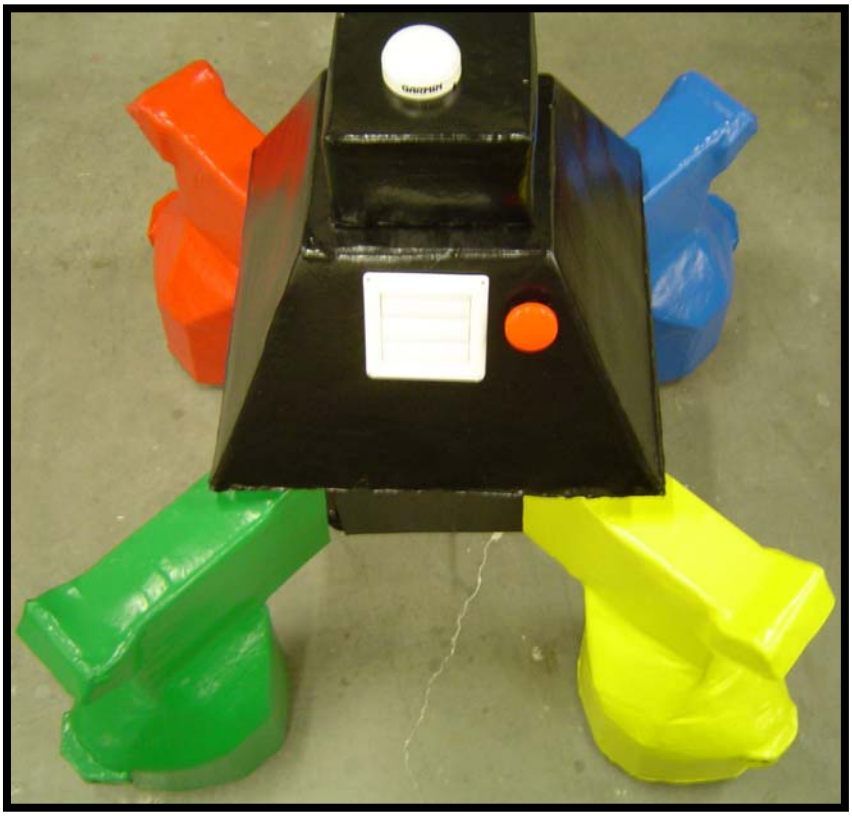
RDB3K



Gary Stein
Graduate Computer Engineering: Software Systems, Simulation, Sensors, Electronics
Geoff Decker
Undergraduate Engineering Technology/Design - CAD, Frame, Construction, Wiring, Safety
Gavin Barnes
Undergraduate Mechanical Engineering - Mechanical, Outer Cover
Chase Hansel
Undergraduate Mechanical Engineering - Mechanical, Outer Cover
Tim Roberts
Graduate Electrical Engineering: Funding, Advising

Faculty Statement:
I certify that the work done by all students on this project is consistent with a senior design course and that the vehicle has been significantly modified for this years competition.

Dr. Fernando Gonzalez
fgonzale@pegasus.cc.ucf.edu

1. Introduction

RDB3K (Research and Development Bot 3000) is a four wheeled autonomous ground vehicle that treats each of its four subsections as an independent system which was developed by the Robotics Lab at the University of Central Florida. Each subsection contains identical motor controllers, motors, machine vision, and sensors that are combined together at a central location to provide a 360 degree sensor panorama of the outside world. Position sensors are used to place the vehicle in a world map. Its vision and obstacle sensors are combined to gain an overall outlook on the current situation. The independent drive system allows for immediate motion in any direction and contains 18 degrees of freedom for reconfiguration of shape to deal with different environments. Each of the four subsystems contains a dedicated computer board to facilitate its vision, control, and sensing. All of the vehicle's complex systems including its vision and motor control are managed by a machine learning system that uses induction to make decisions by comparing the current situation to examples given by a human oracle. A voting system between the independent subsections provides the best current decision. RDB3K is a unique system that allows more complex motions over traditional systems and its use of machine learning provides greater flexibility over preprogrammed systems.

