Vol Retriever - 2005 IEEE Hardware Design Competition

Phillip Reid Rowan Jr.
*University of Tennessee - Knoxville*

Quincy Neil Styke IV
*University of Tennessee - Knoxville*

Follow this and additional works at: https://trace.tennessee.edu/utk-chanhonoproj

Recommended Citation
https://trace.tennessee.edu/utk-chanhonoproj/914
Vol Retriever – 2005 IEEE Hardware Design Competition
Senior Honors Project

Phillip R. Rowan Jr.
Quincy N. Styke IV

May 2, 2005

Mentor: Dr. Mongi Abidi
Department of Electrical and Computer Engineering
University of Tennessee, Knoxville
## Table of Contents

1.0 ABSTRACT ............................................................................................................................... 4

2.0 TASK DEFINITION .................................................................................................................. 5

3.0 INTRODUCTION OF BASIC ROBOT SYSTEMS ............................................................... 6
   3.1 INTRODUCTION .................................................................................................................. 6
   3.2 CHASSIS ............................................................................................................................. 7
   3.3 BALL RETRIEVAL ............................................................................................................. 7
   3.4 DRIVE TRAIN ...................................................................................................................... 7
   3.5 CPU .................................................................................................................................... 8
   3.6 POWER ................................................................................................................................. 8
   3.7 SENSORS ........................................................................................................................... 9
   3.8 VISION ................................................................................................................................. 9

4.0 PRELIMINARY RESEARCH/PROTOTYPES ......................................................................... 9
   4.1 INTRODUCTION ................................................................................................................ 9
   4.2 BALL RETRIEVAL DESIGNS ......................................................................................... 10
   4.3 DRIVE TRAIN SELECTION ............................................................................................ 12
   4.4 OBSTACLE AVOIDANCE ............................................................................................... 14
   4.5 SHAFT ENCODING ......................................................................................................... 15
   4.6 FLOOR MARKING SENSORS ........................................................................................... 16

5.0 DESIGN SHIFT ....................................................................................................................... 16

6.0 FINAL DESIGN ....................................................................................................................... 16
   6.1 INTRODUCTION ................................................................................................................ 16
   6.2 CONTROL SYSTEM .......................................................................................................... 17
      6.2.1 Overview of Controller Board ..................................................................................... 17
      6.2.2 Selection of I/O Board ............................................................................................... 17
   6.3 MIXED VISION/NON-VISION SOLUTION ..................................................................... 18
   6.4 BALL RETRIEVAL ............................................................................................................ 18
      6.4.1 Actuation ..................................................................................................................... 19
      6.4.2 Neodymium Magnets ................................................................................................. 19
   6.5 SENSOR SYSTEMS ............................................................................................................ 19
      6.5.1 Shaft Encoders ............................................................................................................ 19
         6.5.1.1 Optical Encoders ................................................................................................. 19
         6.5.1.2 Mechanical Encoders .......................................................................................... 20
      6.5.2 Infrared Start Sensor .................................................................................................. 20
      6.5.3 Floor Marking Sensors ............................................................................................... 20
      6.5.4 Obstacle Avoidance Sensors ..................................................................................... 21
   6.6 SOFTWARE ....................................................................................................................... 21
      6.6.1 Overview ..................................................................................................................... 21
      6.6.2 Integration of Sensors with a Non-Real-Time Operating System ................................ 21
      6.6.3 RTAI ............................................................................................................................ 22
      6.6.4 Pulse-Width Modulation Issues ................................................................................. 22
1.0 Abstract

Each year, the IEEE holds a hardware design competition at their Southeastern Conference. Many universities design and construct robots to compete in a specific task which is predetermined by the event organizers. The task for 2005 was to retrieve several metal balls from a playing course. The University of Tennessee’s entry utilizes a computer vision solution to complete the task. The design process is discussed, focusing on areas of the robot that the authors were involved in. These areas include the drive train, navigation sensors and algorithms, ball retrieval mechanism, and the obstacle avoidance system. The robot, known as the Vol Retriever, was unable to compete due to technical difficulties; however it did complete several successful runs in the lab.
2.0 Task Definition

The 2005 hardware design competition involved two robots attempting to retrieve several balls from a playing field and then to return with them to their respective starting points. The playing surface was constructed from a four by eight foot piece of ¾-inch thick plywood. The board was divided into two sides and each robot started on opposite sides of the playing field.

A scale diagram of the playing surface is given in Figure 1. In this diagram, the colors have been inverted for readability. That is, the black areas are actually white and vice versa. Each robot begins the round in a 6 inch by 6 inch white starting square. The robot is signaled to begin the round by an infrared light emitting diode, which is located beneath the robot. Once a round starts, each robot must navigate to the far side of playing course. Once there, the robot must locate and retrieve five steel balls. Each ball is located in a ¾-inch diameter hole drilled through the playing surface. Each hole is surrounded by a thin ring of white paint. There is a minimum distance of approximately one inch, edge to edge, between holes. The holes may only exist in a region defined by the blue squares in Figure 2, below.
Each team retrieves one of two colors of balls during each round, black or silver. After retrieving all five balls, each team’s vehicle must then return to its original starting square and be mostly within the square in order to finish the course. After the balls are acquired, they must remain in the robot and need not be deposited in any receptacle. If a robot fails to park properly, it is penalized and the time taken to complete the round is recorded as the maximum length, which is five minutes. When the robot has completed its task, it must illuminate a blue LED to signal that it has finished. Each round of the competition has a time limit of five minutes, and the robot to complete the task in the fastest time is declared the winner of the round.

Each robot that enters the competition must not exceed dimensions of 6” by 6” by 8” tall. The robot is not allowed to change shape at any point in the competition. However, for the purpose of ball retrieval, a robot is allowed to extend some type of arm or apparatus up to an additional 3” of one side of the robot. All robots must be completely autonomous and may not be remote controlled in any way. The robot may have a maximum weight of twenty pounds. Each robot must be safe for the judges, spectators and playing course. A robot is not allowed to pick up or otherwise disturb a competitor’s targets. If a robot does this, it is heavily penalized. Robots are also not allowed to attempt to interfere with each other. Links to a more detailed list of rules and playing surface construction details are located in Appendix A.

3.0 Introduction of Basic Robot Systems

3.1 Introduction

The robot developed by the design team is made up of several subsystems, which will each be briefly introduced here.
3.2 Chassis

Initial prototypes of the chassis were built using plexiglass and PVC plastic panels. The PVC panels, which are pictured below, were chosen for use in the final chassis. The PVC proved to be inexpensive, durable, easy to work with, and was aesthetically pleasing.

The chassis went through a number of changes prior to the final design. The modifications made throughout the semester were small, and the initial design remained intact. The chassis was designed to meet the dimension constraints while maximizing the available volume and surface area for hardware placement. The chassis was based upon a tiered design. The bottom tier was responsible for holding drive motors, the main battery, and the ball retrieval system. The top tier held the camera and the main processor board. Sub-tiers were added in the back half of the robot to house the power boards and motor drivers. These sub-tiers could be easily slid in and out, to allow quick access to the various subsystems of the robot.

![Figure 3: PVC plastic panels](image)

3.3 Ball Retrieval

The purpose of the ball retrieval system was simply to retrieve and store the metal balls as the robot moved across the course. The retrieval assembly must be able to retrieve five steel \( \frac{1}{2} \)" ball bearings, with either a black oxide or steel zinc coloring. The ball bearings will be located in an area measuring approximately 4' x 4' (see Figure 2). Each bearing-containing hole will be approximately \( \frac{3}{4} \)" deep and \( \frac{1}{4} \)" in diameter. In addition, the robot is not allowed to pick up opponent balls, located in similar holes on the opposite half of the board.

3.4 Drive Train

The drive train was made up of the drive motors, wheels and associated hardware, as well as the control circuitry for the motors. This system was responsible for moving the robot around the playing field. It is discussed in more detail in a later section.
3.5 CPU

This system was made up of a computer and/or a microcontroller. It was responsible for the overall control of the robot. This system takes all data gathered by the various sensors, interprets it, makes a decision, and issues commands to the various motors and actuators on the robot.

