsets
        accx = accx_filtered - 128;
        accy = accy_filtered - 128;
        gyro = gyro_filtered - gyr_adc_ref;

        //Calculate Tilt Angle in Degrees
        tilt_angle = atan(accy/accx)*180/PI;     //+ = forward, - = backward


        //PD Algorithm
        lmotor_torque -= KP*(tilt_angle-rest_angle) + KD*gyro;
        rmotor_torque += KP*(tilt_angle-rest_angle) + KD*gyro;

        //Steering Algorithm
        //Scale the Right Turn magnitude
        if (steer_filtered < steer_zero)
                steer_filtered = steer_zero - 0.76*(steer_zero - steer_filtered);

        steer_demand = KS*(steer_filtered - steer_zero) / (KV*vehicle_speed+1);

        if ((steer_demand < 50) && (steer_demand > 0))
                steer_demand = 0;
        else if ((steer_demand > -50) && (steer_demand < 0))
                steer_demand = 0;
        else if (steer_demand >= 50)
                steer_demand -=50;
        else if (steer_demand <= -50)
                steer_demand += 50;

        //Angle Limit
        if ((tilt_angle > 45) || (tilt_angle < -45))
        {
                lmotor_torque = 0;
                rmotor_torque = 0;
                steer_demand = 0;
        }

        // Speed Limit
        if (lmotor_torque > 1000) lmotor_torque = 1000;
        if (lmotor_torque < -1000) lmotor_torque = -1000;
        if (rmotor_torque > 1000) rmotor_torque = 1000;
        if (rmotor_torque < -1000) rmotor_torque = -1000;

        //Motor Controller PWMs
        left_motor(lmotor_torque + steer_demand);
        right_motor(rmotor_torque + steer_demand);

}
```

```
void left_motor(float duty)
{
        if (duty > 0)
                PORTD &= 0xFE;
        else if (duty < 0)
        {
                PORTD |= 0x01;
                duty *= -1;
        }
        OCR1B = duty;
}

void right_motor(float duty)
{
        if (duty > 0)
                PORTD &= 0xFD;


        else if (duty < 0)
        {
                PORTD |= 0x02;
                duty *= -1;
        }
        OCR1A = duty;
}

void lcd_update(void)
{
        int tblock, bblock, tempty, bempty;

        //Determines number of full or empty battery blocks needed
        if(battery_filtered >= 22)
        {
                tblock = bblock = 4;
                tempty = bempty = 0;
        }
        else if(battery_filtered >= 16)
        {
                tblock = bblock = 3;
                tempty = bempty = 1;
        }
        else if(battery_filtered >= 10)
        {
                tblock = bblock = 2;
                tempty = bempty = 2;
        }
        else if(battery_filtered >= 4)
        {
                tblock = bblock = 1;
                tempty = bempty = 3;
        }
        else
        {
                tblock = bblock = 0;
                tempty = bempty = 4;
        }
        //Display String and Symbol
        lcd_rc(1,1);
        lcdstr("Speed:");
        lcd_rc(2,1);
        lcdnum(vehicle_speed);
        lcd_rc(2,5);
        lcdstr(" mph");
        lcd_rc(3,1);
        lcdstr("Angle:");
        lcd_rc(1,12);
        lcdstr(" Battery:");
        lcd_rc(4,1);
        lcdnum(tilt_angle);
        lcdrite(0xDF);
        lcdstr(" ");
        lcd_rc(2,15);
        lcdnum(battery_filtered);
        lcd_rc(2,19);
        lcdstr(" V");
        lcd_rc(3,15);
        lcdrite(0x03);                  //Displays left (top) side battery block

        while(tblock != 0)
        {
                lcdrite(0x00);          //Displays full (top) battery block
                tblock--;
        }
        while(tempty != 0)
        {
                lcdrite(0x02);          //Displays empty (top) battery block
                tempty--;
        }
        lcdrite(0x05);                  //Displays right (top) side battery block
        lcd_rc(4,15);
        lcdrite(0x04);                  //Displays left (bot) side battery block
        while(bblock != 0)
        {
                lcdrite(0x01);          //Displays full (bot) battery block
                bblock--;
        }
        while(bempty != 0)
        {
                lcdrite(0x5F);          //Displays empty (bot) battery block
                bempty--;
        }
        lcdrite(0x06);                  //Displays right (bot) side battery block
}
```