2. Team Organization

Consisting of four team members, the RDB3K team was lead by Gary Stein (Graduate in Computer Engineering), who also programmed and made all of the electrical systems for the robot. Geoff Decker (Undergraduate in Engineering Technology/Design) focused on the frame design, RDB3K's mechanical CAD, mounting and organization of sensors on the vehicle, wiring all the components together, and safety factors on the platform. Before making any decisions, though, he would report back to Gary who would make the final call. Chase Hansel (Undergraduate Mechanical Engineer) and Gavin Barnes (Undergraduate Mechanical Engineer) focused on the production of the RDB3K covers and would assist Geoff with any tasks he could not complete on his own. Their tasks were assigned by Gary in collaboration with Geoff. Timothy Roberts was head of funding for the group, and when funding was not immediately available, smaller expenses came out of personal funds. Due to the small size of the development team, tasks were divided on a job-to-job basis, depending on team availability and area of expertise. There would be a small goal each week to complete the larger goal by a certain time. Within a year, the total amount of time spent on the RDB3K project is 3730 hours. The break down of total hours each person committed to the project can be seen in table 2-1.

Team Member	Gary Stein	Geoff Decker	Chase Hansel	Gavin Barnes
Hours Devoted	1950	1312	246	222

Table 2-1: Team RDB3K Man Hours

3. Design

RDB3K was developed as a platform test-bed for the trial and testing of several research interests including indoor mapping and artificial intelligence. As a platform, the main function is to recognize obstacles and avoid them while mapping out the surrounding area. The emphasis of this vehicle is to overcome the issues with vehicle dynamics and control by creating additional degrees of freedom that will facilitate both vehicle movement and help sensors map the physical world. The system was designed to have vehicles sensor analysis and control governed by trained examples rather than preset rules.

3.1: Customer and Team Standards

According to the customer's requirements, team RDB3K developed a design which could meet the following: detect white and/or yellow painted lines, drive over an incline with a gradient of 15%, travel off road (sand, grass, dirt), fits within the specified size limitations, have a turning radius of five feet or less, avoid obstacles (barricades, barrels, buckets, pot holes), and travel less than five miles per hour. Based on these specifications, the team came up with a unique design plan on the general aspects of the vehicle:

1. The system's chassis must allow for omni-directional movement.
2. The system must allow for vision to see in all directions.
3. The system must be able to make reactive decisions based on local environment, yet store its global process.
4. The system must be more intelligent and less preprogrammed.
5. The system must have subsections that can operate independently.
6. The system must be different that others in the field.

Team RDB3K decided that by combining our standards with the customer's requirements would yield an innovative vehicle that would accomplish all of the goals.

3.2: Computer-Aided-Design

The second step of the design process was to use Computer-Aided-Design (CAD). A CAD schematic of the entire vehicle was created to simulate how the vehicle would work together. As the vehicles components changed, it was redesigned in CAD to make sure the new design is compatible with the rest of the system. The electrical system was also designed and tested in CAD for the PCB (Printed Circuit Boards). When the board needed to be changed, it was first bread boarded. Once it proved it could work, it was ordered. Individual components were then tested before system integration to make sure the consumer's requirements could be met with them.

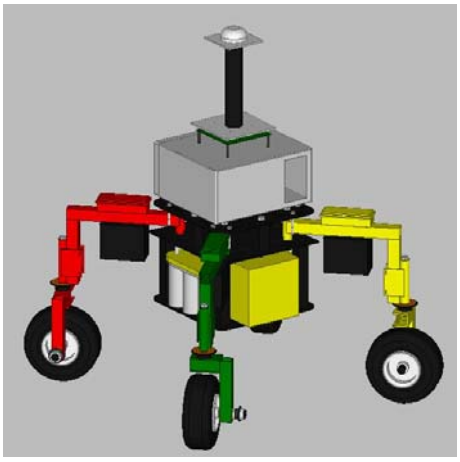


Illustration 1: Mechanical CAD, Early Iteration

3.3: Iterative Design

Team RDB3K used an iterative design process with an initial rapid prototype that would work but may not be ideal. After the first iteration was complete, basic testing with components was performed. If the team found that something did not work, they redesigned the platform so it would work while keeping the consumer's requirements in mind. As flaws were discovered from the original design, new parts were produced and iteratively changed out. An example would be our carbon-brush system. When the drive motors reached high amperage, the original plastic enclosures melted, causing the brushes to fall out of position. The team decided to make a new enclosure consisting of the same type of resin that one would use on fiberglass. This worked out very well; unlike the thin plastic that was used previously, the resin was able to handle the high temperatures that the brushes distributed.