3.6 Power

The competition rules state that each robot must operate under its own power. The robot required voltage levels of 5V and 12V. Rayovac IC-3 fifteen minute rechargeable batteries were ultimately selected to power the robot. These batteries were advantageous for several reasons. These batteries fit the AA form factor and could be arranged in a variety of configurations to gain the voltage and/or current required. They also were rated at 2000 mAh, meaning that each cell was capable of sourcing a large amount of current. This was important as current draw for the total design was estimated to be 6.5 amps. Finally the batteries could be rapidly charged in as little as fifteen minutes. This was important since the competition required that the robot be able to compete in several rounds. The design team had battery packs custom made, each containing twelve IC-3 cells in series in order to obtain a suitable voltage level. One of these packs is shown below, installed on the robot in Figure 4. Directly above it is the power regulation board, which is responsible for stepping the battery voltage down to the level required by the computer.

![Figure 4: Battery pack and power regulation board](image)

3.7 Sensors

This system is responsible for providing all non-visual input to the control system. This system keeps track of the robot's position and orientation on the playing course. It also is responsible for detecting whether the robot is passing over a black or white region of the floor and for detecting obstacles in the robot's path.
3.8 Vision

The Vol Retriever appeared to be the only robot at Southeast Con 2005 to implement a computer vision solution. This method was more efficient, as it would take a picture of the course, predict where the targets were located based upon that image, and then command the robot to travel directly to those locations. Other robots simply swept the entire half of the board with their ball retrieval mechanism engaged. This vision approach was implemented using a Logitech Webcam 4000, and an open source image recognition software package called Open CV. Combined with the supervisory level software, the vision system guided the robot around the course and was a major advantage compared to the other competing robots’ methods of acquiring the targets.

![Logitech webcam](image)

*Figure 5: Logitech webcam*

4.0 Preliminary Research/Prototypes

4.1 Introduction

This project began as a two semester effort. The first semester was oriented toward research, design, and general education of the availability of consumer parts for robotic construction. Two groups were formed with different implementation strategies as their goal. The first group took a previously constructed “base” robot, shown in Figure 6, and began implementation of a computer vision system and ball retrieval system onto the existing chassis. Several different methods of ball retrieval were tested along with multiple cameras.
The second group began a totally new design of a robot and pursued a non-vision, sensor-oriented approach to completing the course. This group researched a variety of sensors that could detect distances, changes in color, and shaft rotation. The chassis of this design was the first to use the PVC paneling that would ultimately be used in the final robot design. This robot is seen below in Figure 7.

4.2 Ball Retrieval Designs

When brainstorming first began for ball retrieval designs, the amount of space available for the ball retrieval mechanism was unknown. Without a known restriction, the team came up with several different concepts for retrieval mechanisms. The first concept, seen in Figure 8, was similar to a propeller. It was made up of several rotating blades, each of which had magnets affixed to them. The blades would spin beneath the robot, pick up the targets through magnetic force, and then deposit them in some type of holding bin.
The next design involved lowering an array of magnets using a linear actuator. The array was made up of several neodymium magnets, which provide a much larger amount of coercive force than ferrite magnets. The retrieval unit shown below occupied too much space and was too heavy.

The use of a conveyor belt offered a design that took up considerably less space. This design was powered by a DC motor that would constantly rotate a conveyor belt which had three magnets equally spaced along its length. Although it was smaller than the two previous approaches, concerns arose with the method the magnets were attached to the belt. When turning along the corners of the system, the magnets were being separated from the belt and were held on by a very small contact point.
Although each approach had advantages and disadvantages, the team was still not satisfied with these concepts. These designs allowed the team to learn and gain useful information that would be used in the spring semester to design and create the final ball retrieval unit.

4.3 Drive Train Selection

The first issue to be discussed in the design of the drive train was the method of locomotion to be used. Several options were generated. One of these was a robot designed to move solely in an orthogonal manner. The design had two sets of wheels that were offset from one another by ninety degrees. By activating one set of wheels, the robot would be able to move from side to side. The other set was responsible for moving the robot forwards and backwards. This design presented an advantage in that the robot would never have to change its orientation, but instead could simply translate around the board. Since the robot’s orientation would never change, it would not have been necessary to take that into account when calculating the robot’s position, greatly simplifying the calculations. However, this design was discarded due to the high parts count and difficulty of implementation. One of the major drawbacks to this design is that a wheel occupied each side of the robot, leaving no good space to place the ball retrieval system.

Another idea discussed was to have a three wheeled robot. Two powered wheels would be in the back, while a third wheel in the front would be responsible for steering the robot. This design had many problems as well. In addition to the high number of actuators and wheels required, it also made it difficult to calculate the robot’s position. With this design when the robot turned it would not rotate about its center. Because of this, not only would each turn cause a change in the robot’s heading, but also in its position. This design was discarded because a much simpler solution was devised.
The ideal design for the drive train was one that had a low parts count and did not over complicate the calculation of position. The chosen design for the drive train uses only two motors and wheels. The wheels are aligned with the robot’s center line, as shown in Figure 11. The robot is balanced by skids in the front and back to keep it from tipping over. This allows the robot to turn in place, much like a tank does. As a result, the robot’s position is not affected by turning which simplifies the calculations considerably.

![Prototype version of the Vol Retriever](image)

All parts for the drive train were ordered through a company called Lynxmotion, which offered wheel hubs and motor mounts designed to work with the motors they offered. It was advantageous to purchase motor mounts and hubs designed to work with the motors since this reduced the amount of machining necessary. The motors used in the design have a maximum speed of 138 RPM and are connected to wheels 2.63 inches in diameter. This allows the robot to move at a maximum rate of 19 inches per second. The playing field is only 4 by 8 feet, so this system offers more than enough speed. An h-bridge circuit, pictured in Figure 12, was used to control direction and speed of the motors. Each motor was controlled by three electrical control lines. Two of the lines controlled motor direction; it was commanded by raising one of the two lines high for each direction. The third line received a pulse width modulated signal in order to control motor speed.

The motors used were 12V DC motors. Their average no load current was approximately 123 milliamps and their stall current was approximately two amps. The h-bridge purchased was capable of delivering 2 amps continuously; therefore even in a stall condition the control electronics were adequately sized. Both R/C servos and stepper motors were proposed for use in the robot; however both were dismissed due to the fact that they operate at a slower speed than a comparable DC motor. The servos would have required extensive modification for use in this application, as servos are not designed to rotate continuously. Rather, they generally have range of motion of 180 degrees. Stepper motors require that each of their coils be fired in a certain order, and would have required a larger number of control lines to operate.
4.4 Obstacle Avoidance

While there were no obstacles built into the playing course, it was necessary to avoid hitting the other robot. Several options were discussed for this system, including contact switches, sonar, and infrared distance sensors. Contact switches were disadvantageous for two reasons: the robot would not detect anything until a collision had occurred, and a collision could throw off the robot’s position tracking function. The effective range of most sonar systems was beyond what was desired. The systems functioned well at distances of approximately 2 to 20 feet, but the robot would need to be aware of what was happening at a closer range, considering the size of the playing surface.

Eventually infrared distance sensors were decided upon. The model the team selected was the Sharp GP2D120. It had an effective range of 4 to 30 centimeters and output an analog voltage corresponding to the distance of the object. An example output characteristic from the Sharp datasheet is shown below. This sensor required no external control circuitry and was easily interfaced to the microcontroller the team was using at the time. The sensor was also advantageous due to the fact that its readings were not very dependent on the object’s reflectivity. Many infrared sensors simply measure the brightness of reflected infrared light from an object in order to determine distance. In this scheme a mirror and a brown paper bag held at the same distance would yield different readings, as the mirror would reflect more light than the bag. The GP2D120 actually measures not the intensity, but the angle of the incoming reflected light. As a result, the reflectivity of the object plays a much smaller role in the reading received.

Two distance sensors were mounted in the front of the robot. It was decided that it would not be necessary for the robot to be aware of objects to the side or behind it. As long as the robot’s path in front of it was open, it was assumed that either the other robot would stop and/or UT’s robot would escape the situation.
4.5 Shaft Encoding

In order to keep track of the robot's position, it was decided to use devices called shaft encoders. Shaft encoders work under the basic principle of counting fractions of each revolution of a shaft. In optical shaft encoders some type of a wheel is attached to the rotating shaft. The wheel is marked with alternating black and white stripes that extend radially from the center of the wheel. A sensor is placed close to the wheel which detects the transitions from black to white as the shaft spins. If one knows the diameter of the spinning wheel, then the accumulated number of transitions can be resolved into a distance. For example, a one inch diameter wheel has a circumference of \( \pi \). If the wheel is fitted with a shaft encoder wheel that has 50 divisions, then each division represents a distance traveled of \( \frac{\pi}{50} \). If the wheel is spinning and the shaft encoder detects 25 transitions, then a distance of \( \frac{\pi}{2} \) can be calculated easily. A picture of one of the robot's shaft encoders is shown in Figure 14.
The sensors used to construct the shaft encoders were Hamamatsu P5587 photoreflectors. These devices house a tiny infrared LED and phototransistor pair. Light is emitted from the LED, and the phototransistor is switched on and off according to whether the sensor is viewing a light or dark region. As a result when the wheel is spinning, the sensor generates a pulse train. The logically high regions represent white areas, while the low ones represent the dark areas. These pulses are then counted by the control system and the distance traveled is calculated.

