

**EEL 5666**  
**Intelligent Machines Design Laboratory**  
**Fall 2003**

**NumBot**  
**The Number Reading Robot**

Final Report  
12/9/03

Max Billingsley  
Participant, REU 2003

## **Table of Contents**

<b>1. Abstract</b> .....	<b>3</b>
<b>2. Executive Summary</b> .....	<b>4</b>
<b>3. Introduction</b> .....	<b>5</b>
<b>4. Integrated System</b> .....	<b>6</b>
<b>5. Mobile Platform</b> .....	<b>6</b>
<b>6. Actuation</b> .....	<b>8</b>
<b>7. Sensors</b> .....	<b>10</b>
<b>8. Electronics</b> .....	<b>11</b>
<b>9. Behaviors</b> .....	<b>12</b>
<b>10. Experimental Layout and Results</b> .....	<b>14</b>
<b>11. Conclusion</b> .....	<b>14</b>
<b>12. References</b> .....	<b>16</b>
<b>13. Appendix</b> .....	<b>17</b>

## **1. Abstract**

This document outlines the development of my individual robot during the Summer 2003 Research Experience for Undergraduates program and for the Intelligent Machines Design Laboratory course at the University of Florida. The robot, which has gone from inception to completion over the course of the summer, is designed to locate blocks with numbers on them, position itself with a block, grab an image of the block using a camera, determine the number on the block using image processing software, and finally respond based on that determination.

## 2. Executive Summary

NumBot is an autonomous mobile agent designed to determine the number displayed on a block using an image-processing system. NumBot is also capable of avoiding obstacles in its path while operating and can follow a dark line on the floor.

In order to accomplish these tasks, NumBot uses a number of mechanical and electronic components and executes programs written in C. The basic components comprising NumBot are the base platform, the microcontroller, the image-processing computer, and the sensors. These are all integrated to form the full NumBot system to accomplish the intended behaviors.

The development of NumBot was an incremental process over the course of several months. From the inception of the idea, sketches were made to begin work on the platform. Research into image-processing techniques was also performed at an early stage. Once initial decisions were made about the electronics to be used, work began to write software for each of the needed functions.

Based on the initial platform ideas, CAD work progressed until the design was ready to be fabricated out of wood. The sensors and microcontroller were then added and software written to form a robot capable of avoiding obstacles autonomously. The next step was to integrate the image-processing computer and camera on the platform. Finally, software was developed to implement the behaviors for which NumBot was designed.

Because of the scope of this project, not all of the original goals were perfectly met. But the results have not been disappointing, and the whole process of developing NumBot has been worthwhile and rewarding.

### 3. Introduction

My robot design has grown out of plans to build a robot that manipulates children's blocks. The specific idea has emerged from the realization that many block are imprinted or painted with numbers intended to teach children. The goal was to be able to design a robot that can successfully identify the numbers imprinted on the blocks. Some inspiration for my design is taken from Cyrus Harrison's *Ranzor* from Summer 2002, which used image processing to read the face of a die.

My interest in my robot idea came from my desire to learn about image processing and apply it to a real-world problem. As the idea has been fleshed out over the course of the project, many challenges have been revealed. But as I have tackled these challenges, I have enjoyed the design process and am happy with the results I was able to achieve.

## **4. Integrated System**

The robot system as a whole is set up to allow for the successful completion of each stage of the robot's functionality. These stages include locating a block and positioning the robot with the block (while avoiding obstacles), grabbing an image of the block using the camera, and responding based on the determination of the value on the block.

NumBot has an on-board image processing system. This system consists of a Mini-ITX computer and a USB camera mounted on the robot platform. Because of the size, weight, and power needs of the robot's image processing components, the integrated system must accommodate this subsystem in particular.

The integrated system, consisting of the platform, actuation system, sensor array, image processing computer, and microcontroller, comprises a mobile, autonomous system for reading the number on a block and responding to this information.

## **5. Mobile Platform**

Though my original plan was for NumBot to have a small round platform, the decision to implement on-board image processing introduced needs that could not be met by such a simple design. The NumBot platform is designed to support the size and weight of the on-board computer system needed for image processing as well as the large 12V, 7Ah battery needed to supply sufficient power.