3.4: Dynamics Simulation

The final step was to test for error conditions that might damage the physical frame by simulating the system dynamics before the frame was finalized. The idea behind this was to test physical movement based on real world parameters, including motor constants, torque, and friction. Once the simulation was programmed in Matlab, the results were displayed in both a two dimensional and three dimensional frames for easy visualization. After a final design was decided, the vehicle dynamics were modeled and tested to see if all the various movements were physically possible. The vision system was also tested within a simulation of stored results before it was used on the final system.

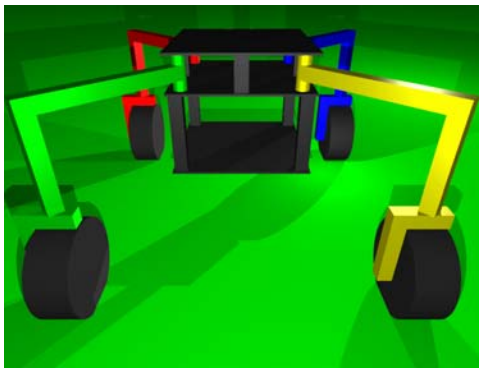


Illustration 2: Still Image from 3D Illustration

4. Vehicle Frame and Mounting

4.1: Platform Frame

4.1.1: Frame Material

Based on the omni-directional system developed for this platform, the team decided that the structural metal for RDB3K needed to be strong. The team felt that aluminum would dent and bend to easily, so the team opted to go with steel instead; 3/16 inch steel plate was used in conjunction with 1 1/2 inches wide box steal with 1/8 inch thick walls. Less material was also used because of steel's strength, whereas more would have been needed in the case of aluminum. For example, cutouts of the metal were made without losing the durability the platform needed. Once assembled, the platform's total weight is about 235 pounds, including payload, batteries, and all other components.

4.1.2: Layer Separation

RDB3K is split into three layers to allow the maximum amount of storage possible. The first section (bottom layer) holds the pair of batteries and payload. This section alone contains 1/3 of the vehicles total

weight. By making the bottom segment the heaviest, the vehicles center of gravity becomes as low as possible. It was designed so the batteries and the payload would be symmetrical, allowing for the center of gravity to remain centered.

The second section (middle layer) is attached to the bottom section by welds in four places. This layer holds the relay, fuse box, and 12 volt "hot plate." Easy access to these items is provided through the sides or the top layer of the platform through the cutouts. The leg system is also held in place through a bolt, bearing, and compression system which keeps the middle and top layers from shifting. This form of movement would cause misalignment.

Illustration 3: View of Basic Frame Layers

The third section (top layer) attaches to the middle layer through the compression system. This layer holds the computer and the sensor shelving. The computer is mounted on vibration dampeners to alleviate oscillation. The sensor shelving structure is then mounted to the computer, which holds the two compasses, joystick receiver, video transmitter, wireless E-Stop receiver, and the SiRF GPS receiver. The top layer is also where the leg positioning bolts are mounted, which keeps the legs within a certain degree of movement.

4.2: Consistency of Subsections

4.2.1: Legs

Each of RDB3K's four leg sections is consistent with each other. If any of the legs were assembled in a random order instead of the standard arrangement (red, blue, yellow, and green), the robot would still work and drive exactly the same. Team RDB3K made sure that all parts were interchangeable, so if one leg's part broke, the team would not have to manufacture a special part. The platform's steering motor, the Pittman GM14904S014 motor, is mounted vertically on the back side of the leg by two clamps to allow easy adjustment. The Unibrain Fire-I digital cameras are mounted at the leg's "knee." The MCU (Microcontroller Unit) boards are mounted in parallel with the H-Bridge box; the MCU is on top of the leg, while the H-Bridge is on the bottom. The forks are attached by a bearing system which uses a free axel located on the front of each leg. On top of the free axel is the home locator sensor system and mounted to the bottom of the leg is the carbon-brush system.

Illustration 4: Example of Leg Assembly

4.2.2: Fork

Like the legs, each fork was made consistent to allow the use of interchangeable parts. The top of the fork holds the free axel, which goes through the turning sprocket of the steering system. The power plate for the carbon-brush system is directly held against the sprocket. On the underside of the sprocket, the drive motor, a Pittman GM9236S024, is held onto the fork by a faceplate which is mounted on the side. At the bottom of the fork, a live axel is attached directly to the drive wheel. This system consists of a one inch diameter axel, two locking collars, two press fit bushings, the wheel, and drive sprocket.