4.6 Floor Marking Sensors

The Hamamatsu photoreflector was used extensively in the design of the Vol Retriever. In addition to shaft encoding, these sensors were used as floor marking sensors. Since the playing surface is either black or white, these devices were well suited for detecting color transitions as the robot moved across the board. An array of several of these sensors was assembled and placed on the front edge of the robot. Smaller arrays were constructed and placed under the rear corners of the chassis as well.

5.0 Design Shift

As previously stated, the first semester was used as a research and learning experience with two different groups trying a variety of approaches toward achieving a common goal. When the spring semester began, the groups decided it would be best to combine in order to best utilize the talents of each person. By combining vision and non-vision approaches the Vol Retriever would have a redundant system with backup capabilities. With this decision, some hardware changes were made in order to accommodate both methods. The goal of the second semester was to move both teams to a common development platform. Different systems could then be put together in the most effective combination.

6.0 Final Design

6.1 Introduction

Having used the fall semester to research and test equipment, the team began the spring semester by selecting and completing prototypes from the previous semester.
These individual systems would be incorporated into one robot which would later be used in the 2005 IEEE SECON Robotics Competition. The systems described below may differ from preliminary designs and are the final concepts built into the competition robot.

6.2 Control System

In order to satisfy the processing needs and communication between hardware needed for operation of the robot, a computer board was used as the command center of the robot. An additional I/O board provided necessary interface channels to communicate with lower level hardware on the robot.

6.2.1 Overview of Controller Board

The main controller board is very similar to the motherboard located in personal computers and laptops, except on a much smaller scale. Packaged in the PC-104 form factor, this compact size motherboard is only 3.6” x 3.8”. Its small size was a major factor in its selection. Manufactured by Kontron Inc., the MOPSledVE offered computing power that rivaled desktop machines. This board allowed the team to connect a laptop hard drive for storage, run a Linux operating system, and program in the C and C++ coding languages.

![Kontron PC-104 controller board](image)

6.2.2 Selection of I/O Board

The non-vision group used the MC68HC812A4 microcontroller as its control system during the first semester. Shifting from a microcontroller to an actual computer made it necessary for the team to seek out a new input/output solution. While a microcontroller has many pins available for I/O, a computer’s ports are generally more specialized, and not well suited to connecting logic level devices. Several PC-104 I/O boards were available from different manufacturers. The one decided upon was the Onyx-MM manufactured by Diamond Systems Incorporated. This board stacked on top of the Kontron board and offered 48 lines of digital I/O. It also had 3 16 bit counter/timer ports with hardware interrupt capability. These ports allowed the team to be able to read devices that would update periodically, like the shaft encoders, without having to poll the status of the devices constantly.
6.3 Mixed Vision/Non-Vision Solution

For the both the first semester, and the beginning of the second semester, the vision and non-vision sensor systems were being developed separately. The goal was to have each system be able to provide the robot with all the data necessary to navigate the course successfully. However, it was decided that the robot would be much more successful if all available data was pooled from each sensor system. For instance, it was much simpler to detect an obstacle with an infrared sensor, than to pick out the opposing robot from the image of the playing board and background using computer vision. However, as a result of this combination, some information became redundant. The floor marking sensors were given much less attention after this point, as the camera was capable of determining with reasonable accuracy what part of the board the robot was on.

6.4 Ball Retrieval

For the final ball retrieval design, it was decided to take advantage of a rule which states that a robot may increase in size in one direction from its original dimensions by three inches. This stipulation allowed the team to implement a drawbridge type approach for retrieving the balls. Since the robot had to begin each round at no larger than 6” x 6” x 8”, the drawbridge could be positioned upright and lowered once the competition began. This gave the team increased flexibility in working with space constraints and also allowed the robot to extend its reach so that it wouldn’t need to get too close to the edge of the board in order to cover the required area. For the drawbridge a T-brace was used with an array of five magnets on the upper section and hinged the brace on the bottom level of the chassis as shown in the figure below. The placement of the hinge on the chassis helped prevent the drawbridge from being lowered too close to the playing surface. When tested, this method proved to be the most reliable and effective method of ball retrieval.
6.4.1 Actuation

The drawbridge was pulled up and down by piece of cable attached to an actuator. Originally the drawbridge was tested using a Hitec 3.5" sail winch servo which was able to extend and retract a sufficient length of the cable. Unfortunately the team was not able to fully integrate this servo into the robot due to extremely small timing constraints that were needed to accurately control the servo. These difficulties are discussed in greater detail in section 6.6.2. As a replacement for the sail winch servo, a DC motor was used with the 3.5" drum from the servo attached to it. The DC motor was controlled by limit switches, which alerted the control system when the drawbridge had either been fully raised, or fully lowered.

6.4.2 Neodymium Magnets

In order to pick up the balls which would be located in the holes of the 3/4" deep playing surface, extremely strong magnets were needed. Neodymium magnets proved to be exactly the type of magnet the team needed and seemed to be a popular choice among other teams at the IEEE competition. These 1" x 1" x 3/8" magnets were placed in a row of 5 along the T-brace and provided the coercive force necessary to extract the targets from their holes with excellent repeatability.

6.5 Sensor Systems

6.5.1 Shaft Encoders

6.5.1.1 Optical Encoders

The optical shaft encoders that were ultimately used in the final design were the same ones developed during the fall semester. The number of divisions on the encoder wheel was increased however, from 50 to 100 segments. It was discovered, however, that with a larger number of division per rotation, the photoreflector sensor was no longer
putting out a pulse train with a 50% duty cycle. The sensor was unable to react quickly enough to the transitions occurring before it, and this resulted in a signal with a near 70 percent duty cycle. This caused the position calculation software to periodically not detect black/white transitions. As a result the robot’s position estimates became worse and worse. This problem was remedied by increasing the width of the black divisions in relation the white ones. This allowed the photoreflector time to adjust to the transitions and restored a 50 percent duty cycle.

6.5.1.2 Mechanical Encoders

For a short period of time the robot used mechanical shaft encoders. These were chosen because there was concern that the optical encoders were not providing the necessary resolution to keep track of position. These mechanical encoders were actually designed to be used as a knob in an instrumentation panel. The team obtained custom made gears and constructed new drive wheels in order to interface these encoders to the drive shafts. The new system gave the robot approximately 125 pulses per revolution, a great improvement over the 50 pulses that were being used at the time.

However, there were many problems with these shaft encoders. Since these encoders were designed to be used on an instrument panel, they did not hold up well being rotated hundreds of times in only a matter of seconds. The mounting scheme was complicated, difficult to adjust, and the gears either did not mesh at all or they were pressed too tightly together. This caused motors to jam at low speeds as well as heavy wear and tear on the encoders. Replacing encoders was a lengthy process, and each encoder lasted approximately two days before requiring replacement. These interruptions seriously stalled development for a number of days, and eventually the optical encoders were reinstalled.

6.5.2 Infrared Start Sensor

In normal competition, once the robots are placed in the starting square, they will receive a start signal from an infrared LED placed in the center of the starting square. This signal will be steady on and when detected, the robot should begin competition. In order to detect this signal, an infrared sensor was placed in the bottom tier of the chassis with the “eye” of the sensor looking down on the board.

6.5.3 Floor Marking Sensors

The number of floor marking sensors was substantially reduced in the new design. In the initial design, the floor marking sensors aided in keeping track of position and located target holes. Once the shaft encoding system was integrated with the vision system, neither of these tasks was necessary. The floor marking sensors were retained however, with the hope that they could help the robot park in the start square at the end of the round. The robot’s camera could not see in the area immediately around the robot, so the floor marking sensors would be the only information that would hint at the position of the robot. During normal operation however, the camera was capable of detecting at what coordinates the targets were located.
6.5.4 Obstacle Avoidance Sensors

The same line of sensors from the Sharp Corporation was chosen for the new version of the robot. However, it was necessary to switch from the analog GP2D120 to the digital GP2D202. While the digital I/O board offered several I/O ports, it did not have any sort of analog to digital converter. It was less expensive to switch from analog to digital sensors instead of purchasing and interfacing an ADC. Instead of an analog voltage, the GP2D202 output an 8-bit number which corresponded to the sensor reading.