### ***5.1 Platform Design***

The platform base was designed using AutoCAD; it has two levels providing space for various components. The layout was based on the decision to mount the image processing computer board vertically. A simple block capture area is integrated into the front of the platform base. This (simple) design and other aspects of the platform were implemented after initial experimentation using cardboard. The platform measures 14 inches in length and is 8 inches wide. Figure 1 shows an AutoCAD rendering of the design.

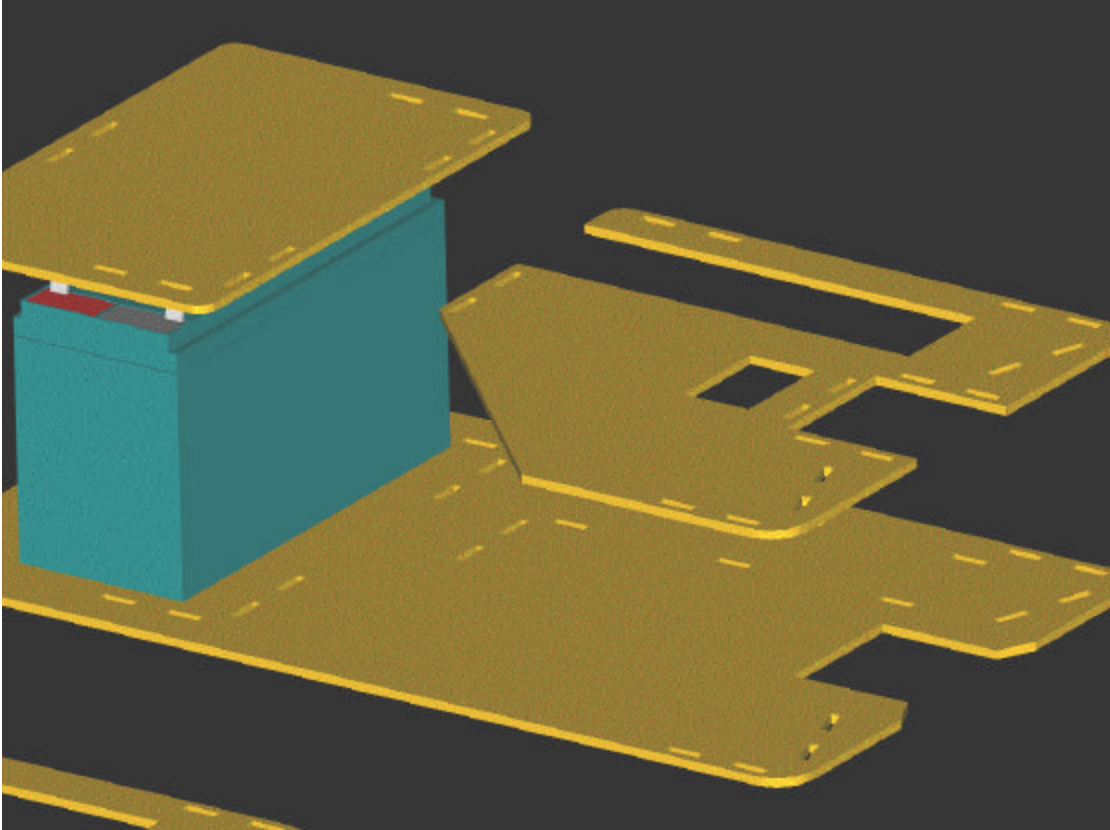


Figure 1

### ***5.2 Platform Implementation***

The platform was implemented from the AutoCAD design. The T-Tech machine was used to cut parts for the base from a sheet of 1/8-inch birch plywood. In order to mount the wheels and support the battery, the platform integrates an aluminum motor mount and battery support. Thanks go to Jim in the chemical engineering machine shop who helped with the fabrication of the aluminum motor mount.

