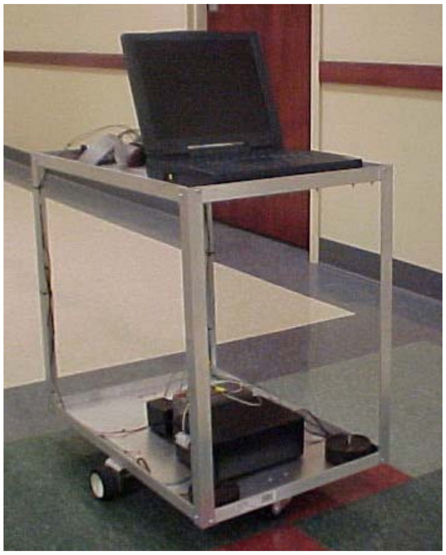
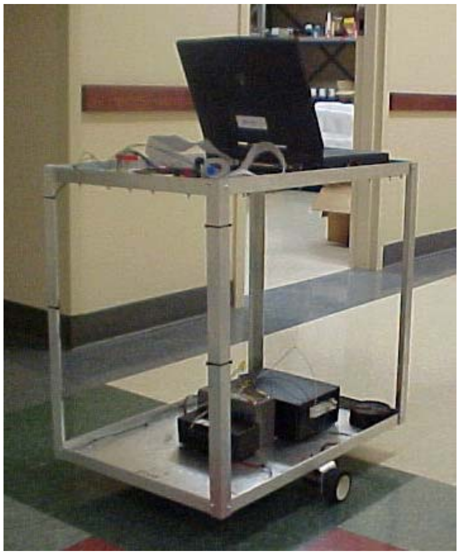
Adventures in Robotics

Benjamin Good
Information Science & Technology and Physics Departments
Doane College
bgood@doane.edu





Abstract

Robotics is a very interesting, diverse, and complex field. It is an ever growing field of research and innovation. This paper discusses the development and construction of a simple robot named Rovie to serve as an exploration into the basic concepts of robotics and a platform for future research projects at Doane College.

Introduction

To fulfill graduation requirements I designed and built a robot. My robot will serve as a foundation for future projects in robotics at Doane College. This paper will outline the process I followed. It will cover many design considerations I made such as chassis design, motor control, power, and overall control of the robot. Also it will discuss what I had to do to take the robot from the drawing board to real life, including designing and developing the circuit boards to designing and developing a Java software interface. Finally, it will briefly talk about the future uses of my robot.

Purpose

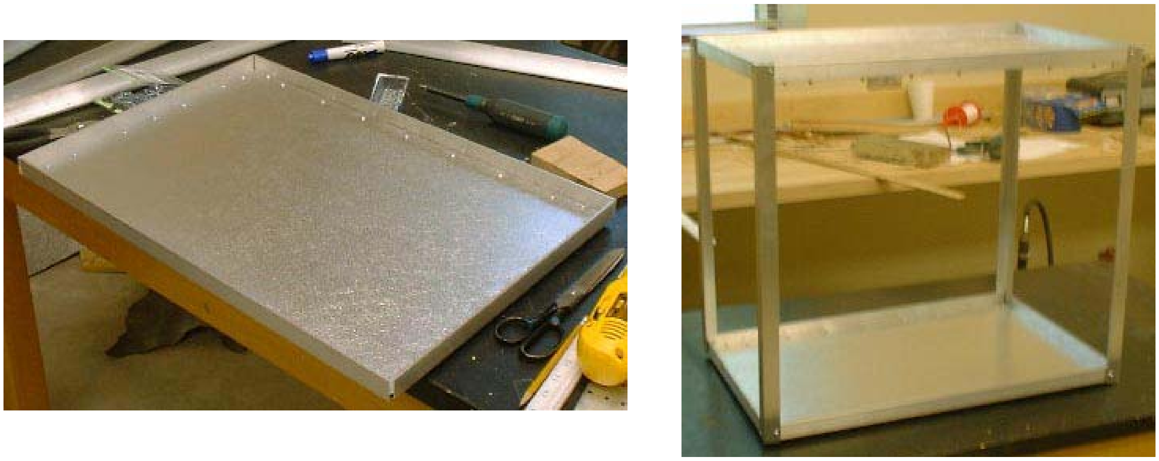
In the 21st century, we are seeing robots in many forms and robots that have a variety of functions. As humankind continues to progress robots will become more important and more versatile. Today, there are robots mowing lawns, vacuuming floors and people in car factories work side by side them. In the future, robots will be our personal chauffeurs, cook our meals, and do a multitude of other things for us.