6.6 Software

Once hardware was decided upon and implemented, software was written to operate and control the webcam, motors, sensors, and other devices used to detect holes, move to desired locations, and retrieve the balls. As the software is what is really behind the operation of the robot, the following sections detail the different pieces of software written to run the robot.

6.6.1 Overview

Writing the software for the robot was perhaps the most time consuming aspect of this project. The robot had to gather information from different sensor systems, make calculations, and output commands all at once. This was especially challenging with the overhead of an operating system running in the background.

6.6.2 Integration of Sensors with a Non-Real-Time Operating System

Using a microcontroller in the first semester to control the robot did come with one major benefit: timing. When the microcontroller was told to perform a task, it did it exactly when it was programmed to and performed consistently from run to run. While running a computer greatly increased the robot’s computing power, the use of a non real time operating system like Linux caused some problems. The average computer, even when “idle”, is doing many things at once. It is updating the image on the screen, monitoring any network traffic sent to it, and performing other tasks in the background. While there are many tasks that need to be done, a computer can only perform one task at a time. It accomplishes this multi-tasking behavior by scheduling processing time for each task. The computer works on one task for a short time, then works on a different task for a short time, and this process repeats itself over and over. A good analogy would be juggling. Each hand can only comfortably hold one ball, but if one is a good juggler, then five or six balls could be handled by only two hands.

The problem with this is that in a system such as a robot, some tasks cannot wait. For instance, as the shaft encoders spin, pulses are ticking away no matter what the computer does. If the computer is busy with another task, even for an incredibly short period of time, and misses that pulse, then data is lost and the calculated position begins to become inaccurate. A big challenge in the design of this robot was overcoming this delay when dealing with sensors that needed to be read in real time. This was also the reason that the Hi-tec servo had to be abandoned as an actuator for the ball-retrieval mechanism. Servos require a pulse train signal with a very consistent pulse width. The servo is commanded where to turn based upon the width of each pulse it receives. In a
scheduled operating system, it is very difficult for a computer to output a consistent
signal. This would cause the servo to rotate to random positions nearly all the time that it
was being sent a signal.

6.6.3 RTAI

RTAI stands for Real-Time Application Interface and is a real-time extension to
the Linux operating system kernel. An extension such as this allows for real-time
operations to be carried out without the usual delay inherent with most operating systems.
With a real-time extension, commands issued to motors or other devices would be carried
out instantaneously instead of waiting on the operating system to service these requests.
It was desired to incorporate this extension to the Linux operating system kernel we were
running, but unfortunately we were not able to get these two separate pieces of software
to work together. Had this been accomplished, it would have allowed us to use the Hitec
servo mentioned above as well as given us more precise control of our motors.

6.6.4 Pulse-Width Modulation Issues

Pulse-width modulation control works by switching the power supplied to the
motor on and off very rapidly. The DC voltage is converted to a square-wave signal,
alternating between fully on (nearly 12V) and zero, giving the motor a series of power
"kicks". If the switching frequency is high enough, the motor runs at a steady speed due
to the momentum of its rotor. By adjusting the duty cycle of the signal (modulating the
width of the pulse), that is the time fraction it is "on", the average power can be varied,
and hence the motor speed. This technique was used in the Vol Retriever to control the
speed of the drive motors. Since a real time extension to the operating system could not
be implemented, the team was limited in the amount of precision with which the duty
cycle could be varied. The team was able to overcome this obstacle since the motor speed
was only dependent on the average pulse-width modulation over a period of time.
Though during some periods the duty cycle may have been 100 percent when the target
was 50 percent, on average the same value was approached. An example of pulse-width
modulation is shown below in Figure XX.

![Pulse-width modulation example](image_url)
6.6.5 Navigation

Once the pulse-width modulation was tested and designated as adequate for our motor control, an overall navigation software package had to be written. This software package would take in data from the vision system and shaft encoders to determine the robot’s current position and how the motors would need to be adjusted in order to drive the robot to the desired targets. In order for all processes to run simultaneously, an advanced programming technique called threading was used to implement the navigation software. Threads are similar to a process, except that they are called from a main process and they share processor time. Using threads allowed the robot to have the pulse-width modulation for motor control constantly running while concurrently adjusting the speeds to the motors in order to navigate to the holes. Several separate functions were running together in our navigation software package and are described in the following sections.

6.6.5.1 Constant Calculation of Position

In order for the robot to be able to navigate to the positions specified by the vision system, the robot needed to know where it was currently located in the playing field. In order for this to occur, the robot’s software constantly updated the position of the robot. This part of the software relies on the optical shaft encoders which would detect how many revolutions each wheel had turned. Through a set of geometric calculations, the revolutions of each wheel were manipulated into data that represented how much the robot had moved in x and y coordinates. This data was also manipulated in a manner that let the team know the robot’s current heading in degrees relative to the playing field. By setting up a coordinate system relative to the playing field and since the robot would always begin a round in the same position, the team was able to have a constant reference point from which to start each round of competition. Once the robot began moving, it calculated its position and heading as mentioned above and at any point in the round could tell a user where it thought it was located and at what heading. After numerous revisions this software function could accurately keep track of the coordinates throughout an entire round without much error in the data.

6.6.5.2 Constant Calculation of Navigation Vector

Once the robot was able to keep track of its position, the team had to implement a means of getting to the desired targets. This was also based around the coordinate system established relative to the playing field. By knowing the current coordinates and having a set of coordinates fed into the navigation software by the vision software, the robot was able to calculate a navigation vector that would take the robot to the desired target. This vector contained the distance and heading to the target and was used to adjust the motors accordingly.

6.6.5.3 Adjusting Motor Control Feedback

As mentioned above the robot’s drive train consisted of a pair of DC motors that were controlled using pulse-width modulation. The modulated signal sent to the motors controlled the speed at which the motors turned the wheels of the robot. By interpreting the navigation vector, current position and heading, a set of conditions were set up in
order to adjust the duty cycle for each motor independently. This gave the team the
ability to issue commands to each motor to move forwards and backwards at various
speeds. With the independent control of each motor this allowed the robot to move in a
straight path, adjust the heading in small degrees, turn in place or turn using one wheel as
a pivot. Since the pulse-width modulation of the motors ran as a thread in the
background, another piece of code was written to gradually adjust the speed of the motors
in order to accelerate or decelerate in a manner that prevented the wheels from breaking
traction. If the wheels were suddenly subjected to a 100 percent duty cycle while they
were stopped, the wheels would spin, and the robot’s calculated position would be
several inches off as a result.

6.6.5.4 Local Speed Zones

In most cases when trying to acquire the balls, the robot would need to be turning
to adjust its heading. While testing it was realized that if the turning speed was too great
that the robot might overshoot its target and miss the ball. In order eliminate this, a set of
“speed zones” were created which limited how fast each motor would turn depending on
the distance to the target. When the robot started the competition it would run at full
speed until it came into new speed zone at which point it would slow down to that speed.
By doing this the robot could adjust its heading adequately to find the target without
overshooting the holes and missing the balls.

6.6.5.5 Reorienting to New Heading

After acquisition of a ball, it was desired to orient the robot so that it was facing
the next hole. Once facing the next hole, the robot could take another picture, update the
coordinates of the desired targets, calculate the navigation vector to the next point, and
proceed to that target. In order to reorient the robot, the coordinate of the next hole
previously given by the vision system was used and the robot would then pivot to the new
heading about one of its wheels. Although there may have been some variance between
the previously calculated coordinates of the next hole and the actual coordinates, the
robot only needed to be facing in the general direction in order for the vision system to
reanalyze the board and correct for any error. By operating in such a manner the vision
system was able to accurately determine coordinates for each ball and direct the robot to
each target.

6.6.6 Object Avoidance

The obstacle avoidance code consisted of two main functions. The first was a
driver to read data from the GP2D202 distance sensors. The sensors were equipped with
an enable line and a data line. The function brings the enable line low for approximately
70ms to signal the sensor to take a reading. Then the enable line is pulsed eight times in
very rapid succession. With each pulse the data line on the sensor outputs one bit of the
8-bit reading. This value is read into a variable, converted from binary to decimal, and is
then interpreted into a distance based upon the output characteristic of the sensor.

