IEEE Southeast Conference 2008 Hardware Competition – Team: TKOG

W. P. Markiewicz III, Team Leader, *IEEE*, C. C. Nuckols, *IEEE*, D. W. Hensley, *IEEE* and T. A. Ridgell, *IEEE*

Abstract-- An autonomous robot was designed to compete in the 2008 IEEE Southeast Conference Hardware Competition. The robot, R3D3, was constructed to search, retrieve, store, and return four (4) out of the seven (7) possible painted wooden blocks to R3D3**'s starting position** within six (6) minutes. The purpose of the project was to compete against other IEEE Southeast colleges. Therefore, the competitive part consisted of building a robot that would perform the task more efficiently than the other robots in the competition. The project was constructed by developing two independent systems: the drive/navigational system and the retrieval/storage system. After the integration, testing, and debugging of the two systems, the robot was able to consistently pickup blocks and return to the starting position within six minutes. **Team TKOG's r**obot, R3D3, tied for 5th place out of 41 competing teams at the SoutheastCon 2008 Hardware Competition, creating positive publicity for The Citadel Department of Electrical and Computer Engineering.

Index Terms—Color, IEEE Student Activities, Image Color Analysis, Mobile Robots, Object Detection, Robot Sensing Systems, Robots, Servo Motors, Sonar Measurement, Space Vehicles

I. Introduction

THIS document will provide details on how R3D3 (Fig. 1) was built and used to compete in the competition. It will also discuss the competition and the strategy that was used to maximize the chances of winning. R3D3 was able to collect four blocks and return to the starting position within six (6) minutes.

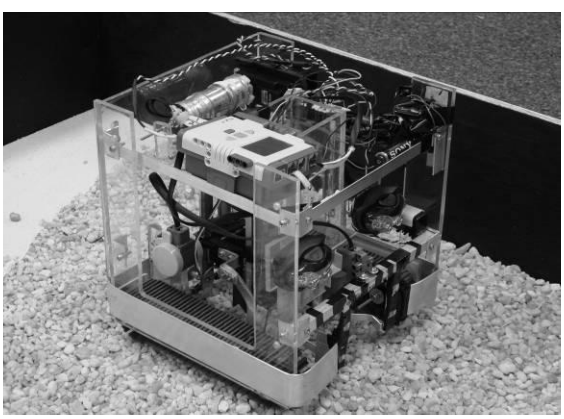

Fig. 1 - R3D3, team TKOG's robot used in the 2008 IEEE Southeast Conference Hardware Competition

II. Competition

In order to compete in the 2008 IEEE Southeast Conference Hardware Competition there were certain criteria the robots had to meet.

A. Robot Design Parameters

The robot had to operate completely autonomously once started and be entirely self contained, including any power source. The robot had to fit inside a 10¼ inch by 10¼ inch by 11¼ inch box to qualify for the competition. The maximum size of the robot was 10 inches by 10 inches by 11 inches tall. This maximum size applied to the robot when it was in its home base at the start of the match or was in motion on the playing field. When not in motion, the robot could have extended a maximum of six inches by six inches in any one direction at a time. The extension had to be physically connected to the main robot at all times. There was no weight limit or construction material restriction. Each robot had to have a bumper that surrounded a minimum of 80% of its perimeter. This bumper had to be the outermost structure at all times when the robot was moving. The bumper had to present a vertical surface at least 1" high and cover, at a minimum, the space from 1 ½ to 2 ½ inches above the playing field. The bumper could be of any shape around the robot and need not be outwardly convex on all surfaces but could not have any radius of curvature less than ½ an inch. The bumper had to be included in the maximum 10 by 10 inch by 11 inch tall overall size. The robot had to have a button or switch somewhere on its top surface to start the robot in play.

B. Playing Field Parameters

The playing field (Fig. 2) was based on a 6-foot by 6-foot plywood deck surrounded by walls that extended 8-inches above the playing field surface. The walls were attached to the outside edge of the plywood deck. The playing field was divided into three zones plus robot home bases. Zone 1 was painted BLUE, except for the home bases, which were painted WHITE; Zone 2 was covered with sandy WHITE paint; Zone

This work was supported in part by The Citadel's Electrical and Computer Engineering Department.