Currently, the field of robotics is as diverse as the possibilities for robots. In short, the field of robotics is absolutely fascinating and diverse. For as long as I can remember, I have been fascinated with how things work. In particular I have been interested in computers and robotics. As a result, I have always been striving to increase my knowledge in these fields. To further my knowledge of robotics, I built a simple robot.

The robot I designed and built provides Doane College and me the foundation for further development in robotics. I built a simple two-tiered robot with basic functionality including movement (forward, reverse, right and left turns). The robot is controlled through a laptop computer that rides along and communicates through the parallel port.

Design

When creating a robot there are a number of design considerations that must be taken into account for it to be a success. The first design issue that I considered was the chassis. Weight and center of gravity are the two key areas. To keep the weight of the chassis minimal, my robot's chassis is constructed from aluminum stock, with the exception of steel brackets for the joints. Ideally, the chassis of the robot would also have a low center of gravity, so the robot will be as stable as possible. To maintain a low center of gravity, the chassis must be short to allow for as much flexibility as possible. I designed and built my robot's chassis to be 22" tall. I cut platform each from 1" L-shaped aluminum. Each platform dimensions of 24" long by 16" wide. Textured sheet aluminum finishes out each platform by forming the bottom which I fastened to the frame by a screw and nut every 3". Figure 1 shows a finished top or bottom tray and the finished chassis. The schematic for the chassis is shown in Appendix A.

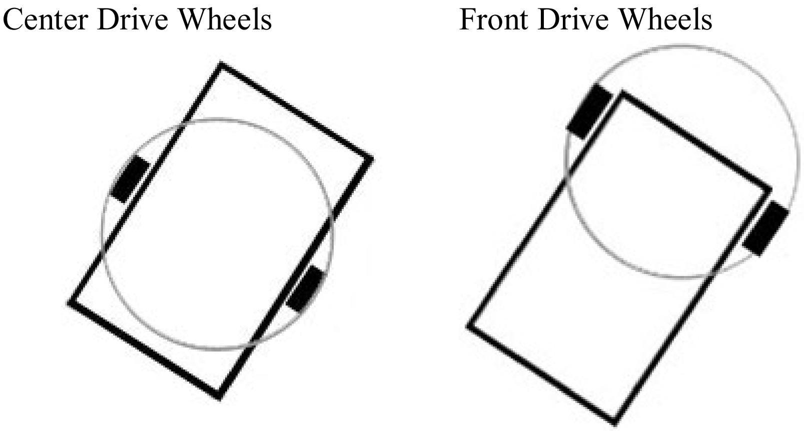


Completed top or bottom tray

Finished chassis

Figure 1: Chassis construction

The next design consideration was locomotion and motor placement. The robot uses two driver wheels placed in the center of the chassis with a single caster wheel at the front and back to provide stability. Placing the drive wheels in the center is done to keep the turning radius next to zero. The other option is to place the two drive wheels at the front and the two support casters at the rear. Placing the drive wheels at the front extends the turning radius in front of the chassis. This may possibly present problems in the future with automation. Figure 2 illustrates he differences in the turning radii.



Center Drive Wheels

Front Drive Wheels

Figure 2: Comparison of Turning Radii

Achieving forward and reverse is simple since it only requires turning the two wheels the same direction at the same time. Turning is accomplished by turning one motor off, so it acts like a pivot point, thus the additional requirement of independent motor control. The drive motors must also be capable of moving the chassis and associated control components. I chose two HNGH12-1623Y motors, which provide a maximum of 11.80 Kg-cm torque. Finally, I chose foam rubber wheels, since they provide the most traction.