4.3: Covers

Made out of fiberglass, there were five covers constructed for RDB3K. The first one is the pyramid structure which covers the top layer of RDB3K. The pyramid structure contains two 120mm fans for proper system ventilation, the Garmin GPS system, E-Stop button, and E-Stop LEDs. The four leg covers encompass the entire leg without hindering movement. About seven inches from the bottom of the leg cover is where the ultrasonic range finders mount. At the top of each leg cover, there is a hood to protect the Unibrain cameras from water and to help reduce random glare caused by sunlight. Overall, the purpose of these covers is to provide RDB3K with environmental protection.

Illustration 5: Environmental Covers

4.4: Frame Safety

Team RDB3K tried to take into account as much frame safety as possible. The covers provide protection from water to make sure the robot does not short circuit. They also cover up all power systems and chains. All edges were rounded on the frame to make sure there were no unnecessary cuts and to prevent the wire insulation from ripping. Mud flaps were also created to keep undesired dirt off the chains and electrical systems at the bottom of the forks. There were also a set of battery rails set up to hold the batteries in place so they would not topple over.

5. Power and Electrical

5.1: Batteries

To power its system, RDB3K is equipped with a pair of six volt Optima Yellow Top Deep Cycle batteries. The advantage of using two six volt batteries over one 12 volt battery is that it gives us the ability to maintain a low center of gravity. By offsetting the two six volt batteries to the left and right side of the robot, then connecting the two in parallel, the 12 volt system was achieved while maintaining balance. The batteries give out 55 amp hours, which allows our robot to operate, minimally, two hours per set.

5.2: Power Distribution

For each leg, RDB3K has a power distribution strip mounted to the bottom layer support columns. Each of these strips contains unregulated 12 volts (directly from the battery) for the H-Bridge and drive motor, and regulated five volts (from the computer power supply) for the home locator, and the MCU board. From this strip, all of the grounds run into the middle layer of the robot which contains the fuse box. Also located in the middle layer, the 12 volt distribution "hot plate" is used for distribution of unregulated 12 volts from the battery. There are two switches mounted in the middle layer of the robot to easily cut off power to the motors (black switch) and computer (gray switch).

5.3: Motor Control

5.3.1: Steering Motor

The steering motor is controlled by the MCU mounted to that particular leg using PWM (pulse width modulation). This signal is amplified by an H-Bridge circuit, which increases the current and allows the motor to spin in both directions. The H-Bridge circuit was developed in house using discrete components. The system's steering motor turns the free axel, mounted on the leg, and rotates the entire fork assembly.

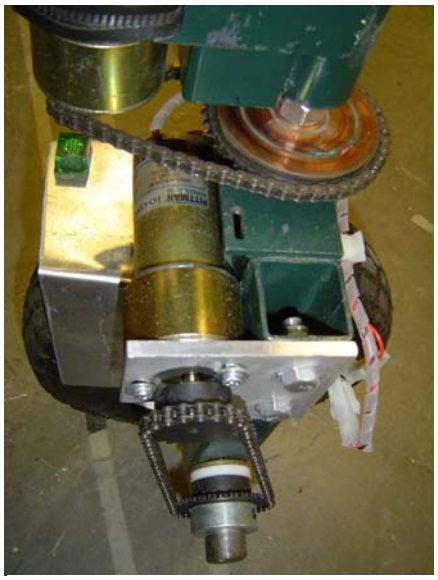


Illustration 6: Drive and Steering System

5.3.2: Drive Motor

To allow free rotation of the fork, without the restriction of wires, the drive motors are powered by a carbon-brush and copper plate system. The MCU sets off the relay, which connects power to the drive motor. The drive motor turns the live axel and causes movement in the direction of the fork.

5.4: Microcontroller Board

Through the use of Microchip PIC16F877A, the MCU board handles low level aspects of controlling the vehicle and reading from raw sensors. For example, the MCU would read the home locator and motor encoders to provide robust control.

5.5: Communication Board

The communication board provides a common ground source to all the MCUs and centralizes the communication between external components and the computer. Furthermore, it provides the regulated 12 volts from the DC/DC power supply to all sensors that need it. There are status LEDs located on top of the board which shows the direction in which each leg is going.

5.6: Electrical Safety

The concern for both team and spectator safety, along with reducing the risk of electronic discharge throughout RDB3K's components, made electrical safety a high concern. For example, our frame is not

grounded. So if any voltage source touches the metal, the frame will not electrify any sensors or persons. Team RDB3K also went through and made a color system for the wiring in order to reduce confusion and keep wiring consistent throughout the vehicle. For both the steering and drive motors, there are status lights which light up when a motor is active and turn off when it is idle. This lets the team know if a motor is trying to move when it should not, which prevents motor and H-Bridge burnout. The Optima batteries are a sealed gel cell battery which prevents any battery acid spill. On the front of the platform, there is also a battery life indicator which allows the team to see when the battery is fully charged, and to prevent a system failure due to lack of power.