```
void battery_alg(void)
{
        int batterytotal = 0;
        int i;
        int battery_filter_time = 60;
        //  Samples = Window that Battery is being averaged over in sec.
        batteryfilter[bfindex++] = adc_raw[0];
        if (bfindex > battery_filter_time)
                bfindex = 0;

        for(i = 0; i < battery_filter_time ; i++)
                batterytotal += batteryfilter[i];

        //Convert value to volts and average for filter
        battery_filtered = (float) batterytotal/battery_filter_time/8.4;
        // 8.4 is to scale ADC output to 30V
}

void parameter_filter(void) {
        int acceleration_x_total = 0;
        int acceleration_y_total = 0;
        int gyro_total = 0;
        int steer_total = 0;
        int i;
        int acceleration_x_filter_time = 20;//120
        int acceleration_y_filter_time = 20;//120
        int gyro_filter_time = 10;
        int steer_filter_time = 120;

        accxfilter[afxindex++] = adc_raw[1];
        accyfilter[afyindex++] = adc_raw[2];
        gyrofilter[gfindex++]  = adc_raw[3];
        steerfilter[sfindex++] = adc_raw[5];

        if (afxindex > acceleration_x_filter_time)
                afxindex = 0;
        if (afyindex > acceleration_y_filter_time)
                afyindex = 0;
        if (gfindex > gyro_filter_time)
                gfindex = 0;
        if (sfindex > steer_filter_time)
                sfindex = 0;

        for (i = 0; i < acceleration_x_filter_time; i++)
                acceleration_x_total += accxfilter[i];
        for (i = 0; i < acceleration_y_filter_time; i++)
                acceleration_y_total += accyfilter[i];
        for (i = 0; i < gyro_filter_time; i++)
                gyro_total += gyrofilter[i];
        for (i = 0; i < steer_filter_time; i++)

        steer_total += steerfilter[i];
        accx_filtered = acceleration_x_total/acceleration_x_filter_time;
        accy_filtered = acceleration_y_total/acceleration_y_filter_time;
        gyro_filtered = gyro_total/gyro_filter_time;
        steer_filtered = steer_total/steer_filter_time;

}
```

```
void calculate_speed() {
        vehicle_speed = fabs((fabs(lmotor_torque + steer_demand) + fabs(rmotor_torque
        + steer_demand) - fabs(KP*tilt_angle))/132);
}

void ioinit(void)
{
    //1 = output, 0 = input
    DDRA = 0b00000000; //Pin 7 output -> PA7 = RS
    DDRC = 0b11111111; //All outputs -> Data byte
    DDRD = 0b11111111; //Pin 6&7 output -> PD6 = RS, PD7 = E
}

void varinit (void)
{
        int i;

        for (i = 0; i < 60; i++) {
                batteryfilter[i] = 204;
        }
        for (i = 0; i < 240; i++) {
                steerfilter[i] = adc_raw[5];
        }

        parameter_filter();
        steer_zero = steer_filtered;
}

void lcd_rc(int row, int col)
{
        int address;

        if (row==1)
        {address = 0x80 + col - 1;}       //addresses for the first row start at 0x80
        if (row==2)
        {address = 0xC0 + col - 1;}
        if (row==3)
        {address = 0x94 + col - 1;}
        if (row==4)
        {address = 0xD4 + col - 1;}
        lcdaddr(address);
}

void lcdcch(int a,int b,int c,int d,int e,int f,int g,int h,int addr)
{
        int base[8] = {a,b,c,d,e,f,g,h};

        addr = addr*8+64;
        lcdaddr(addr);
        for (addr = 0; addr < 8; addr++)
                lcdrite(base[addr]);
}
```

```
void lcdaddr(int hexaddr)
{
        PORTC = hexaddr;                  //Store command in PORTC
        PORTD &= ~(_BV(PD6));             //Clear RS
        delay_ms(1);                      //Delay
        PORTD |= _BV(PD7);                //Set E
        delay_ms(1);                      //Delay
        PORTD &= ~(_BV(PD7));             //Clear E
        PORTD |= _BV(PD6);                //Set RS
}

void lcdrite(int hexchar)
{
        PORTC = hexchar;                  //Store character in PORTC
        delay_ms(1);                      //Delay
        PORTD |= _BV(PD7);                //Set E
        delay_ms(1);                      //Delay
        PORTD &= ~(_BV(PD7));             //Clear E
}

void lcdstr(char string[15])
{
        char *strptr;
        strptr = string;
        while (*strptr != '\0')
        {
                lcdrite(*strptr);
                strptr++;
        }
}