D.W. Hensley (dwhensley@gmail.com).
W.P. Markiewicz III (wlater.mizark@gmail.com).
C.C. Nuckols (ccnuckols@gmail.com)
T.A. Ridgell (ridgellmmcs@yahoo.com)

3 was covered with pea pebbles. The pea pebbles were bonded to the plywood base with tile adhesive. The playing field walls were painted flat BLACK. One navigation aid was placed in each of the playing field home base corners. Each navigation aid consisted of one quarter section of an 8-inch-tall, 4.5-inch-diameter, schedule 40 PVC cylinder into which six LEDs were placed. The navigation aid on the right (as viewed from the robot home base side of the playing field) flashed its LEDs at 4.0 kilohertz, while the navigation aid on the left flashed its LEDs at 2.5 kilohertz. Both navigation beacons used a square waveform and both were painted BLACK. The 12 1/2-inch-square robot home bases were located in separate corners on the same side of the playing field. The playing field contained seven wooden two inch cube blocks. The layout pattern had bilateral symmetry with respect to a line dividing the playing field in half between the robot bases. Each block had a round; passive RFID tag attached to one surface and was painted according to its point value. Two blocks were painted RED and were worth 15 points each, two were painted WHITE and worth 20 points each, two were painted BLUE and worth 25 points each, and the last block was painted BLACK and worth 30 points. The RFID tags were not painted. Each RFID tag was programmed with its own identification number and was positioned to face the front (home base side) of the playing field. The playing field was well lit.

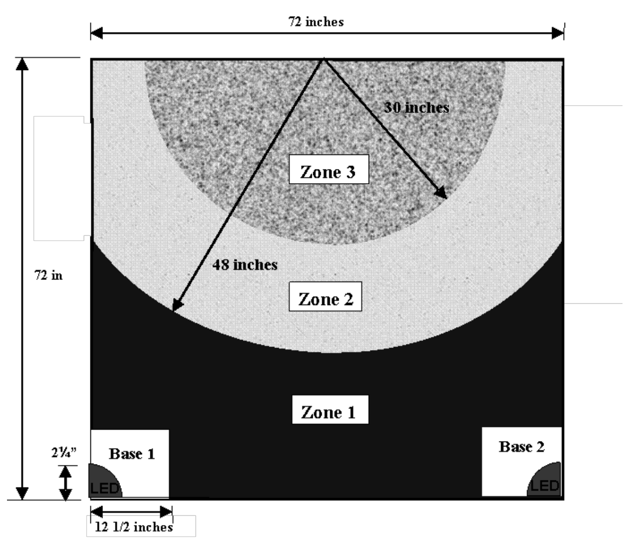

Fig. 2 - Playing Field

C. Scoring

Points were scored by placing blocks in the robot's home base. "In the robot's home base" was interpreted to mean that any portion of a block that was within the boundary of the home base square. If a block was placed wholly or partially on another block in the home base square, it would count as being in the home base square. Ten points were awarded to a robot that completely left the starting square. If a robot carried blocks into any part of the home square and remained in the home square at the end of play, all the blocks it carried, up to the maximum number allowed, would score. The first two rounds of play were preliminary rounds. During these rounds, a block would only score if placed in the robot's home square. After the two preliminary rounds, the top eight scoring robots of both preliminary rounds combined moved on to a single elimination playoff bracket. These playoff rounds had two robots on the field at a time, competing head to head. During the playoff rounds, if a block was placed in the opposing team's home square its value would score points for the owner of that square.

Hoarding Penalty: If more than four blocks were placed in home base, points would only be scored for the four lowest value blocks. During the playoff rounds, any blocks in excess of the four lowest scoring blocks were scored for the opposing team.

Blocking Penalty: In the playoff rounds, a robot could not persistently position itself near the opposing team's home base so as to block the other robot from returning home. If this situation occurred, the blocked robot scored all the blocks it possessed or deposited in the vicinity of its home base. A stalled (doesn't move for one minute), blocking robot was removed from the playing field.