Another design consideration is motor control. To control the direction the motor turns, I decided to use two H-bridges. An H-bridge is a set of two pairs of transistors with each pair arranged diagonally. When one pair is turned on current will flow across the motor, but when the other pair is turned on, current will flow the other direction causing the motor to turn the other way Figure 3 illustrates this.

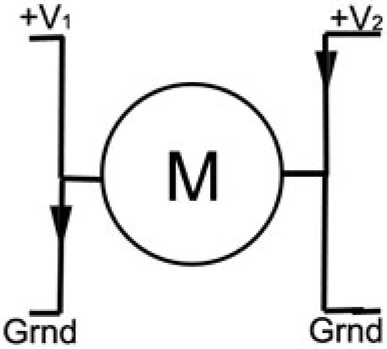


Figure 3: Conceptual workings of an H-Bridge.

Originally, I had planned to use a design for the H-bridges that came from The Robot Builder's Bonanza on page 260, but after I had built them and done some testing, I found out that the H-bridges would only supply a maximum of 5 volts (which is equal to the input or logic potential) to the motors, not the desired 12 volts.

In an attempt to supply my motors with maximum potential, I found a new design for my H-bridges. I decided on a design by 4QD-Tec (Torrens, 2000). The new design uses half-darlington pairs (an NPN and PNP transistor connected together) as replacements for the pairs of single NPN transistors and requires that one of the inputs be set low and the other input allowed to float. In initial testing, these H-bridges supplied the motors with the desired 12 volts. However, for an unknown reason, once the H-bridges had been built, they stopped supplying 12 volt and only supplied approximately 7 volts. Also, these new H-bridges required that I redesign my interface circuit to accommodate the new input values. However, once the H-bridges and interface circuitry were connected, the H-bridges stopped working. After troubleshooting the combination, it was determined that the TTL logic integrated circuits in the interface circuit were keeping the floating H-bridge input from floating up to approximately 12 volts. So, once again, it was back to the drawing board. I then decided to use two TC4244's which are pre-built H-bridges put into an 8-pin DIP package.

The next design consideration was how to control the robot. In the design requirements, it was stated that the robot would be controlled by a laptop, and communicate through the parallel port. I decided to use the circuit laid out in Chapter 30 of The Robot Builder's Bonanza. In this design, all of the bits output through the parallel port go straight through to the H-bridges. However, when the H-bridges changed the input types also changed. This required me to change the parallel interface circuit so that I would be able to send one H-bridge input low and have the other float up to 12 volts. However, as I later found out, the tri-state buffers used in the interface circuit were not allowing the floating inputs to float to 12 volts, but only to 5 volts, thus causing the motors not to work. Since the motors would not work with the 4QD-Tec H-bridges I had to once again change H-bridges which require a logical 1 or 0 as input values. To accommodate this, the interface circuit was changed back to the original design. Finally, I powered my interface circuit with a constant +5 volts, and my motors with +12 volts from the same power source.

To solve this power issue, I decided to get a 12V battery model TR7-12 with a 7 AMP hour capacity. Since the control circuitry requires 5 volts I could either add some extra batteries or use some sort of voltage regulator. Since weight is a major consideration in the design of any robot adding more batteries is not desirable. Thus I chose to use a linear voltage regulator. The voltage regulator is built around an LM7805 voltage regulator IC. The design I used is available in The Robot Builder's Bonanza on page 204.

Implementation

After designing my robot, affectionately called Rovie, it was apparent that I would have to call upon many skills as well as learn new skills. One of the new skills I acquired was designing and building circuit boards. All of the circuit boards were designed using Eagle Layout editor. This is a computer-based editor which is free for academic purposes and can be obtained from the manufacturer's website (www.cadsoft.de). Eagle was very easy to learn and use since it has a large built-in part library and will auto-route the traces after you have placed the necessary components. After all of the components and traces had been placed, I had to then build the circuit board. Through the course of the project, a number of different techniques for building circuit boards were tried.