I am happy that the initial platform implementation proved sufficient for NumBot. Because of the challenge of predicting the requirements of the platform early in the design phase, I was worried that oversights and omissions might render the platform useless. Fortunately those worries were unwarranted. Figure 2 shows the finished platform before electronic components were added.

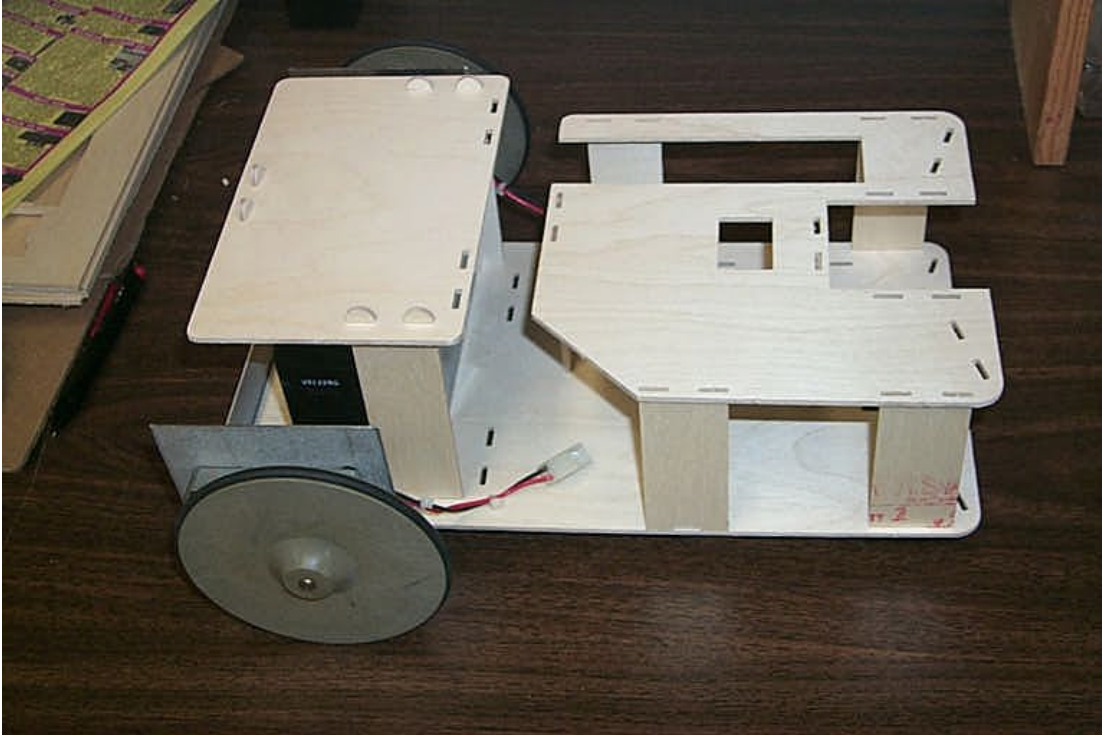


Figure 2

## 6. Actuation

I originally intended for NumBot to use simple hacked servos for actuation. However, the size and weight of the platform made it necessary to have more power and torque. This is provided by two “Robot Wheels” purchased from BG Micro, each mounted to a DC motor via a gearbox. Though these motors actually draw less than 80mA each under normal operation, the gearbox and large wheels make this actuation system sufficient.

Figure 3 shows a picture of the robot wheels used on NumBot.

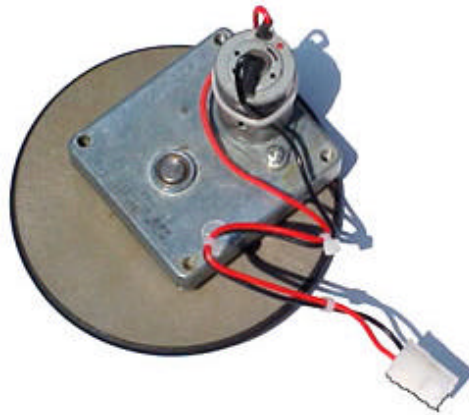


Figure 3



The main drive motors are connected via an H-bridge motor driver circuit based around the L293DNE. This circuit allows the full 12V supply to power the motors in either the forward or backward directions. Speed control is accomplished by varying the duty cycle of a TTL-level PWM signal from the microcontroller; this signal and direction control are inputs to the motor driver circuit. Figure 4 shows the Protel DXP schematic for the motor driver circuit; the 16-pin DIP is the L293DNE.