Poaching Penalty: Disturbing or removing blocks from an opposing team's home base, whether intentional or not, did not decrease the score of the "poached". This action WAS considered grounds for disqualification. The disqualified robots were removed and play continued.

Extension Penalty: Extending a portion of the robot beyond the bumper while the robot was in motion resulted in a penalty of one point per second of violation.

III. Strategy

We divided the playing field up into four, three foot square sections. Dead reckoning placed the robot into the center of the first quadrant. The robot then completed a 360 degree turn while using a camera to locate any blocks within an eighteen inch radius. When a block was detected, the camera navigated the Drive Assembly to align and drive R3D3 over the block. As the block was driven over, it would trip a control switch when in position to be collected. This switch did two things: 1) caused R3D3's LEGO NXT Brick to record a capture, and 2) triggered the Parallax BS2 to activate the Gripper and Elevator Assembly to collect and store the block. R3D3 would then back up to the center of the quadrant and continue this process of looking for and collecting blocks until the 360 degree turn was completed. Then R3D3 would drive forward to the center of the next quadrant. This entire process of searching one quadrant and then the next was then repeated until either four blocks were collected or four and a half minutes had elapsed. The only difference being a 450 degree effective turn in each quadrant after the first was used instead of a 360 degree effective turn. This ensured that R3D3 searched the quadrants in a clockwise or counter clockwise manner. If R3D3 started in the left base, all of the turns would be clockwise. If R3D3 started in the left base, all the turns would be counter clockwise.

IV. Drive/Navigation

The purpose of the Drive/Navigation system was to ensure R3D3 was able to move about the playing field to positions that were advantageous to acquiring the blocks.

Fig. 3 – NXT Intelligent Brick used for drive and navigational purposes

A. Drive Assembly

The Drive Assembly consisted of two LEGO Interactive Servo Motors driving Vex Robotics Tank Treads. The Motors were controlled by the LEGO NXT Brick which allowed for accurate control of the robot. The drive motors were positioned two and one-half inches from right and left sides of the Robot. This allowed the drive/idle gears and the Vex Robotic Tank Treads to be placed at the outer edge of the Robot. This configuration ensured stability on any and all terrains.

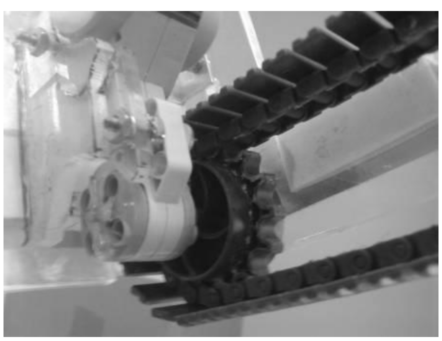

Fig. 4 – NXT motor driving Vex tank tread

B. Navigation Assembly

The Navigation portion of the system consisted of dead reckoning, a LEGO NXT Camera from mindsensors.com, a LEGO ultrasonic distance sensor, and a LEGO Compass. Dead reckoning was used to put R3D3 in the center of the next quadrant to be searched. The NXT cam was used to find and navigate to the blocks. The camera was positioned an inch and a half above the playing field. A physical mask was placed over the bottom half and top third of the lens to keep the camera from picking up colors from above the playing field and from the playing field itself. The US sensor was used to determine when a block was being driven over. The compass was used to navigate back to home base.

Fig. 5 - Vision Subsystem for NXT (NXT Cam) from Mindsensors.com, used for navigation.

V. Retrieval/Storage

The purpose of the Retrieval/Storage system was to ensure R3D3 was able to pickup and securely store the blocks removed from the playing board.

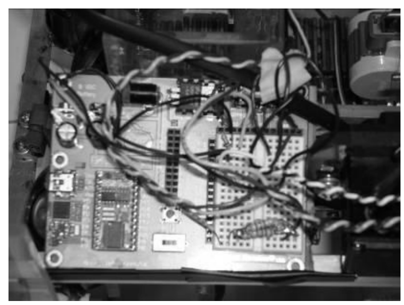

Fig. 6 - Parallax BS2 (with wiring shown) used to power and control the retrieval system.