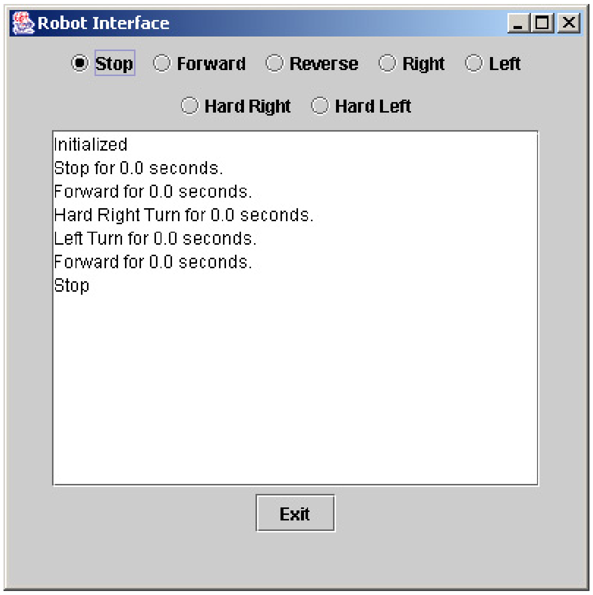
The first technique I used was done with Toner Transfer System transfer paper. The circuit is printed on the transfer paper using a laser printer. Next, the blank circuit board and transfer paper are placed on a hot plate with four 1-kg masses placed on top to provide pressure. The board and paper are then heated to about 375° F so that the toner melts and sticks to the copper. The board and toner are placed in ferric chloride or PCB etchant for about 15 – 20 minutes. This etchant will eat away all of the exposed copper leaving only the copper that was covered by the toner. Holes are then drilled at the appropriate locations for the components and the traces are then "tinned" or coated with solder so it is easier to attach the components and keep the copper traces from corroding. This process does, however, have some limitations due to the restrictions placed on the size of the traces. Due to the width restrictions routing can be inefficient at times. Also, the reproducibility of the technique is a problem due to inaccuracies in pressure and temperature while trying to re-melt the toner. In an attempt to remedy the situation, a laminator was used so the pressure and temperature would be more even, but a temperature high enough to melt the toner could not be achieved.

The second technique and the one I used, has been using PNP Blue transfer paper and Tin-It tinning solution. The circuits are designed the same way but printed using a laser printer onto what looks like a transparency coated with blue chalk. The transfer paper is then placed on blank circuit board and heated with a regular clothing iron to about 275° F and medium pressure is applied for about 3 minutes. Since a clothing iron is used, the pressure is very easy to control and the temperature is fairly steady. The board is then etched in a similar manner, but placed in the Tin-It solution heated to 140° F for about 15 minutes. This solution leaves the copper traces coated with a solder-like material. This process has been very reliable, and allows for finer traces so I can make better use of the entire circuit board.

Throughout the project, I had to develop the skill of troubleshooting and logically thinking though the progression of signals. In addition to these new skills I had to call upon some existing skills. During construction of the chassis, I needed to use some craftsmanship and ingenuity to make the different pieces and components come together correctly.

Rovie's "brain" is a laptop computer with a Java interface. Thus, I had to recall and hone my Java programming skills. As with all projects there were some problems here. I initially tried to use the Java Communication API, but was unable to send out any data through the parallel port. After doing some research online I was able to find a Java class that directly accesses the parallel port. Information on the parallel port class can be found at www.geocities.com/Juanga69/parport. Using the new parallel port class, I was able to successfully send data to the interface circuit. LEDs included in the circuit were used for diagnostic purposes.

The control program for Rovie uses two classes, namely the Robot class and the RobotInterface class, both of which I developed. The RobotInterface class is quite simple. It contains seven radio buttons labeled with the appropriate movement, so someone can tell Rovie what to do. There is a timer that refreshes every 1/10th of a second and checks the radio buttons and calls the appropriate function from the Robot class. Finally, there is a text area where all previous movements of that session and the length of each movement are recorded. Figure 4 is a screen shot of the interface program.