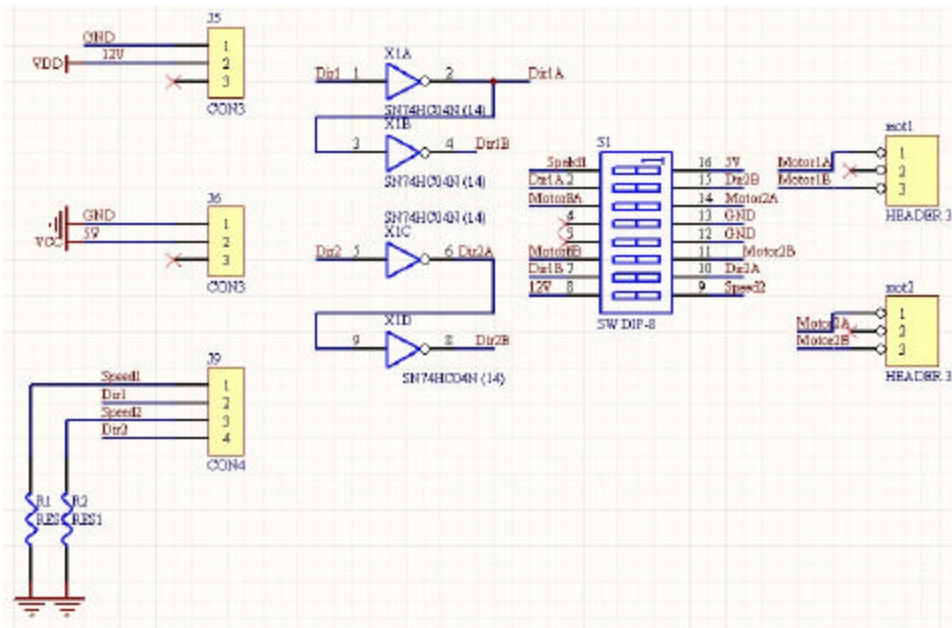


Figure 4

The motor-controller circuitry was implemented on a stand-alone motor-driver board. This board is pictured in Figure 5.

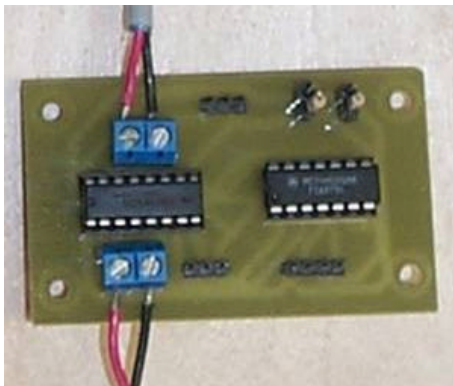


Figure 5

The two wheels which drive NumBot are mounted at the rear of the robot and a single caster is used at the front of the robot for a level and stable stance. Differential drive is used for turning; this gives NumBot enough mobility and maneuverability to perform the tasks for which it is designed.

## 7. Sensors

NumBot operates on the floor when searching for blocks and positioning itself with a block. Thus it needs basic obstacle avoidance capabilities while roaming this “dangerous” environment.

To prevent collision with walls or other large obstacles, the robot uses infrared range finders to determine the distance between it and an object. While I originally planned to integrate bump sensors on NumBot, the large size of the platform and effectiveness of the range finders led me to omit them from the design. Figure 6 shows the Sharp range finder that NumBot uses.



Figure 6

Though I originally used the camera to detect the presence of a block, the delay in detection with this method led me to add an infrared break-beam sensor on NumBot. This uses an 880-nm IR LED and a corresponding IR phototransistor to produce a signal sent to the microcontroller when a block is present.