The second piece of code determines what action should be taken if an obstacle is
detected. The function generates a waypoint for the navigation function to travel to. If the
robot is near the edge of the board, or is not close to facing the center of the board, then it
issues a stop command to the navigation function. Otherwise, the function generates a
waypoint that is approximately 15 inches away from the robot’s current position, and in
the direction of the center of the playing surface. Once the robot has evaded the target,
normal operation resumes.

6.6.7 Vision Overview

As stated earlier, the Vol Retriever was the only robot to use a vision approach to
acquire the targets. This approach allows the robot to go directly to the holes instead of
traversing the entire playing field. The vision software relies on OpenCV, an open source
computer vision software package available free of charge. This software allows the
robot to take images from a webcam and manipulate the image in order to gather
information from it. A “raw image” from the webcam is a color picture of the playing
field. This image is processed to go from the “raw image” shown on the left of Figure 19
to an image after basic image processing shown on the right. The image processing first
does basic edge detection and will draw out lines between two distinct colors or “edges”.

![Figure 19: Raw image vs. image after edge detection](image)

The next step in the image processing is more advanced in that it examines the
various edges and determines which edges are boundary lines, which are the circles
around the holes, which edges form the center stripe, and which edges should be ignored.
This uses a complex set of conditions and falls into the category of advanced image
processing. With the image in Figure 20 our vision software also calculates distances
from its current position to each of the holes it finds. If the holes are determined to be
valid targets, a set of coordinates are stored for the holes and are used to tell the robot
where to go. Figure 21 displays an image of the robot’s current position, the viewable
area of the robot, and where it thinks each hole is located as compared to its actual
location. As the robot moves closer to the center of the playing field the image
processing becomes more accurate and will find the holes to be at their true locations.
Not only will the software find all the targets, but it will also calculate the shortest path to
take to acquire all five balls and return home.
6.6.8 Vision/Navigation Integration

Originally developed as individual pieces of software, a significant amount of time was spent integrating these two software packages. The vision system had originally been developed on a robot chassis that was not capable of movement and would be moved by hand to simulate motion in order to run through the process of a competition. The navigation software was originally developed using a set of hard-coded coordinates for targets in order to fine tune the motor control. Once these two packages were ready for integration, a top level program was written to allow for these two programs to work together in order to accomplish the goal of retrieving the balls and returning home. Modifications were made to each package in order to set up methods of communications for each process to relay information to one another. Several issues faced are discussed in the following sections.
6.6.8.1 Initialization timing

As with most hardware and the software that controls it, timing is critical issue and must be established in order for proper execution. Both the vision and navigation software required certain initialization and startup functions to be performed before operation. Since the webcam functioned off the main board and the shaft encoders and DC motors functioned through the I/O board, one task of integrating the software was to establish an order of initialization processes so that each operation could complete without conflict with another.

6.6.8.2 Threaded Operation

It was originally desired to have the webcam constantly take images and update the coordinates while the entire navigation software ran in the background as a thread. When the motor control function ran as a thread, the navigation software alone would function properly. Unfortunately when it was attempted to run the navigation software as a thread, timing issues arose that kept the navigation software from controlling the motors quickly enough and ended with the realization that integration of the two software packages would need to be accomplished in a different manner.

6.6.8.3 Non-Threaded Operation

Without being able to run the navigation software in the background, the top level software required significant changes. Instead of allowing the vision software to constantly run, operation was changed so that the vision system was given time to take and process an image and then hand control over to the navigation software. The navigation software was adjusted so that once it reached its desired coordinates, it would hand control back over to the top level software. The top level software continued to hand off control in this manner so that each particular software package could perform its task and then return control to the main program. This manner of execution seemed to work very well when tested and required less overall processing since there were less functions running simultaneously. In order to accommodate this execution, additional global variables were created to keep track of where the robot was in its overall progress and to ease communication between vision and navigation software.

If for some reason the vision system was unable to acquire targets after image processing, a fail-safe method of operation was programmed. This fail-safe method directed the robot to sweep the field in order to cover all possible locations of the balls, raise the drawbridge, and then return to the starting square. Although designed only as backup plan, this form of operation would have proved adequate after seeing the methods which other robots in competition used.

6.6.9 Ball Retrieval Code/Limit Switches

When ball retrieval was switched over to being controlled by a DC motor, this simplified the manner in which the cable of the drawbridge would be released and retracted. In the overall scheme of the software, once the robot reached the half-court mark, the robot was instructed to lower the drawbridge. This software simply sends a signal to the I/O board which in turn sends adequate voltage to the pins the DC motor is connected to. The motor is energized and will continue to operate until the drawbridge
triggers the limit switch. When the limit switch is triggered, a signal is sent to the I/O board which in turn is understood by the software to mean that the motor should be switched off. When the robot has completed its task of retrieving the balls and attempts to return home, a similar operation occurs except that the software commands the motor to turn in the opposite direction in order to retract the drawbridge.

6.7 Troubleshooting

Unfortunately, the development of the Vol Retriever was hampered by what seemed like constant hardware problems. The computer system as a whole was very sensitive, and would frequently crash and refuse to reboot. Generally the system would start back up eventually, but several hours were lost in bringing the system back up. The hard drives were also incredibly sensitive to vibration and heat. Although the robot was handled carefully, several hard drives failed.

The team traveled to the IEEE competition in Fort Lauderdale, Florida on April 7, 2005. The team qualified, however a computer hardware failure prevented the team from competing in the competition. Eventually the problem was traced to a burnt pin on the computer board’s IDE port. After replacing this board, the robot has been much more reliable. The Vol Retriever has completed several successful mock runs in the lab.

7.0 Conclusion

Although the Vol Retriever was unable to compete in the IEEE competition, the team feels that the project is an overall success. The robot has successfully completed several mock runs, each with a run time of less than one minute. The project was a very challenging capstone to the college careers of each member. Each member of the team gained a great deal of experience in robotics, project management, and teamwork. The team was able to develop a robot that can serve as a research platform for future students and hopefully be useful to the department for many years to come.
Figure 22: The Vol Retriever with ball retrieval mechanism removed
Appendix A

Competition Rules
Student Program
Hardware Design Rules

All questions dealing with the hardware competition should be directed to:

General Chair: Dr. Eric S. Ackerman
3301 College Avenue
Ft. Lauderdale, FL 33314
(954) 262-2063 - 1-800-986-2247 ext 2063
(954) 262-3915 Fax
esa@nsu.nova.edu

**********************************************************************
SoutheastCon 2005 Hardware Competition Rules
Hosted by: Nova Southeastern University

Robo Knights

Overview:

SoutheastCon 2005 provides a challenging game of robotics skill. Each team must successfully pickup five small metallic balls located in random positions on the opponents side of the playing field and bring it back to your side and then return with the balls to your original starting square. In the process they must avoid the opponent's vehicle which is trying to do the same thing. Each team will
have 5 minutes to get the opponents metallic balls. The metallic balls will be different for each team. Five will be matte black (black oxide plating) and five will be matte silver (clear zinc plating). Each team that registers will receive 10 (5 of each) free balls and one 1/2 inch washer. New balls will be provided at the competition.

This competition is open to undergraduate teams and graduate teams separately. Undergraduate students will compete only with undergraduate students and the same for graduate students. If possible one final round will be played that is made of the first place winner for undergraduate versus the first place winner of the graduate teams. Teams can only bring one robot per level. One for the undergraduate team and one for graduate team is applicable.

**NOTE: PLEASE DESIGN YOUR ROBOT TO TAKE INTO ACCOUNT VARIATIONS ON THE MEASUREMENTS OF THE COMPETITION TRACK. WE WILL DO OUR BEST TO ENSURE THAT ALL MEASUREMENTS ARE WITHIN 15% OF THE PUBLISHED COORDINATES.**

**DESIGN TO ADAPT!**

**The Playing Surface:**

The competition will take place on a single sheet of plywood 4' x 8'.

**The Playing Surface:**

The competition will take place on a single sheet of plywood 4' x 8'.