A. Retrieval Assembly

The Retrieval Assembly consisted of a Gripper and Elevator. The Gripper utilized a Parallax (Futaba) 180 degree Rotation Servo connected to a Lynxmotion Little Gripper. The Elevator utilized a pulley system powered by the HiTEC HS-805BB Mega Torque 1/4 Scale Servo which raised and lowered the Gripper.

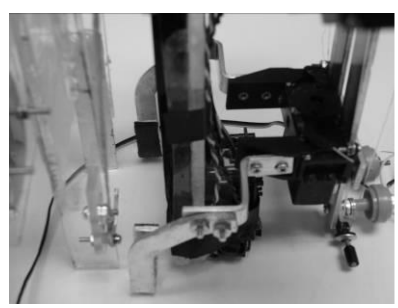

Fig. 7 – The Gripper, shown positioned at the bottom of the elevator and "Tower" systems, ready for block retrieval.

B. Storage Assembly

The Storage Assembly consisted of a two and one quarter inch square silo, we dubbed “The Tower”. The Tower held the blocks above the playing field by the use of a mechanical latching device. The Tower also housed the LEGO Camera and the control switch.

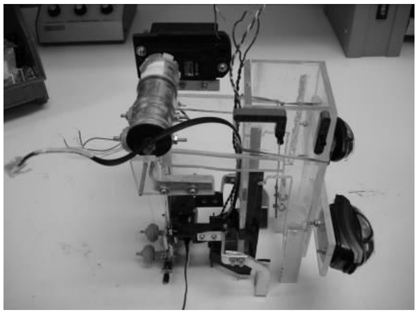

Fig. 8 – The Tower: Rectangular silo used for storing blocks. Gripper and elevator also shown.

VI. CODING

R3D3 had two independent systems, the Drive/Navigation system and the Retrieval/Storage system. Each system had its own independent microcontroller, the Lego NXT Intelligent Brick and the Parallax Basic Stamp 2, respectively. The NXT was programmed using the NXT-G drag-and-drop block programming language (similar to LabVIEW) and the BS2 was programmed using the BASIC 2.0 language. Both systems made use of the switch.

B. Lego NXT Intelligent Brick Coding

The NXT was coded to have a MAIN program routine with five subroutines running in parallel (Fig. 9). These subroutines consisted of a BLOCK subroutine, a BUTTON subroutine, a HOME TIMER subroutine, a GoHOME subroutine, and a MOTORS subroutine.