The most interesting sensor on NumBot is the camera. This is used to grab an image of the block, providing input to the image processing system. The camera used is a simple, inexpensive (\$5) USB camera which implements the OV511 interface. The camera is located directly above the slot at the front of the robot and faces down at the upward face of the block. Figure 7 shows an actual image grabbed by NumBot’s camera.



Figure 7

## 8. Electronics

The brain behind the operation of NumBot is an Atmel ATmega128 microcontroller on a letATworkII board. This microcontroller is responsible for all control of every component in the system, including the IR sensors, motors, LCD, user buttons, and the Mini-ITX board. The letATworkII is mounted on the central processor board, which was designed using Protel and cut out using the T-Tech machine. This board also has the voltage regulator as well as the 12-V power connector for the Mini-ITX board. Figure 8 shows the central controller board with the letATworkII attached.

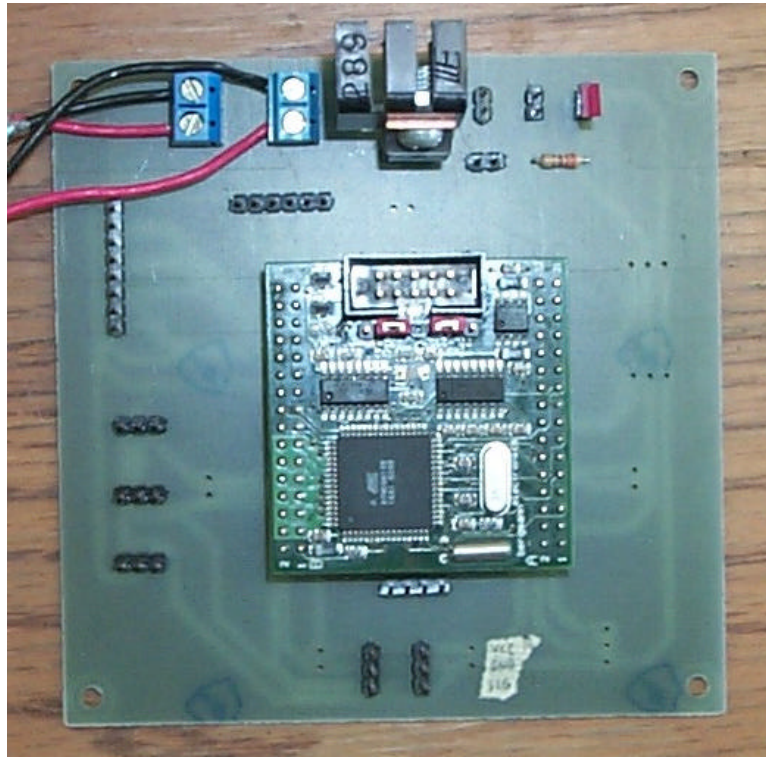


Figure 8

### USER INTERACTION BOARD

Another part of the electronics package on NumBot is the user-interaction board. This board houses the LCD and has connectors for the buttons which can be used to interact with NumBot. Figure 9 shows the user-interaction board with the attached LCD screen.

The LCD was crucial during the debugging process, and it also serves to let the world know what NumBot is doing while the robot is operating.

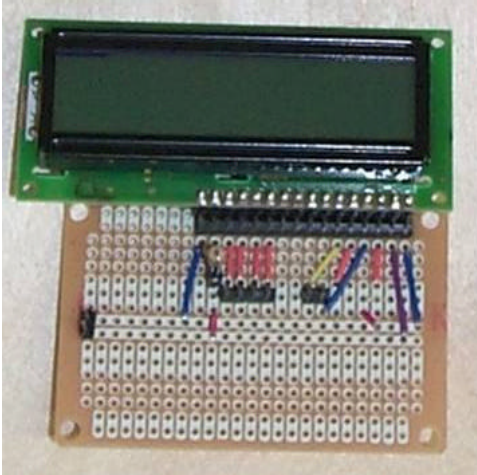


Figure 9

### MINI-ITX SYSTEM

The image-processing system is based on the VIA EPIA 5000 Mini-ITX form-factor computer. This is a full-fledged PC system which has an x86-architecture VIA processor, full bus, IDE controllers, VGA hardware, and more on a single board measuring 170 mm on each side. Figure 10 shows a picture of the Mini-ITX board used on NumBot.