6. Sensors and Computer System

6.1: MCU/Data Acquisition

The MCUs handle the data acquisition, operating in real-time, with interrupts to handle specific tasks. This means that while the MCU is controlling the two motors, they can take a quick break when new information comes in from sensors. For example, when the ultrasonic range finders see an obstacle, the MCU handles the raw signal and converts it into a serial data stream for the main computer. The reason why the MCU handles this process and not the computer is because the computer does not have the ability to process voltages directly. The MCU is needed to convert that signal for the computer. Refer to section 8 for greater detail.

Illustration 7: MCU Board

6.1.1: Steering Position

The steering position is done by using encoders and a home locator. The encoders are connected directly to the input shaft of the steering motor which the MCU interprets to find the direction and position of the motor. This feedback is used to maintain position and lets the computer know also when it is requested.

Due to the application of dead-reckoning from the encoder, a position sensor was added to increase accuracy over time when the steering assembly is in the home position. Over time, the encoder error accrues. To counteract this error, a mechanical switch, acting as a reference point, resets the steering position once it reaches the preset location.

6.1.2: Ultrasonic Range Finders

The ultra sonic range finder sends out a pulse signal and calculates the time it takes for the pulse to return in order to find the distance to an object. The signal spreads out in a 45 degree cone. Any surface within that cone, which the signal can deflect off, will be detected. Two range finders, the Daventech SRF08's, are used per leg mounted at height such that they can be used to detect objects within three feet of the vehicle. In addition, they are positioned so they will avoid each others signal and will be able to provide coverage for the blind

Illustration 8: Ultrasonic Range Finder

spots that are a result of the vision system.

6.1.3: E-Stop

The wireless E-Stop and the hard wired E-Stop button systems are wired in series such that if either is set off it immediately grounds the MCU interrupt pin. So, if one E-Stop system is set off, both are set off. The wireless system uses a Visonic VS-WT201 wireless transmitter with a Visonic VS-WR200 wireless single channel receiver, which provides a distance of 50 feet using a remote transmitter. The hard wired E-stop is a water sealed momentary ground button mounted on the outside of the cover. The E-stop system is connected to the communications board and a signal is provided to the MCU on each leg, causing an interrupt that immediately halts the running system.

6.2: Computer Side

6.2.1: Cameras

The cameras are Unibrain Fire-i Firewire cameras that capture 640x480 RGB at 15 frames per second. They are mounted one per leg, giving each leg its own perspective of the ground in front of it. Combining the four leg views together forms a panoramic view around the vehicle.

6.2.2: Navigation

Two digital compasses, a PNI TCM2 and a KVH C100, are used in tandem and filtered to produce the angle versus geographic north. These are both sent over serial to the main computer. Two GPS receivers were used in conjunction, a Garmin 16A and a SiRFstarIIe, and combined together to accurately track the position of the vehicle more than each could do alone. Both communicate over serial to the main computer.

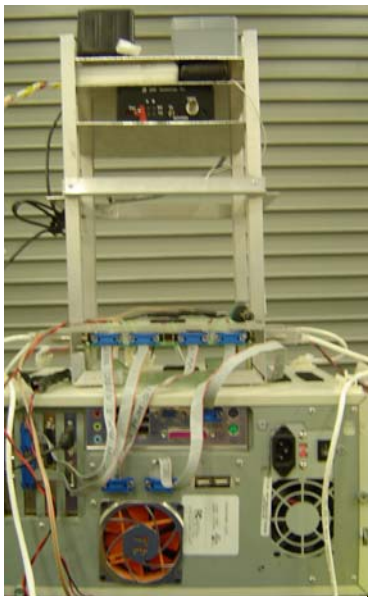


Illustration 9: Computer with Sensor Stack

6.2.3: Computer

The computer is a commodity FoxConn micro-ATX motherboard with a 2.0 Ghz Pentium IV processor with 512 MB of RAM. This system is running GNU/Linux with the Preemptive 2.6 kernel for stability and its ability to run many concurrent threads. Additional boards were purchased to provide additional fire wire buses, USB buses, and RS232 serial ports to allow communication with the rest of the components on the vehicle. A wireless joystick, the Logitech Freedom 2.4, was used for control and training of the vehicle.