void lcdnum(float num)
{
        int full;
        int part;
        char string[20];
        char *numptr;

        numptr = string;
        full = (int) (num*100);
        part = (int) num;
        sprintf(string,"%d",part);

        while (*numptr != '\0')
        {
                lcdrite(*numptr);
                numptr++;
        }
        part *= 100;
        full -= part;

        if (full < 0)
                full *= -1;
        lcdrite(0x2E);
        numptr = string;
        sprintf(string,"%d",full);
        if (full < 10)
                lcdrite(0x30);

        while (*numptr != '\0')
        {
                lcdrite(*numptr);
                numptr++;
        }
}

void delay_ms(int x)
{
        while (x)
        {
                if (delflag == 1)
                {
                        x--;
                        delflag = 0;
                }
        }
}

void perinit(void)                      //Initializes Peripherals
{
        //Initialize TIM
        TCCR0 = 0x0B;
        TCNT0 = 0x00;
        OCR0 = 0xFA;
        TIMSK |= 0x02;
        TIMSK &= 0xFE;

        //Initialize PWM
        TCCR1A = 0b10100010;    //B7-B4 Clear OC1A/B on Compare Match
                                //B3,B2 Sets PWM Mode B1,B0 WGM 11/10

        TCCR1B = 0b00011001;    //B7-B5 Don't mess with
                                //B4,B3 WGM 13/12 B2-B0 Prescaler
        ICR1 = frequency;       // 132 clock cycles for 15 KHz Frequency
                                //Duty cycle(OCR1) = OCR1/ICR1

                                //Initialize ADC

        currentADCChannel = 0;
        ADCSRA = 0x87;          //Enable ADC and set diviser to 128
        ADCSRA |= (1 << ADIE);  //Enable ADC interrupts
        ADMUX = 0x20;           //B7,B6AREF, B5Left Justified, B6,B5 Don't change
                                //B2-B0 Channel Select last 3 bits
        ADCSRA |= (1<<ADSC);    //Start conversions

        //Initialize LCD
        PORTD = !_BV(PD7);              //Sets E low initially
        PORTD = _BV(PD6);               //Sets RS high initially

        delay_ms(40);                   //Delay
        lcdaddr(0x38);                  // Init
        delay_ms(5);
        lcdaddr(0x0C);                  // Init
        delay_ms(5);
        lcdaddr(0x06);                  // Init
        delay_ms(5);
        lcdaddr(0x01);                  // Init LCD Clear Command

        lcdcch(0,31,31,31,31,31,31,31,0);//Full Battery Block (top)
        lcdcch(31,31,31,31,31,31,31,0,1);//Full Battery Block (bot)
        lcdcch(0,31,0,0,0,0,0,0,2);     //Empty Battery (top)
        lcdcch(0,1,1,1,1,1,1,1,3);      //Left Side Battery (top)
        lcdcch(1,1,1,1,1,1,1,0,4);      //Left Side Battery (bot)
        lcdcch(0,16,16,16,16,28,4,4,5);  //Right Side Battery (top)
        lcdcch(4,4,28,16,16,16,16,0,6);  //Right Side Battery (bot)
}