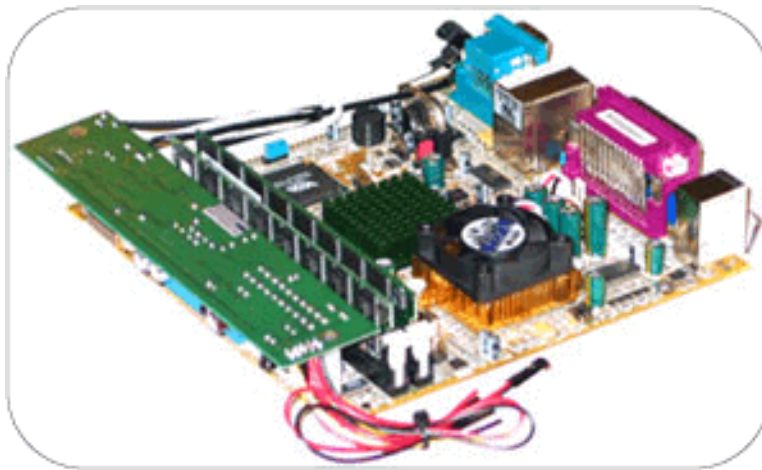


Figure 10

The figure above also shows the 60-W ATX power supply which attaches directly to the Mini-ITX board. This was used to supply power to the PC system from the 12-V DC line coming from the central controller board.

## **9. Behaviors**

While NumBot was designed with number detection in mind, the robot exhibits several distinct operating behaviors. Upon completion at the end of Fall 2003, NumBot's

behaviors include obstacle avoidance, partial number detection, and camera-based line following. It is worth noting that the line following behavior was not part of the original design plan; it was added towards the end of the summer.

### ***9.1 Obstacle Avoidance***

The most basic of NumBot's behaviors is obstacle avoidance, which is fundamental for an autonomous mobile agent. The technique used to avoid obstacles is very simple. NumBot will simply drive forward, measuring left and right IR values as it moves. Once the average of the last five measurements from either the left or right sensor has exceeded the threshold, NumBot will stop, move in reverse a set distance, and then turn approximately 30 degrees before moving forward again.

The forward-facing IR sensors proved very effective at detecting obstacles in front of NumBot. With the threshold value used, NumBot could successfully stop (and back up) when approximately 10 inches from a potential obstacle.

### ***9.2 Number Detection***

Number detection was the prominent behavior for which NumBot was designed. The goal was to determine the number represented on a children's block painted with a single decimal digit.

The overall process of number detection consists of the high-level steps listed below.

1. Obtain image of the block using the on-board camera
2. Perform image pre-processing
3. Run primary number-detection algorithm on processed image
4. Report results of the algorithm

When I originally decided on NumBot's behavior, the plan was to use a "simple" template-matching algorithm to complete step 3 of the above process. A lot of experimentation was performed to explore this technique. Unfortunately, things were not so simple.

Because of the difficulties encountered with various lighting conditions, the actual step-3 number-detection algorithm used for NumBot was based on object-detection code from Dr. Nechyba's image-processing library. This technique involves using sample images of each of the numbered blocks to be recognized to build a database of test images. This test-image collection is created (by hand) before the robot operates. During the number-detection process, the object-detection algorithm searches for the 'object' represented by each test image in turn. For each test image—for each number value—the algorithm returns a coefficient representing the degree of match with the source image. The number value associated with the smallest coefficient of difference from the source image is then identified as the result.

As this is being written, NumBot can successfully perform partial number-detection. The robot can respond to the presence of a block, grab an image of the block, determine whether it is either a 1, 2, 3, or 4, and then respond by pushing the block either to the left or right based on its value or stopping if a 4 is found.