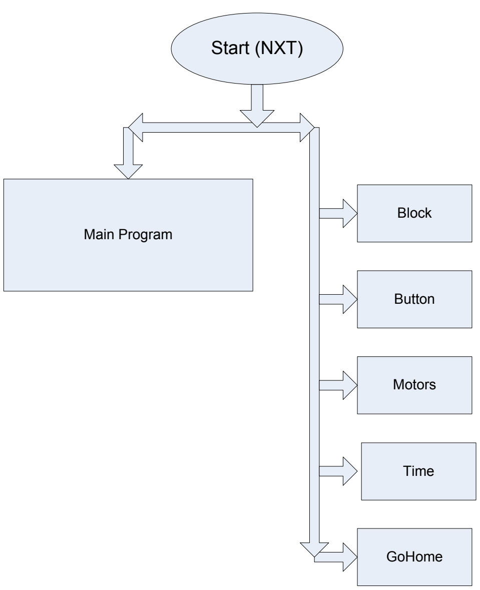

Fig. 9 – NXT Top Level Flow Chart

When the start button was pushed, the MAIN program routine (Fig. 10) had R3D3 drive to the middle of the first and closest quadrant. It would then enter a loop turning clockwise if it had started in the left base or counter clockwise if it had started in the right base, and using the camera to detect any blocks in the vicinity. If a block was detected, the program would enter a loop to have R3D3 use the NXT cam to position it’s self so that the opening on its face was in line with the block. This same loop would then have R3D3 drive to the block, and ultimately over the block, using the NXT cam to keep the opening in line with the block as it drives. Once R3D3 had begun driving over the block, the US sensor would detect a block is being driven over and the BLOCK subroutine would take over control while the block is retrieved. Once the BLOCK subroutine had relinquished control back to the MAIN program, R3D3 would back up to the center of the quadrant and continue searching for any other blocks in that quadrant. Once a 360 degree rotation had been made, determined by a loop counter and dead reckoning, the MAIN program had R3D3 drive forward to the center of the next quadrant. The program would then enter the loop described above with the only difference being an effective turn of 540 degrees would be used. This allowed R3D3 to move from quadrant to quadrant in a progressive search around the playing field.

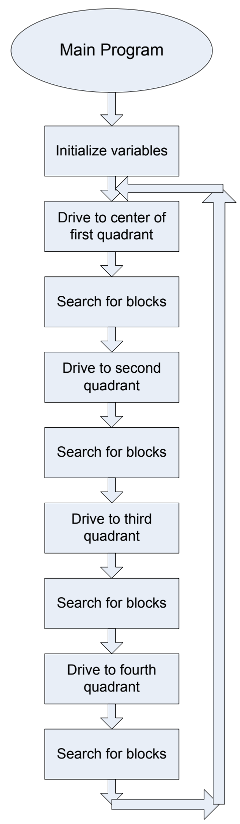

Fig. 10 – NXT Main Program Flow Chart

The BLOCK subroutine (Fig. 11) was a loop that constantly monitored the US sensor. If the US sensor registered a reading less than 9 cm, meaning a block was underneath it, then the subroutine would stop the motors and ensure the block was straight by turning R3D3 to the left 25 degrees. This subroutine would then make three attempts to drive forward so that the block was positioned between the gripper arms for retrieval. If R3D3 was successful in positioning the block between the arms, the switch would be physically closed by the block, activating the BUTTON subroutine, which would take over control. Once the BUTTON subroutine gave control back to the BLOCK subroutine, the BLOCK subroutine would subsequently release control back to the MAIN program. If the three attempts did not yield a successful positioning, the BLOCK subroutine would have R3D3 back up and control would be relinquished to the MAIN program. This would most likely have allowed R3D3 to reposition itself for a more successful attempt on the block, but was also a safeguard against something other than a block triggering the BLOCK subroutine.

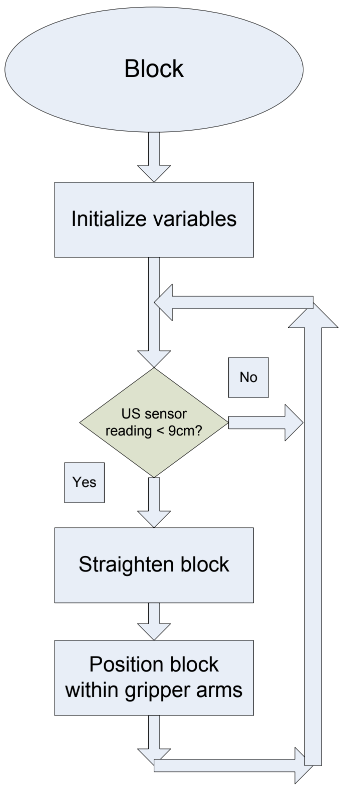

Fig. 11 – NXT BLOCK Subroutine Flow Chart

The BUTTON subroutine (Fig. 12) was another loop that constantly monitored the switch. If the switch was closed, this subroutine would stop the motors, increment the "blocks collected" counter, and simply keep R3D3 from moving for 5 seconds while the Parallax BS2 microcontroller collected and stored the block. It then checked to see if the "blocks collected" counter was equal to four. If it was, then the BUTTON subroutine set a Boolean value named "home" to true and then relinquished control. If not, control was relinquished with the "home" value remaining false. The "home" value was used by the GoHOME subroutine and will be explained later.