6.2.4: Video Transmitter

A wireless video transmitter was used to provide output of the computer monitor information that is passed to a portable LCD screen. This allows the user to monitor the outputs of the computer system, like camera output and position information, without violating the competition rules by sending any

information to the computer during the run. This allows for remote debugging while the vehicle is running.

7. Software System

7.1: Software plan

The software strategy from the beginning was to create a modular system that had many completely independent programming blocks that do only a specific task and has a common format. In this fashion, blocks of the same type can be interchanged quickly through configuration files and will not affect any of the blocks that follow. This allows for quick assembly of blocks to create a full process. This flexibility allows for any changes in the platform like changing camera type from a USB camera to Firewire without major code changes.

7.2: Threading

In addition each processing block exists in its own thread which operates independently from the other threads and contains a queue of inputs to process. The processing block then passes on its information to the next processing block without knowledge of what the receiver is doing, and proceeds with the rest of its own inputs. In this fashion, the blocks all operate independently of each other such that any delay of one subsystem does not affect the rest. The blocks are split up based on the function in which it performs.

7.3: Types of Blocks

Input blocks read in input from outside devices. These would include camera reading blocks, serial reading blocks, and disk reading blocks. Each operates independently with blocking I/O to process all inputs. For example, to process information from the four cameras, four camera reading blocks are called with the appropriate setup of where the device is and where to output the data.

Preprocessing blocks take input from the Input blocks or from the output of Postprocessing blocks. They reformat or redirect the information to the appropriate location to use within the rest of the system. These would include image preprocessors, which break images up into smaller images, GPS preprocessors that convert degrees into meters, sonar preprocessors that convert time terms into meters, and ray preprocessors that convert the objects in the local map into a set of distance measurements.

Classifier blocks take the series of data produced by the Preprocessor blocks and make a decision as set of outputs. This is

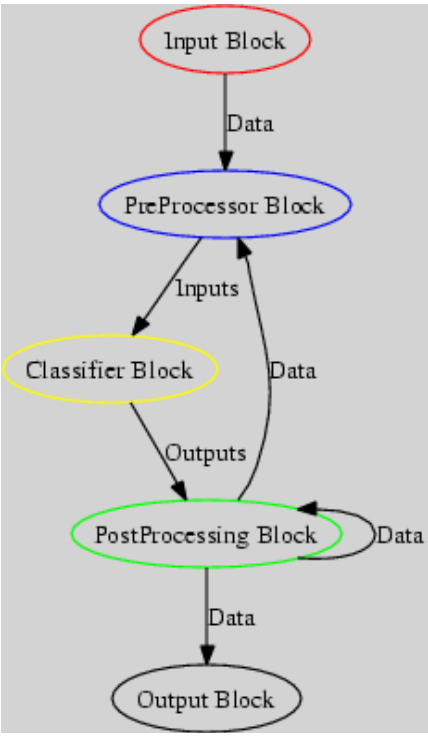


Figure 7-1: Basic Connection Between Programming Blocks

done mainly through an induction step, which can use any series of machine learning algorithms including neural networks, decision trees, or Bayesian classifiers. This step is done off-line by presenting a series of inputs and outputs to the classifier to attempt to teach it the proper relationship. It then builds its own mapping functions for use during normal on-line running. The two main functions of the classifier are to make image classification and driving decisions.

Postprocessing blocks take the outputs of the Classifier blocks or from another Postprocessor block. Its function is to convert those outputs into a more usable form or direct them to the proper place. These blocks include image postprocessors that outputs of the classification to build an image, local map postprocessors that takes the classified images to create a local map, global map postprocessors that take local maps to build a global map, and an MCU postprocessor that turns the driving outputs into commands understandable by the MCU.

Output blocks take the outputs of a postprocessor for output to a device. These would either write images to disk, display them to screen, or transmit them over serial lines.

7.4: Object Oriented Separation

This plan follows the Object Oriented paradigm for the separation of object into their individual tasks and performs data hiding since each block does not need to know what the others are doing. This allows for easy change out of any blocks and ability to rearrange them to perform some task without the need to reprogram the system. This ability allows the same objects to be used for the different competitions or testing with quick configuration changes using the same C++ code base.