● Stop ○ Forward ○ Reverse ○ Right ○ Left

○ Hard Right ○ Hard Left

```
Initialized
Stop for 0.0 seconds.
Forward for 0.0 seconds.
Hard Right Turn for 0.0 seconds.
Left Turn for 0.0 seconds.
Forward for 0.0 seconds.
Stop
```

Figure 4: Screen shot of the interface program.

The Robot class is simply the group of functions that send out the hexadecimal representation of the associated movement. This is done in conjunction with the previously mentioned parallel port class.

Future Development:

Rovie's prospects are extremely bright. Even as I continue to work on him, there is another student who is developing a web-based vision and control system. People will be able to browse to a website, see what Rovie sees, and then command him to move forward, backward, or turn right or left. Work will continue on the Rovie's vision system, which would enable Rovie to comprehend the items he sees and then make decisions based on the images. This work will be done by another student as part of a future project.

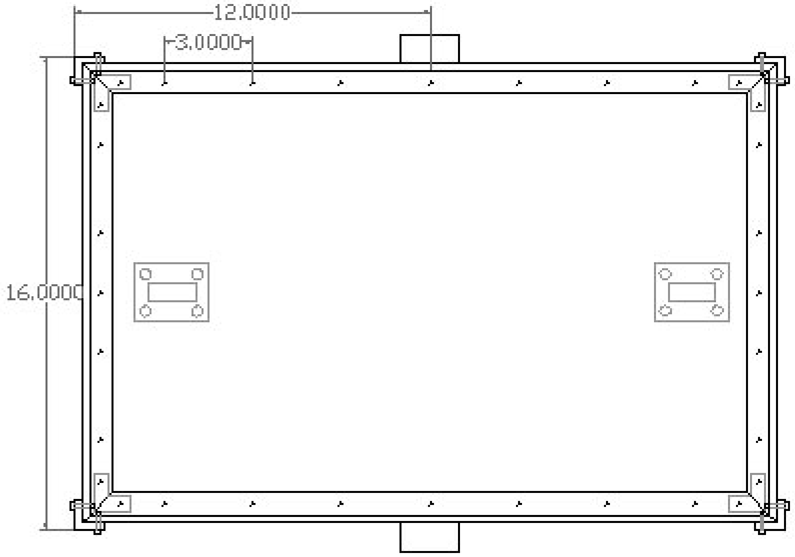
The addition of a vision system and the development of a program to take advantage of the vision system demonstrate the interest and enthusiasm surrounding Rovie. In the future I foresee replacing the laptop computer with a microcontroller and adding object recognition. These are just a couple of ideas. The possibilities are endless.

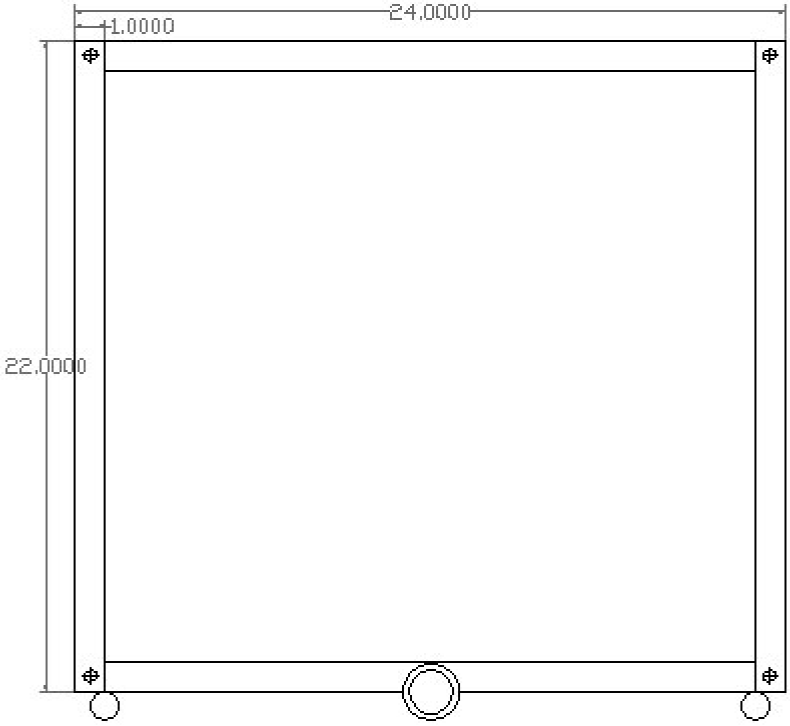
References

McComb, G. (2000). *The Robot Builder's Bonanza 2nd Edition*. New York. McGrawHill.