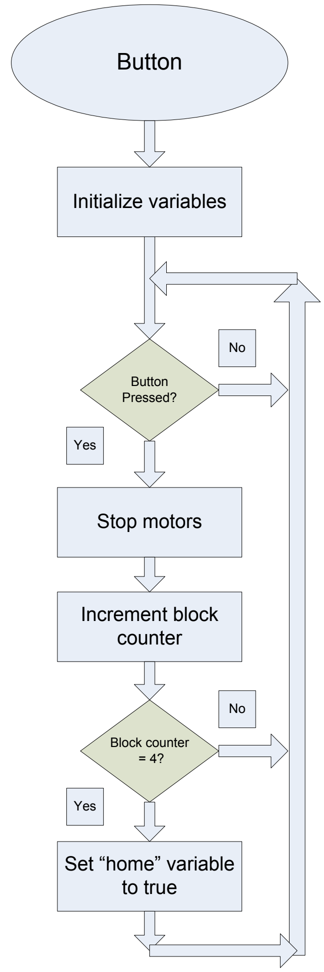

Fig. 12 – NXT BUTTON Subroutine Flow Chart

The TIME subroutine (Fig. 13) simply waited for four and a half minutes and then set the "home" value to true. The purpose of this subroutine was to let R3D3 know to begin driving to home base after four and a half minutes even if four blocks had not yet been collected. A minute and a half was found to be the worst case scenario amount of time needed to return home.

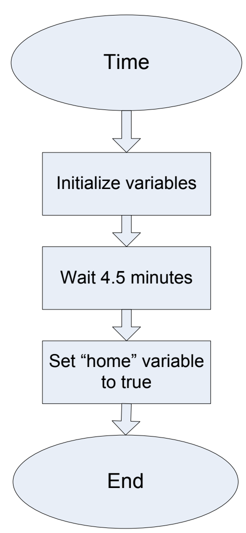

Fig. 13 – NXT TIME Subroutine Flow Chart

The GoHOME subroutine (Fig. 14) initialized the "home" Boolean value to false and then entered a loop that constantly monitored the "home" value. If the "home" value changed to true, the GoHOME subroutine would take over all control of R3D3 and drive to home base using the compass for navigation.

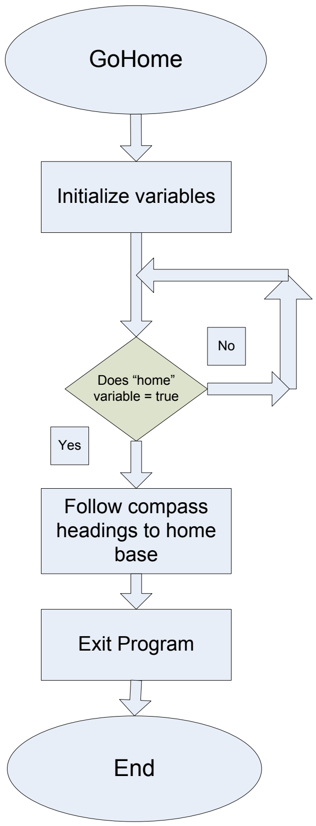

Fig. 14 – NXT GoHOME Subroutine Flow Chart

The MOTOR subroutine (Fig. 15) monitored the NXT motors. If at any time either of the motors became locked, this subroutine would take control and drive R3D3 in reverse for one second to free the track from whatever it was stuck on. This was accomplished by checking the total number of degrees that each motor had turned over the course of an entire run at two second intervals. If, at any time, the number of degrees was equal to the check two seconds before it, this subroutine would activate the reverse drive for one second. This subroutine would go on pause during the BUTTON subroutine only, allowing R3D3 to be stationary while a block was collected. This subroutine proved extremely useful on the rocky terrain.