void print_greeting(void)
{
        lcdaddr(0x01);                  // LCD Clear
        lcd_rc(1,4);
        lcdstr("Two Wheel Deal");
        lcd_rc(3,1);
        lcdstr("Pete D.");
        lcd_rc(4,1);
        lcdstr("Greg E.");
        lcd_rc(3,12);
        lcdstr("Eric G.");
        lcd_rc(4,12);
        lcdstr("Jeremy G.");
        delay_ms(2000);
}
```

Appendix G: FMECA Worksheet

Failure No.	Failure Mode	Possible Causes	Failure Effects	Method of Detection	Criticality	Remarks
A1	Cannot send PWM and direction signals to motor controller	PLD malfunction, optical isolator damaged	Wheels will not spin	Observation by oscilloscope	High	High Criticality because this could cause harm to both rider and device.
B1	Micro remains in reset mode	Reset switch is broken and stays in “pressed” state.	Microcontroller fails to run program, also cannot reprogram FLASH memory	Observation with DMM	Low	Low criticality because user will realize the system won’t balance after power up so user isn’t in danger of falling off.
B2	V_{CC} shorted to ground	C_{12} fails and shorts out	No power to microcontroller—cannot drive motors	Observation with DMM	High	High criticality because if the micro is no longer powered, then the wheels stop and it could throw rider and damage the device.
B3	ADC inputs are all zero	C_{11} fails and shorts out	AREF for ADC channels is 0V causing all ADC outputs to be zero	Observation with DMM	High	If the micro no longer receives vital system feedback, then system becomes unstable which may injure rider and damage the device.
C1	System performance seems random and erratic	Bypass capacitors C_4, C_5, C_7, C_8 became disconnected or failed	Introduces substantial amounts of noise which affects sensor outputs	Observation while riding	High	Medium criticality because system will still operate. If this isn’t fixed right away, personal injury and device damage could occur.
C2	V_{CC} = 0V	Bypass capacitors C_4, C_5, C_7, C_8 became shorted or failed	No power to microcontroller stops the motors from turning.	Observation by DMM measurement	High	