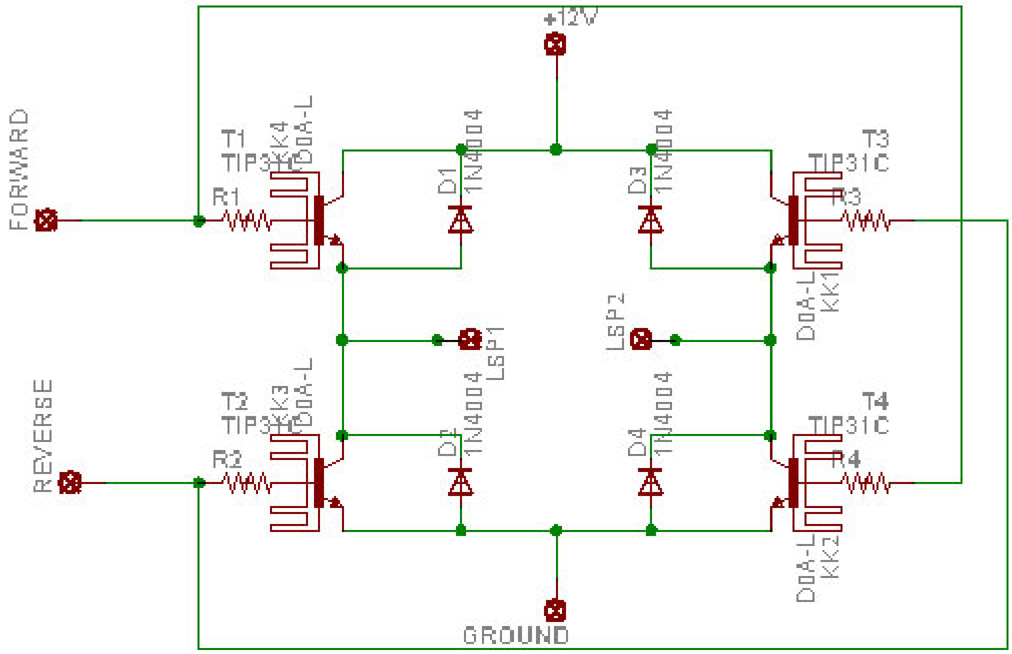
Torrens, R. (2000). *4QD-TEC:H bridge switch for small motors.* Retrieved August 1, 2002 from www.4qdtec.com/h.html