All items listed below can be purchased from Home Depot.
A single sheet of 3/4 inch 4 X 8 SANDEPLY
Home Depot# 454-559
3/4 inch Forstner Bit
Home Depot# 577-619

1 inch 3M Model 2090
Blue Painters Tape
Home Depot# 958-999
When your team registers you will receive the following:

Each team that registers will receive 10 (5 of each) free balls and one 1/2 inch washer.

Construction Steps:

NOTE: PLEASE DESIGN YOUR ROBOT TO TAKE INTO ACCOUNT VARIATIONS ON THE MEASUREMENTS OF THE COMPETITION TRACK. WE WILL DO OUR BEST TO ENSURE THAT ALL MEASUREMENTS ARE WITHIN 15% OF THE PUBLISHED COORDINATES.

DESIGN TO ADAPT!
1) Purchase all of the items above.
2) The plywood is already sanded so there is no need to do any prep work.
3) Paint one side of the plywood white.
4) Once the paint has dried then apply the blue painters tape based on the diagram below. Any place where BLACK is showing is where the blue painters tape is to be used. The white area shown below will be painted black.

NOTE: The majority of the surface of the playing board will be black.
5) Decide where the five holes are to be located on each side of the board. The location are random and be anywhere in the blue squares shown in the diagram below. The only rule is that no hole may be less then one washer from another. In the picture below the red dots represent the closest that another ball could be placed.
6) Stick 1/2 inch flat washer (with a 1-1/4" O.D." - (1) included in the package of balls when you register) in each of the locations (You will have to purchase 9 more of them or just move the one around as you paint). The washer will cover the white paint which will end up being the white ring around each hole.

7) Paint over the blue painters tape and washers using the flat black paint.

8) Before the paint is fully dried remove the blue painters tape and washers. Your board should look something like this (the location of the holes will not be the same).
Note: The sides are symmetric: having similarity in size, shape, and relative position of corresponding parts.

9) With the 3/4 inch Forstner Bit drill out the centers for each ball location. In the example above the locations to drill out are in RED below. Drill completely through the plywood and then attach using glue, tape, etc. a piece of BLACK construction paper to the underside of the plywood at each drilled location.
10) The holes in the board should be deburred with sandpaper and that the inside walls of the holes be painted black.

11) At this point your board is complete except for the hole for the LED. Purchase two (one for each side) High-Output Infrared LED Brand: RadioShack Catalog #: 276-143

Specifications for the 276-143C are as follows:
- Radiant Intensity (mw) 7.368
• Peak Wavelength (nm) 940
• Forward Voltage (V) 1.28
• Half View Angle (Degrees) 36.36

Drill the hole so that the top of the LED is flush with the plywood. The wires should come out under the plywood and then be connected to the switch & battery. The hole for the LED should be in the center of the 6 X 6 starting square (3 X 3 location).

12) At the competition the playing surface will be laying flat on a carpeted floor. The practice area has pergo floors.

The Vehicle

The vehicle must fit in a 6" x 6" square, may not be taller than 8", cannot weigh more than 20 pounds and cannot change in size **SEE FAQ #16. It must be self-propelled, autonomous and may not be remotely controlled in any manner. It cannot contain any flammable liquids, gases, or explosives. The vehicles cannot project any objects either in the playing field or out of the playing field, and all parts of the vehicle must remain attached (i.e. the vehicle may not split into two separate pieces). The vehicle may not present any danger to the judges, the spectators, or the playing board.

Playing Rules

When called, the team will have 30 seconds to place their vehicle in the starting square and wait for the IR signal to be received. Once the signal is sent (steady on) the timer will start, and the vehicle will have a maximum of 5 minutes to get and bring back the metallic balls placed on the opponent's side of the track. The vehicle should aim to return to the starting square (see Scoring, below). When the vehicle has completed this mission, it should illuminate a blue LED at the top of the robot, which will have been designated by the team, to the judge, prior to the start of the competition, as the finish indicator.

Once this indicator is illuminated, a vehicle receives no further points for picking up balls or returning to the starting square. The game will end when both vehicles have enabled their finish indicator or 5 minutes have expired, whichever happens first.

Once the first round has begun and the playing surface has been exposed (viewable) then no team is allowed to work on their robot (hands off). The only exception is for charging of batteries.

Judging and Scoring

In matters of scoring and judging, the judge's word is always final. Scores will be determined by:

1) If you correctly get all of the balls and return to the correct spot.
2) The amount of time it takes.

The final score will be the total of the three rounds. No balls may be pre-loaded on the
robot. The balls can only be on the playing board.

Scoring: 20 points for each ball picked up (from the opponent's side).
If you mistakenly pick up an opponent's ball (on your side), you lose 20 points, and your opponent gains 20 points. If you fail to return to your base before five minutes expires, you lose 50 points.

If you correctly pickup and then return to the designated spot your team would receive 100 points for that round.

The team with the greatest number of points, summed over all three rounds, is the winner. In the event of a tie, the team with the smaller total time elapsed over the three rounds will be declared the winner. Total time for a round is measured beginning with the emitting of the "go" signal, and ending with the illumination of the blue finish indicator. If the indicator does not light, the round is measured as lasting 5 minutes.

Tournament Format

One qualification round (no other cars will be on the track) will be done prior to starting the competition. Qualification will take place on one of the practice tracks prior to the competition starting. To qualify to compete in the competition a car must:

Move off the start square when the go signal is sent and pick up at least one correct ball in the qualification round.

The competition consists of three rounds and one final championship round for the 1st and 2nd place teams. Each team will have three rounds in which to accumulate their final score. Each team will be given a random number and will be matched up against another random team. Each team will complete the game once then each team will go again and once more to total three rounds. At the end of the three rounds, the scores will be calculated and a 1st, 2nd, and 3rd position will be announced. Third place will be then awarded and the 1st and 2nd place teams will then compete once again during the awards dinner for one final round which will determine the final 1st and 2nd place winners. We will have two large projection screens showing the school's logo's and live video of the event.

To ensure a fair game the three playing boards will not be displayed until the start of each round. Three different playing boards will be on hand during the competition and once the call for hands off robots has been announced then the board for that round will be revealed.

Both sides of the playing surface will have the same locations of the balls.

Between each round there will be a fifteen minute break.

Two practice boards will be available to teams starting Friday morning.

Traditional Film, Digital Cameras, and Digital/Analog Video cameras (with FLASH)
will be allowed in the spectator area. All designs should take this into account.

**Additional Awards will be given for:**

The most creative design will be one which implements strategy, shows originality, and one which displays a high level of engineering skill.

1) Additional awards should be given for:
   a) Best original hardware design
   b) Best mechanical design
   c) Best use of custom designed chips (LSI, VLSI)
   d) Best power system
   e) Best ball retrieval system

Dedric Carter will be the main judge for the SEC05 hardware event.

Dedric Carter is an experience practitioner in advanced networks design/development, and the application of emerging technologies. He has participated in robotics competitions at MIT and served as a co-sponsor/VIP of the 2004 MIT ACM Artificial Intelligence/Gaming Competition. His early research pursuits were in human intelligence and machine learning and migrated to advanced network design and development specifically with technologies such as radio frequency identification. He has guest lectured or discussed technology topics at universities and conferences throughout the US and abroad including MIT, the City University of New York, and UVA. He has appeared on multiple occasions on the NBC Digital Edge technology showcase during his tenure as the youngest Director of the CGI-AMS Center for Advanced Technologies.

Mr. Carter has performed research at MIT Lincoln Labs and AT&T Bell Laboratories. He is currently a member of the Board of Trustees of MIT where he sits on the first Engineering Systems Division Board of Visitors. He was the David M. Adler Thesis Award winner from the MIT Department of Electrical Engineering and Computer Science in addition to receiving nomination to the Sigma Xi Scientific Research Honor Society where he is an Associate Member. Mr. Carter is a member of IEEE and ACM. He has a B.S. in Electrical Engineering and Computer Science and a Master of Engineering in Electrical Engineering and Computer Science from MIT. Mr. Carter is a 2005 Candidate for the PhD in Information Systems at Nova Southeastern University.

**FAQ's**

Answers are in **BLUE**

1) Issues dealing with what happens when two cars collide? *If two robots collide they will have 5 seconds to clear themselves. If they do not then they will be separated at the judge's discretion. The clock will not be stopped.*

2) What happens if a ball is dropped? *You can pick it up. (SEE FAQ#64)*
3) What happens if the opponent by mistake picks up your ball(s)? They lose 20 points you gain 20 points.