7.5: Pipelined Data Processing

Because the blocks are separated into threads and because there is data hiding between blocks, a pipelining scheme can be implemented such that no time is wasted while waiting for inputs. For example, once a camera image is captured and passed to a preprocessor, it immediately attempts to capture then next image. In the mean time, it does not stop the processing of the GPS or any of the other cameras. The computer's operating system will properly schedule these concurrent jobs such that it is able to handle data much faster than doing any of these processes serially.

8. System Integration

8.1: System Organization

Following the design plan, computer jobs were split up into the high level and low level tasks. The high level tasks were handled by the main computer in the different blocks, while the low level processes were handled by the MCUs mounted to each individual leg. The high level processes would include vision processing, driving decisions, and visualization. Whereas the low level processes are the basic motor control and signal processing. These processes can be handled in parallel because each leg is independent from one another.

8.2: Communication Organization

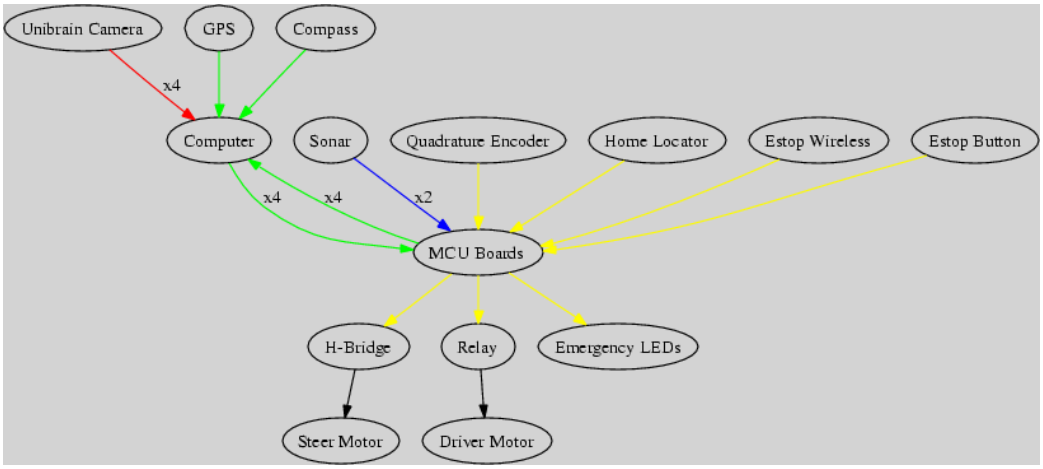


Figure 8-1: External Communication

Referring to figure 8-1, the cameras communicate to the computer over Firewire (red). The GPS, compass, and MCUs communicate to the main computer using standard serial connections (green). All input raw connections (yellow) connect through the MCU and are translated into serial data for the main computer. Furthermore, the MCU outputs a raw signal to the communication board to turn on the E-Stop LEDs. The ultrasonic range finders attach to the MCU using the communication and wiring protocol I^2C (blue) to digitally pass the result of its pulse signal. Finally, power signals (black) are transmitted by the H-Bridge and relay to their respective motors based on the input signal from the MCU.