Appendix A - Chassis

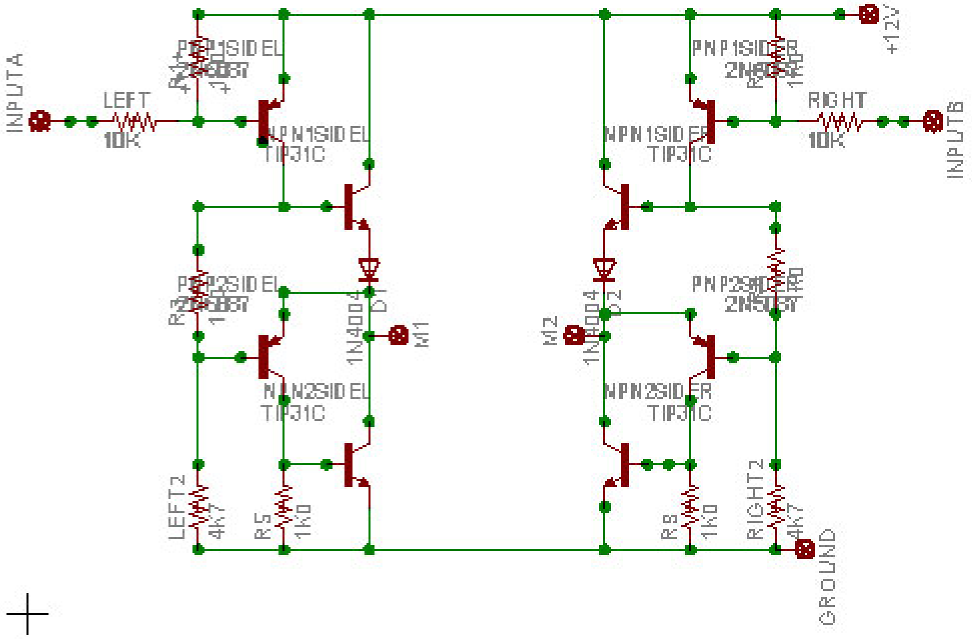




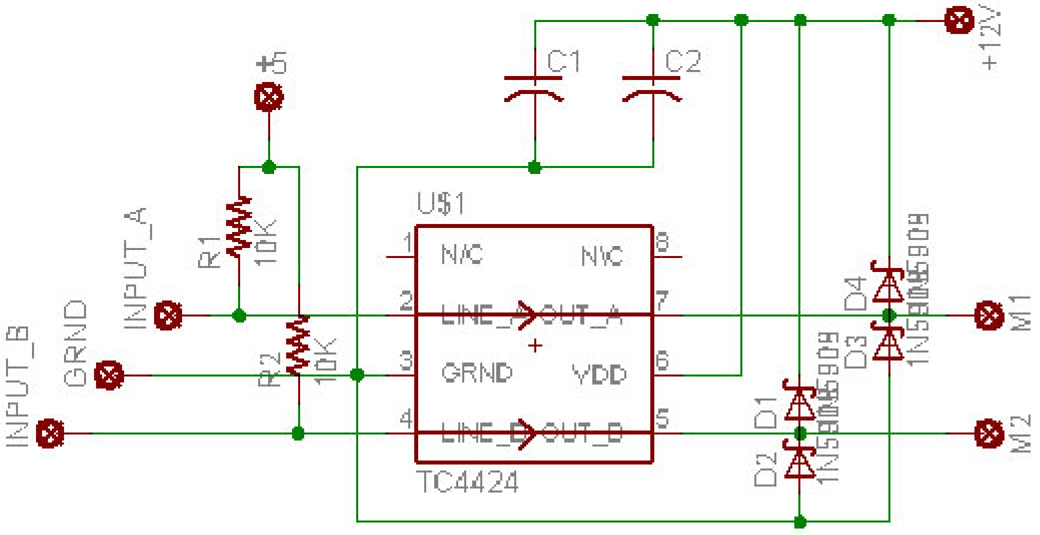
Appendix B - Circuit Schematics



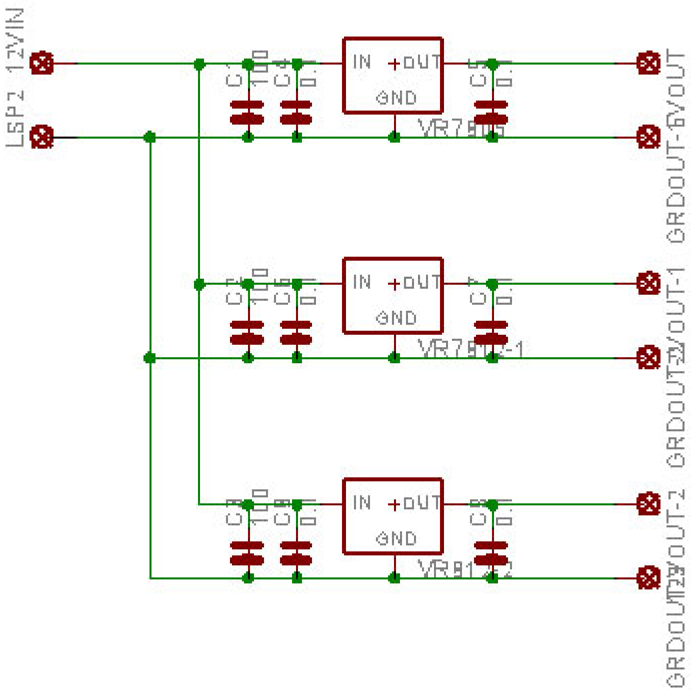
Schematic 1: H-bridge #1 from Robot Builder's Bonanza