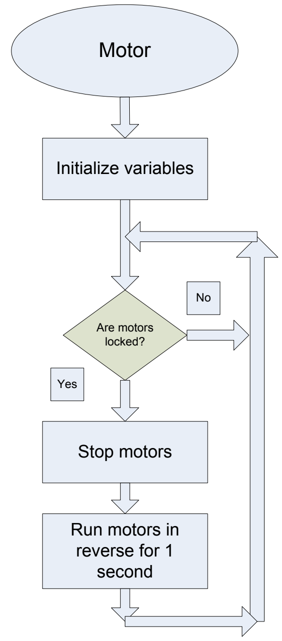

Fig. 15 – NXT MOTOR Subroutine Flow Chart

B. Parallax Basic Stamp 2 Coding

The BS2 had only one program routine (Fig. 16). This routine constantly monitored the switch. If the switch was closed, this routine would close the gripper, rotate the elevator servo lifting the gripper and block into the tower, open the gripper leaving the block in the tower, and then rotate the elevator servo in the opposite direction to lower the gripper back to the waiting position.

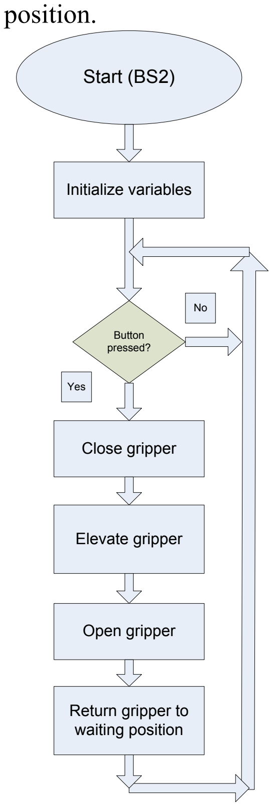

Fig. 16 – BS2 Flow Chart

VII. Acknowledgment

The authors wish to appreciatively acknowledge Dr. Skinner, who was our project advisor, Dr. Jerse and Dr. McKinney, who both gave advice at the beginning of the project, Dr. Peeples for his continuous support as our instructor in ELEC 422-81 Senior Design, Mr. D. Crawford for the many tools and materials that he provided for R3D3 and the construction of the simulated "moon surface" playing board, The University of Alabama at Huntsville, who hosted the 2008 IEEE Southeast Conference, and The Citadel Foundation for their generous financial contribution.

VIII. Reference

Books:

[1] Parallax, Basic Stamp Manual Version 1.9. Rocklin, 1998.

Periodicals:

[1] IEEE Southeastcon, 2008. Available http://ewh.ieee.org/reg/3/secon/08/competition.html

IX. Biographies

Derek W. Hensley was born in Charleston, South Carolina on September 29, 1984. He will be graduating from The Citadel with a Bachelor of Science Degree in Electrical Engineering in May 08.His employment experiences include Superior Electric Company, Orbital Engineering, and CRG Engineering. He also achieved the rank of EIT and is a member of IEEE. His special fields of interest include Power Systems.

Walter P. Markiewicz III, E.I.T, was born in New York on April 21, 1982. He will graduate Magna Cum Laude from The Citadel in May 2008, with a Bachelor of Science Degree in Electrical Engineering. Upon graduation, he will begin working for Albemarle in Orangeburg, South Carolina. He is a member of IEEE, as well as a member of Tau Beta Pi, Phi Kappa Phi, and the Order of the Engineer. In his spare time he plays drum set and is currently working on several musical projects with a group of friends. After a break from schooling, he plans on pursuing a Master's Degree in Mechanical Engineering, as well as continuing his education in chemistry and neuroscience.

Christopher C. Nuckols was born on February 23, 1985 in Virginia. He will graduate from the Citadel with a Bachelor of Science Degree in Electrical Engineering in May 2008.

Timothy A. Ridgell, MMCS(SS)(retired), EIT, was born in Gainesville, Florida on November 24, 1958. He will graduate from the Citadel with a Bachelor of Science Degree in Electrical Engineering in May 2008. His previous employment experience was 26 years in United States Navy Submarine Force as a Non-Nuclear Machinist Mate. He served on five (5) submarines, USS Daniel Boone, USS Mariano G. Vallejo, USS Lewis and Clark, USS Rhode Island and USS Pennsylvania and retired on August 31, 2003. During the 26 years of his enlisted naval service, he obtained the rank of Senior Chief Petty Officer. He is a member of IEEE.