### ***9.3 Line Following***

The line following behavior was added late in NumBot's development stages. Based on the idea originally suggested by Kristen Allen, another undergraduate student in the Machine Intelligence Laboratory, I realized that the downward-facing camera could be used to perform other behaviors than number detection, such as line following. From that realization my goal was to have NumBot follow a black (electrical-tape) line placed on a white surface.

The line-following behavior was implemented with some success. With a fairly smooth electrical-tape line on the laboratory floor surface, NumBot could successfully track the line while moving slowly.

## **10. Experimental Layout and Results**

Experimentation was an important part of the development of my robot. Because the environment in which the robot was intended to operate is inherently unpredictable, no mathematical model suffices as a basis for design details or robot calibration. Thus extensive experimentation was carried out, both during the design phase, to determine how best to design and implement particular subsystems, and later during the integration phase, to determine how subsystems best perform together and how they should be calibrated.

Because of the image processing involved in NumBot's operation, extensive testing of the camera system and associated software were performed. Once the physical camera mounting and setup was established, much time was spent experimenting with sample images to explore image-processing techniques and algorithms.

In particular, the originally planned technique for determining the number, template matching, was exercised extensively. With an ideal test environment and very clean test images, template matching was successful. However, further experimentation with the actual NumBot platform and environment established that template matching was too sensitive to real-world image variations, particularly due to lighting conditions. This is one example of real-world experimentation showing a priori thinking to be unrealistic.

## **11. Conclusion**

At the present time, NumBot is able to correctly carry out obstacle avoidance, slow-speed camera-based line following, and partial number detection to identify a 1, 2, 3, or 4. Thus the essential goals for the robot project have been met, and one additional behavior was completed.

In particular, I am pleased with the successful implementation of on-board image processing. I was initially concerned that it would be difficult to get image-processing hardware on a small platform, but I am glad I took on this challenge and am pleased with the hardware results. I definitely feel that the platform, with the Atmel microcontroller and Mini-ITX form-factor PC, could be used for other computer-vision applications. With appropriate software, many interesting behaviors could be implemented using the platform.

If future work can be carried out with the current NumBot setup, the main goal will be to design and implement more advanced and more mature software, in particular for the image-processing. With more advanced algorithms, full number detection as well as other behaviors could likely be completed.

Overall, the process of developing NumBot has been worthwhile and valuable for applying engineering principles and techniques to develop a robot which interacts with the real world.

## References

Many people contributed to my work during the development of NumBot. Many thanks go to the faculty directors of the Machine Intelligence Laboratory, Drs. Arroyo, Schwarz, and Nechyba. The IMDL TA's, Uriel, Vinh, and Louis, also provided many tips and recommendations based on their experience with robotics. Each of the other participants in the Summer 2003 REU program also contributed helpful advice, input, and counseling throughout the summer.

Two websites which were very important during the development process were the IMDL course website, in particular since final reports from previous semesters were available, and the AVR Freaks website at [www.avrfreaks.net](http://www.avrfreaks.net).

The following vendors were used to purchase the many components which came together to form NumBot.

- BG Micro, [www.bgmicro.com](http://www.bgmicro.com)
- Jameco, [www.jameco.com](http://www.jameco.com)
- DigiKey, [www.digikey.com](http://www.digikey.com)
- Acroname, [www.acroname.com](http://www.acroname.com)
- Mark III, <http://www.junun.org/MarkIII/Store.jsp>
- All Electronics, [www.allcorp.com](http://www.allcorp.com)
- RadioShack, [www.radioshack.com](http://www.radioshack.com)
- Lowes, [www.lowes.com](http://www.lowes.com)



## **Appendix**

Seemingly volumes of source code were written over the course of the development of NumBot. This source code is available in the NumBot file library from the web or on CD-ROM.

It is worth noting that code for NumBot falls into 2 categories: C source code for the Atmel microcontroller, and C source code for the Mini-ITX system running Linux. Essentially all library code for the Atmel was written from scratch, but many libraries, such as Dr. Nechyba's image processing library (`CVDemoLinux`), were used in the Linux code.