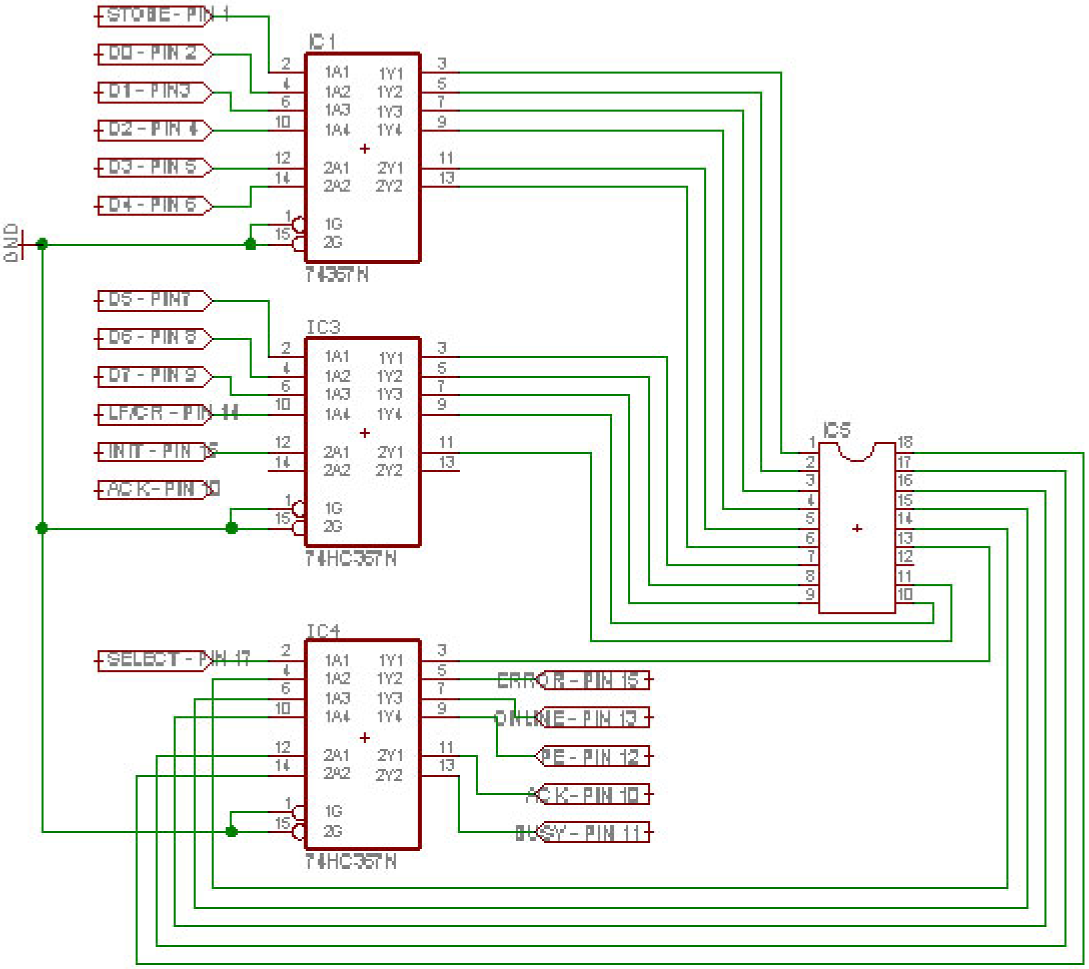
Schematic 2: H-bridge #2 from 4QD-TEC



Schematic 3: H-bridge #3 using TC-4424



Schematic 4: Voltage regulator from <u>Robot Builder's Bonanza</u>



Schematic 5: Interface circuit #1 from Robot Builder's Bonanza