4) What happens if the opponent's car dies in your path? Can you push it out of the way? You can move it away if possible, if not after 10 seconds the judges will move any car that died and allow you to continue to play.

5) What happens if the opponent's car dies on top of one of the balls you need? The judges would move the dead car out of the way.

6) Do we rotate the playing sides? Yes and the color of balls

7) How close do we allow team members/visitors to be to the playing board? Everyone will be at least 4 feet from the board's edge.

8) How do we handle two teams that both complete the objective at the exact same time and accuracy level? If a tie (time and accuracy) then another match will take place.

9) Do we have a preliminary round to qualify which cars should be matched? No.

10) Is more than one undergraduate team per school allowed? Yes but only one robot can compete so pick the best one.

11) Does the robot itself have to remain completely inside the playing field? By this I mean, if, say, half the robot extends past the edge of the plywood but does not contact the floor and is able to recover without making contact with anything but the plywood playing surface, is this allowed or would it be a violation of the rules? Allowed

12) In the Judging and Scoring section, paragraph two, it says, "No balls may be pre-loaded on the robot." Could your please clarify this point? Does this mean we may not collect all five balls before returning to the start area? This means that you cannot start the competition with balls in your robot since the goal is for you to pick all of them up. If balls are found within the robot the team will be disqualified.

13) When you state in the Overview, paragraph one, that the robot must "return with the balls to you original starting square," must the robot be completely inside the square to be considered returned? Or must it be X percent inside the starting square? The robot should fit 100% into the starting square to be considered returned and come to a complete stop.

14) Can cars taken in and then expel a ball without penalty? No

15) Do cars incur a 40 point differential penalty by ingesting the wrong color ball or 20 points by ingesting and 20 points for having it on the wrong side at the end of the contest? 20 points by ingesting and 20 points for having it on the wrong side at the end of the contest

16) You say the robot cannot change shape, but in order to pick up the balls with a
lever, crane, belt drive or similar devise the robot will have to change somewhat. The robot must fit into the 6x6 square before starting. If you require some lever, crane, belt, etc. then it must be connected to the robot and if extended it cannot exceed 3 inches from any side of the robot. If your team will be using a device like this then it will be measured prior to competition starting.

FOLLOWUP QUESTION:

I read FAQ #16 for the 2005 SoutheastCon 2005 Hardware competition and have a question. I read that any type of ball retrieval device when extended "cannot exceed 3 inches from any side of the robot". I would like to know if when the device is extended, may it expand from the "base" of the robot on one or more sides, or is it limited to one side of the robot? In other words, can the size of the robot (originally max. 6"x6") expand 3" on multiple sides to render a dimension up to 12" wide, or can it only expand 3" on one side resulting in a 6"x9" robot? Correct it can only expand 3" on one side resulting in a 6"x9" robot.

17) I read QA 16, and it answers allot. Just to clarify: The robot can have an arm or similar devise that extend 3 inch outside the 6 x 6 box. This arm, or similar devise can rotate, move up/down or whatever so long as it stays attached and stay within the 3 inches. My confusion is this, if we get there and the tires of our robot rotate in such a way that the orientation of the rim of the wheel changes, are we disqualified. I believe what you might mean is that the general shape (about 90% of the robot) cannot change and it must stay within the 6 x 6 box, excluding the arm. Is this correct? Yes

18) Can we use an external camera outside the playing board to transmit to the robot? No

19) So each side of the board will have either black oxide or zinc covered balls, Yes or will the type of balls be mixed on both sides; No Also is it going to be possible that one type of ball be on both sides possible? No Also what happens if the opponent rolls over or moves one of the balls we a supposed to get? If the ball is picked up for any reason then see FAQ #3. Am I allowed to knock the opponents balls off the table purposefully or accidentally? No

20) Will each robot need to accommodate a wireless camera? (There was talk about this at SECON 2004). This is a possibility but we have not decided yes or no at this point. Either way it would be optional.

21) The two white squares (not the starting/finishing squares) in the center of each side of the playing field are for what? They are there to help you navigate the playing field.

22) In the rules you have mentioned that we can't work on our robots during the break. Why is that, why we can't work on it? i.e change batteries etc.? You can work on your robot as long as no playing surface is visible. Once one of the boards is exposed then it will become hands off. The only exception will be to change batteries. This is being done to prevent any team from mapping the playing field and then simply giving positions (matrix) of where the balls are. The game requires active searching and the
judges will ensure that all teams play fairly.

23) The rules also mention about cameras being used. We think that the cameras shouldn't be allowed, because it can be used as a remote control for the robot. All robots will be examined closely to prevent this. No team and school would want to be disqualified from this and future IEEE events. Certain electronic test equipment will be used to watch the room as well.

24) Specifically, what types of lights will be lighting the rooms we will be playing in? You should design for various lighting scenarios (overhead, spotlights, outdoor windows, etc.)

25) In last year competition, we know that some students got their robot designed from outside companies, how will you catch those guys? Teams should design and build their own robots, however all teams get some level of outside help. You can argue that buying an evaluation board, or using a PDA to control the robot did not require you to build your own controller. Some teams have team members that have excellent mechanical engineering abilities and facilities. No team should "buy" a robot or have faculty or outside people design or build it for them. IEEE Code of Ethics

26) Will we be allowed to modify or fix our robot during the break.? Yes but not during any active competition round. Once a round starts then it's hands off.

27) Is there white paint around the plywood's edge? Not specified, the edge (the 3/4" wide surface perpendicular to the playing surface) could be white, black, or just left alone as plain wood.

28) Is the dividing line created using a single strip of the tape?

\[ \text{Two strips of tape 1" side-by-side} \]

29) How long must the infrared LEDs remain on? The LED's will stay on throughout the 5 minutes so that robots can use it as a "parking" signal.

30) Both LEDs should come on at the same time. Will one switch and one battery drive both LEDs in parallel? If not, how will they be synchronized? Yes both LED's will be connected to one switch. The LED's will get power from a battery or a fixed power source.

FOLLOW UP: The "Playing Surface" instructions say that the start LED will be connected directly to a 1.5 V battery. But that makes the diode current (and hence luminous intensity) extremely sensitive to the type and state of charge of the battery. (Note that the LED has a rated (or maybe typical) forward voltage of 1.28 V) It would be better to state the drive current for the LED so that the luminous intensity can be fixed. I suggest to increase the voltage (two 1.5V cells in series, or maybe one 9V battery), then insert a biasing resistor that produces a current of about 10 mA (or
whatever is the rated current -- which doesn't seem to be specified in the RadioShack documents.) - 10 mA is about the right drive current (I am 95% sure that is appropriate), then I recommend to use a 9V battery and a 680 ohm resistor.

31) I didn't state FAQ #27 very well. I meant to ask whether there will be a white border on the playing surface. The diagrams seem to show a white border at the periphery.

Yes - http://sec05.nova.edu/hardware_design_construction.htm - Item #4 - Only at the edge but not within the playing surface. 1" tape should be used around the entire board. The 1" tape when removed should expose a 1" white band.

32) Do the "no hole" zones extend 1 inch around the white waypoint squares? Yes

33) As in FAQ# 32, is the boundary 1 inch from the 6 inch home area? Yes

34) Where exactly will the IR signal be coming from? What height? What coordinates on the playing field, or will it be on the playing field at all? Will the object sending the IR signal be a physical obstacle on the playing field? The only IR signal that we will be controlling will be the two LED's that are used to indicate "go". Please see the construction URL showing the exact location of the LED's.

35) Will members of our team be the ones positioning the robot in its starting square at the start of each round? Yes (only two people allowed)

36) Is there a penalty for hitting the opponents ball, but not knocking it off the playing field? If so, explain. Yes, the balls will not be flush they are slightly sunken in the board. You can roll over the hole and never touch the ball. If you pick up the ball then you lose points.

37) Must the balls be dropped inside the starting square, or can they stay inside the robot once you have returned to the starting square? All of the balls must remain in the robot.

38) Define "picked up"? Must the balls be literally picked up off the surface of the board? Or merely returned to the inside of the square by any means (dragged, for instance), as long as the robot still conforms to the no-shape-change constraints? The balls need to be lifted so that they do not touch the playing surface anymore. If your design picks them up and keeps them clear of the board you are okay.

39) In FAQ 14, you say "Can cars taken in and then expel a target without penalty? No" Would you define "take in" a ball? Moving partially or completely over the ball to detect it's type would not be considered "taking in" a target, right? To clarify, should "taking in" a target be defined as moving the target from it's seat in its hole? Yes, moving the target from it's seat in its hole is not allowed.