Failure No.	Failure Mode	Possible Causes	Failure Effects	Method of Detection	Criticality	Remarks
D1	No response to changes in tilt	Either one or both axis of the accelerometer failed	System will not respond accurately, balancing alg. wont respond to small changes	Observation while riding	High	Depending on which axis fails, system may still balance. Could result in instability, injury, or damage to device.
D2	No response to quick changes in tilt	Failure of gyroscope, filter capacitor is disconnected or shorted out	System will respond the same way for quick and slow movements which may disrupt balancing	Observation while riding	High	The system should still balance itself but response may be slow. If rider changes angle quickly, they will fall off. Damage or injury may result.
D3	Incorrect battery voltage reading	Failure of zener protection diode, or resistors R_3 or R_4 burned up	The battery strength displayed on LCD is wrong	Observation of resistors on PCB or while riding	Low	Low criticality because system will still operate as expected. This failure won't harm user or device, but only be an inconvenience.
E1	Cannot control motors	The MOSFETs could have overheated and burned up	One or both motors stop working	Observation	High	High criticality because if the motor stops while the user is riding, injury and damage to device will occur.
E2	Operation is bumpy or unstable	Gate resistor and/or diode network may be destroyed	Shoot-through current may burn out components, rapidly discharge batteries.	Observation	High	High Criticality because the rider can fall off and get hurt and batteries/electronics could be destroyed.

Failure No.	**Failure Mode**	**Possible Causes**	**Failure Effects**	**Method of Detection**	**Criticality**	**Remarks**
E3	+24 voltage line is grounded	Reservoir capacitor, C_6, becomes shorted or spike suppression diodes D_9, D_{10}, or D_{11} become shorted.	Rapid battery discharge. Power wires heat up and melt, batteries begin to leak.	Observation	High	High criticality because of possible injury or damage to device by leaking batteries or hot wires.