40) The bag of targets reads "1/2". Is that the diameter of the targets (so that the targets fall inside the holes in the field?, or the radius? (so that the targets rest on top of the holes in the field)? The targets fall within the washer used to draw the white ring.
41) The rules state "the vehicle may not present any danger to the judges, the spectators, or the playing board", but what happens if the robots harm each other? Is this allowed? If not, will there be penalties for any accidental robot interaction (if the robots are entangled or knocked over, for instance)? The robots should not intentionally harm each other. If they do get entangled or knocked over by accident one of the judge will decide if the round is still playable or needs to be replayed. If still playable they might just pick up the robot and return it to its last location on the board. Try to design into the robot the ability for it to detect if it has been moved or turned over.

42) If one or more of the robots gets physically tipped over due to robot interaction, will the judges set the robots back on their "feet", so to speak? Or will the robots be removed from the board? Or will there be a rematch? See FAQ #41

43) Why are the center-squares such an odd size? To distinguish them from the starting square.

44) If another robot picks up your ball by mistake and you receive 20 pts, must you still retrieve that ball to get all possible points? Is 100 the max points for each round? No if that ball is lost, you now only need to pick up remaining balls. We will be paying close attention to teams that pick up the wrong ball. If a robot does not have any control and is randomly picking up balls then it will be disqualified.

45) What is the penalty for bumping/moving a ball? The same as if you picked it up? If the ball is removed from its holding spot on the board you lose 20 points the opponent gains 20 points. Cameras that are located in the ceiling will be monitoring each hole.

46) Is each of the three rounds played against the same robot, or will a robot oppose 3 different robots? You will be playing against multiple robots.

47) From what I understand we will be assigned a target color for each round of the competition. How far may we interact in telling our robot what color balls to go for. For example, will the judges give off some sort of signal indication of the color balls our robot must fetch or are we allowed to tell the robot via some hardware or software switch before each round? Good Question - Please have one switch on the top of your robot that will control which type of ball you will be responsible to pick up. A simple toggle switch with a blue handle would be great.

48) If you could please clarify question 16, I must have missed something when I read it earlier. Is it correct that the 3" arm must be inside the robot, the 6" square, when the competition starts? How about when it ends? The entire robot, including an arm or lever, must fit in the starting square at start and finish, correct? Yes

49) Assuming robots A and B are competing on the field, is it absolutely guaranteed that the 5 targets to be collected by robot A are on the side of the field where robot B's home square is located? Is it absolutely guaranteed that those 5 targets are the ONLY 5 targets present on that side of the field? Yes - you should only collect the balls on the other side.
50) From the detailed robot competition playing field schematic, the home square is shown to be 5.88" X 6in". From the drawing I find it unclear whether the 5.88" is measured from the edge of the plywood, or from the edge of the white border on the plywood. If the measurement is from the edge of the plywood, how is a 6"x6" robot to fit 100% inside a 6"x5.88" square? The square is approximately 6 X 6. As long as your robot is 6 X 6 you are fine.

51) Is the 21.22" measurement from the side of the playing field to the center square of the playing field made from the edge of the plywood? Do I interpret the drawing correctly in thinking that the center square is 3.97" wide? If the answer to both of those questions is yes, why is the sum of those widths (21.22"x2 + 3.97") not 48"? If the center square is 3.97" wide and the 21.22" measurement is from the border and not the edge of the plywood, why then is the sum of the widths (21.22"x2 + 2" + 3.97") not approximately 48"? Please take a look at the Example images from Western Kentucky University.

52) The rules specify that once the team is called we have 30 seconds to place the robot in position and wait for the IR signal. Question: At what point do we know which balls we are using (ie. the silver or the black)? Is this known before the clock starts or are we told during the 30 second setup time frame? Good question. Each team can either 1) just know based on the distance traveled or 2) have a simple toggle switch for silver/black.

53) The rules specify that we have to go and retrieve the 5 balls from the opponent side and return them to our side. Question: Does every ball have to be returned independently, or can we obtain all 5 balls and bring them all back an once? Once they are returned, are they dropped or released in some type of compartment/container? You have to collect all five balls and then return.

54) When making the measurements for the playing field, what measurement is used for the width of the 1 inch 3M Model 2090 blue painters tape? Is it one inch, 15/16", or some other measurement? It is the width of the 3M paint tape.

55) Can the white circles around the holes extend into the “no hole” zones? No

56) Is the no-hole margin on either side of the middle line also 1"? Yes

57) In the tournament format section you stated that both sides of the playing field will have the same location of the ball could you please clarify? On the web the location of the balls are not a mirror image of the opposite side of the field. There seems to be a mirror image but it is turned or not position properly. Are the holes going to be symmetrical or random on either side of the field? They will be symmetrical. The drawing is not perfect but the board will be as close as possible. Symmetric - having similarity in size, shape, and relative position of corresponding parts

58) I am still unclear as to which part of the robot can actually extend out 3" and also which sides of the robot can extend. For example, is it a legal operation for the entire body of the robot to go UP three inches, thus having the dimensions 6"x6"x11"? It seems right now, that as long as the
extension is only 3 inches on ANY one side of the robot only, it will be legal. And I would also like clarification as to what extends, i.e. if it has to be some sort of device that picks up the balls only, or if it can be any "appendage", or even the entire robot, itself? 1) The robot may not expand upwards. The three inches is measured as a projection to the playing surface. No part of the robot should be greater then 8" from the playing surface. 2) If an appendage will be used it must have some ball retrieval function.

59) In the answer to question 58 you say that "If an appendage will be used it must have some ball retrieval function." Could this include a line detection system? Yes

60) When the robot returns to the starting square does it have to be in the same orientation when it started? No

61) Can the balls be dragged for any length of time whatsoever. In other words if the balls were only dragged during the time they were being picked up on the opponents side of the playing field and then lifted before the vehicle crossed the white line, would it be O.K? Not allowed - The balls need to be picked up while the robot is over the hole which holds the balls. - SEE FAQ #64

62) Will the playing field have the opponent's targets during the qualifying round? In other words, will there be balls on both sides during that round? Yes

63) Does the finish LED have to be blue? For example, If I have a RED led on my robot, can I tell the judge to watch for that before the round starts, instead of a blue one? Red is much more visible than blue to the human eye. Yes

64) This FAQ clarifies earlier responses for FAQ 2 and FAQ 61.

Definition of dropped ball: If a ball is lifted/moved from its seat, clearing the surface of the board and then regains contact with the surface of the board, the ball is considered dropped. Reasonable judgment will be used in the determination of a balls status as dropped or not. A dropped ball may be recovered as long as no excessive dragging or herding of the ball occurs. Intentional dropping of balls as a strategy is discouraged and may be penalized if the judges rule as such.

Definition of dragging: If a ball is lifted/moved from its seat and pushed with some intent along the surface of the board, the ball is considered dragged. Reasonable judgment will be used in the determination of a balls status as dragged. For instance, in the event that the ball makes slight contact with the surface of the board after being moved from its seat in route to being collected by the robot without intentional pushing for extended time along the surface or herding of one or more balls, the ball may be deemed not dragged by the judge. There is no specified period that it is acceptable to push the ball along the surface of the board as this time should be very minimal if at all. Robots should be design to acquire the balls with minimal contact of the ball with the surface of the board from the balls seated position to avoid potential dragging penalties.
In general, herding balls in an area is penalized; however, we do not want to unduly penalize a robot that in an honest attempt to collect a ball, drops the ball and is able to recover the ball.

65) Can one robot intentionally try to “confuse” another robot? For example, can white circles be painted on a robot to try to “confuse” the other team? **No**

66) How much time does each robot have to get to the other side of the board? If one robot remains on the side of the board where you are retrieving balls, will this robot be moved off of the board after a certain period of time? Or if the robot remains on the wrong side of the board and is retrieving your targets? **See FAQ #3 & 4**

67) I don't think you are understanding my question. Sure, you get points if he successfully picks up the balls, but what if he doesn't? If he doesn't then he may just be moving around on your side "lost". In this case he could be interfering with my ability to pick up balls. I don't mean to harp on this point but avoidance becomes very difficult when two robots are trying to track the same target. I am just considering whether or not to take this into account in my design. **You should take this into consideration but if a robot is just wandering lost then the judge has the right to pick up that robot and remove it from the playing surface.**