F1	+12 shorted to ground	Bypass capacitor, $C_{1,}$ shorts out.	H-bridge driver has no voltage and therefore cannot drive motors. Failure propagates to brain board too.	Observation	High	High criticality because the motors will not turn. This can cause injury to the user and possible damage to device.
F2	Cannot drive motors	High-pass capacitors C_2, C_3 are open circuited.	The PWM signal cannot pass into H-Bridge	Observation	High	High criticality because the motors may suddenly stop, injuring user and damaging device.
F3	Noisy or bumpy response	Pull down resistors R_1 and R_2 fail and are open circuited.	May get shoot-through current because of no dead-time between direction changes. Batteries may short, heat up, or leak.	Observation	High	Highly critical because if these fail, the device may suddenly stop operating; injury to rider or damage to batteries or device possible.

Figure G.1: Brainboard Functionality Block

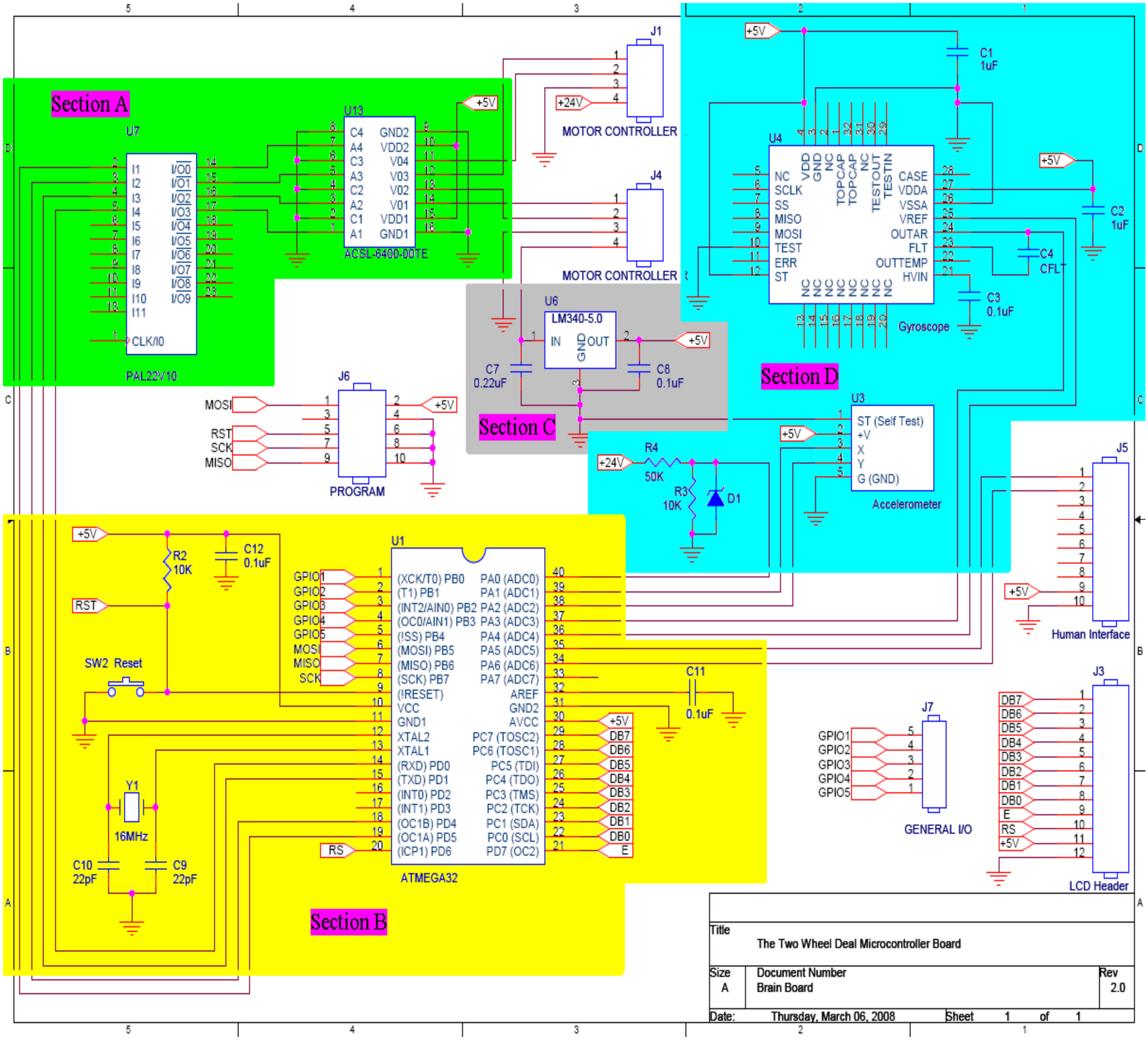
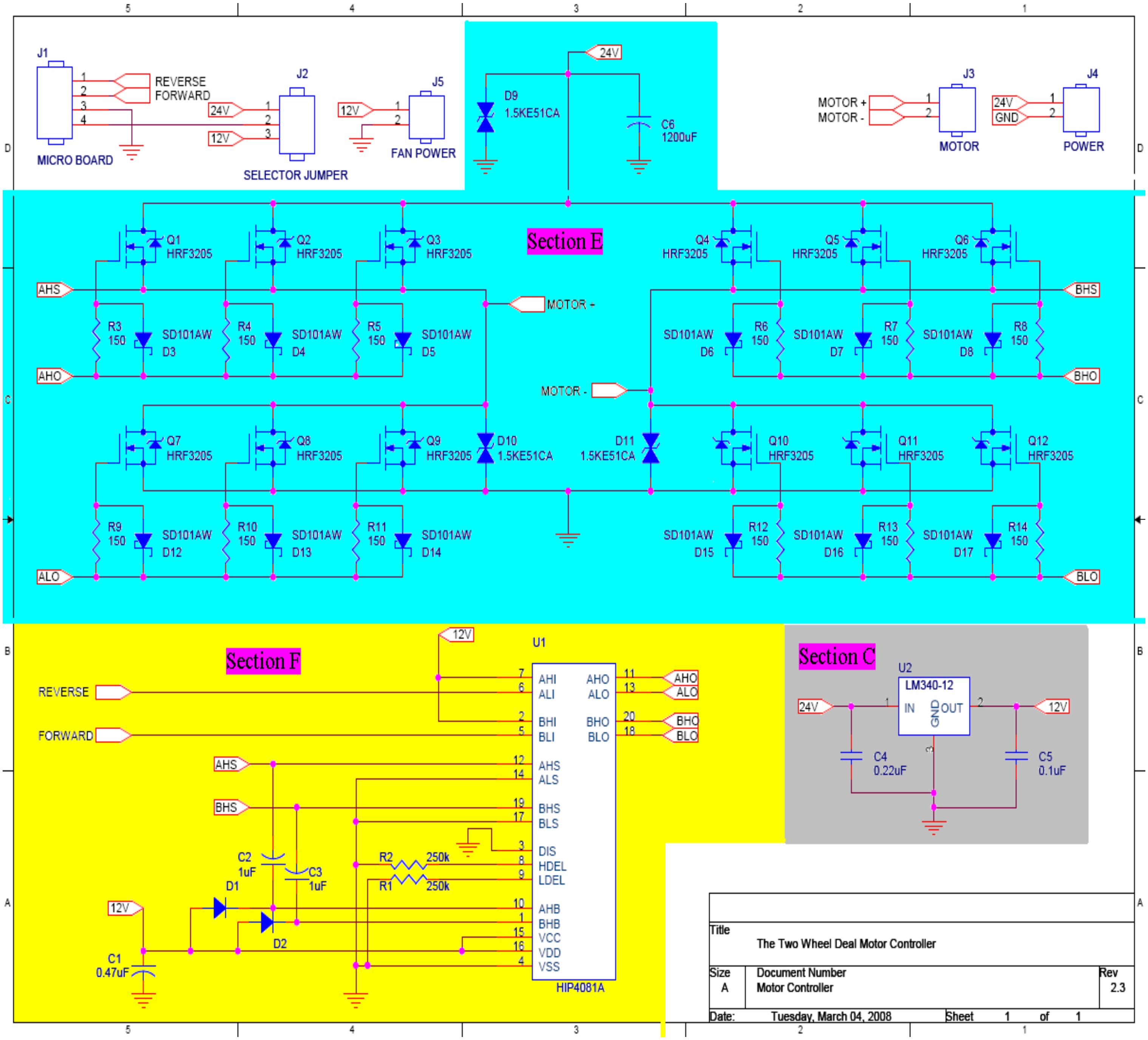


Figure G.2: Motor Controller Functionality Block



5
4
3
2
1
J1
1
2
3
4
REVERSE
FORWARD
MICRO BOARD
J2
24V
12V
1
2
3
SELECTOR JUMPER
J5
12V
1
2
FAN POWER
24V
D9
1.5KE51CA
C6
1200uF
MOTOR +
MOTOR -
J3
1
2
MOTOR
J4
24V
GND
1
2
POWER
D
Section E
Q1
HRF3205
Q2
HRF3205
Q3
HRF3205
Q4
HRF3205
Q5
HRF3205
Q6
HRF3205
AHS
BHS
MOTOR +
R3
150
SD101AW
D3
R4
150
SD101AW
D4
R5
150
SD101AW
D5
SD101AW
D6
R6
150
SD101AW
D7
R7
150
SD101AW
D8
R8
150
AHO
BHO
MOTOR -
C
Q7
HRF3205
Q8
HRF3205
Q9
HRF3205
D10
1.5KE51CA
D11
1.5KE51CA
Q10
HRF3205
Q11
HRF3205
Q12
HRF3205
R9
150
SD101AW
D12
R10
150
SD101AW
D13
R11
150
SD101AW
D14
SD101AW
D15
R12
150
SD101AW
D16
R13
150
SD101AW
D17
R14
150
ALO
BLO
B
Section F
12V
U1
REVERSE
FORWARD
AHS
BHS
7
6
2
5
12
14
19
17
3
8
9
10
1
15
16
4
AHI
ALI
BHI
BLI
AHS
ALS
BHS
BLS
DIS
HDEL
LDEL
AHB
BHB
VCC
VDD
VSS
AHO
ALO
BHO
BLO
11
13
20
18
AHO
ALO
BHO
BLO
HIP4081A
C2
1uF
C3
1uF
R2
250k
R1
250k
D1
D2
12V
C1
0.47uF
Section C
U2
LM340-12
24V
1
IN
GND
OUT
2
3
12V
C4
0.22uF
C5
0.1uF
A
Title
The Two Wheel Deal Motor Controller
Size
A
Document Number
Motor Controller
Rev
2.3
Date:
Tuesday, March 04, 2008
Sheet
1
of
1