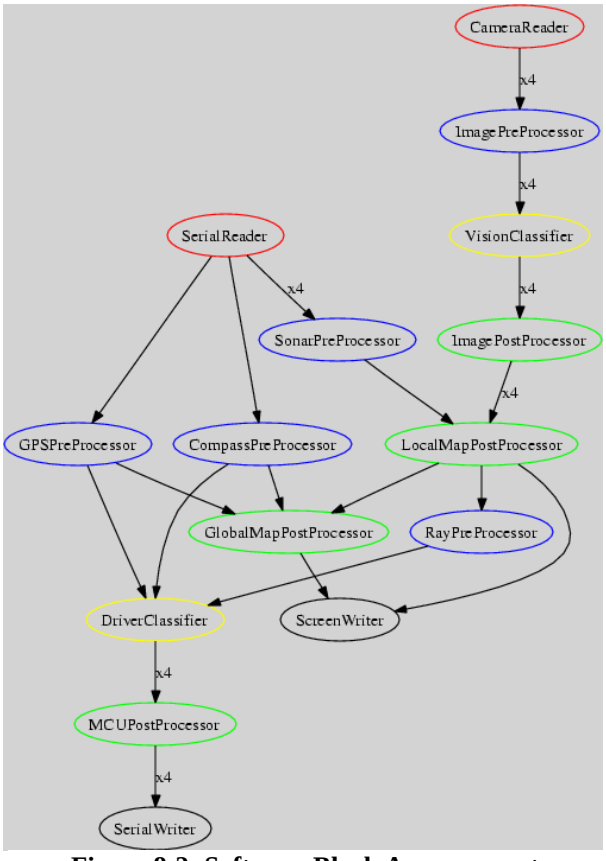


Figure 8-2: Software Block Arrangement

8.3: Procedure of Normal Operations

The standard configuration of the software blocks allows for the full operation of the vehicle is listed in Figure 8-2. Images are captured from the camera blocks and passed into the image preprocessing block. The preprocessing blocks split those images into smaller inputs that are then passed into the vision classifier. The vision classifier accepts those inputs and uses its learned mapping function to produce a set of vision outputs. Those outputs are then combined together by the image postprocessor to create a fully classified image.

That image information is then combined with the sonar information that was gained over the serial line from the MCU to create a local map in the local map postprocessor. That local map is then passed to the ray preprocessor which determines the distance of all obstacles and combines that information with the GPS and compass values for the driver classifiers. The driver classifiers then make a decision on the direction to travel based on its training for its particular leg. Once a consensus is reached, the decision of the direction to travel is passed to the MCU postprocessor, which formulates that direction into a command. That command is then passed to the actual MCU through the serial interface. Additionally, that classified local map is displayed to the screen for the user to see.

8.4: The Difference in Competitions

8.4.1: Autonomous Challenge

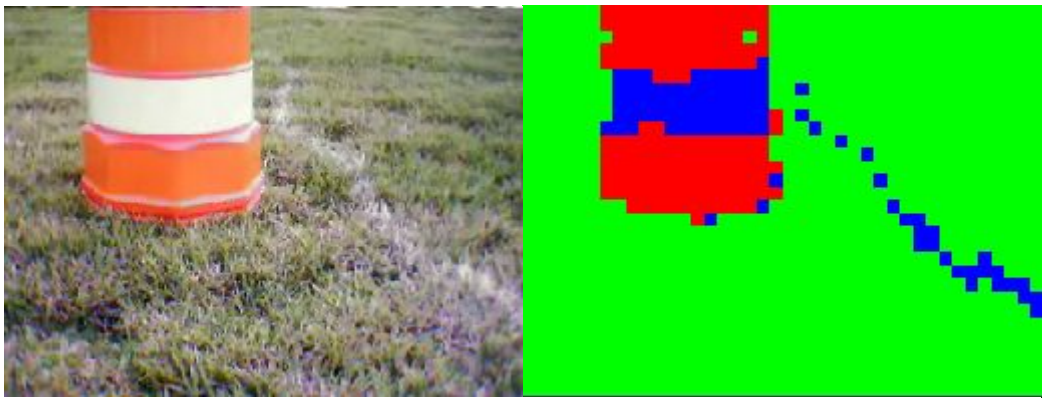


Illustration 10: Example of Classified Image from Original

The Autonomous Challenge integrates the procedure described above, with the addition of storing previous locations to back out of traps and dead ends. This allows the platform to realize where the vehicle has already been such that it moves further down the track in proper direction. Dashed lines are generally handled by the fact that the space between lines is smaller than the safety factor for the vehicle. The ramp and the sandpit are recognized as non-obstacles and traversed over as normal.

8.4.2: Navigation Challenge

The navigation challenge differs from the autonomous challenge by the fact that it uses the GPS data directly to navigate to the list of waypoints. In addition, by using a global map, RDB3K can preemptively avoid obstacles by remembering the obstacles previously passed and are now outside the local sensor range. Moreover, painted lines are not recognized as obstacles and are treated like grass.

9. Predicted Performance

9.1: Speed

The Pittman drive motors are geared to a 65:1 gear ratio and turn at 71 RPM. With a 10 inch wheel and a 4:1 gear reduction, the top speed of RDB3K is 1.6 mph. It takes about 5.3 seconds to rotate 360 degrees in place, and it takes about 1.3 seconds to spin a single wheel in place.

9.2: Ramp Climbing Ability

Based on the torque of the drive motor and the gearing between the motor and live axel system, the four motors combined has enough force to pull the entire weight of the vehicle up the ramp by a factor of two without stalling. In addition, the frame was constructed that it would be able to ride up an incline of 32 degrees, or a 65% grade, before bottoming out.

9.3: Reaction Time

Reaction time is a highlight of the RDB3K platform, due to the fact it was built to be a reactive system. The vehicle itself can stop within 1/10 of a second because the camera system is reading at 15 frames per second and the sonar is polled twice a second. The reactive nature of RDB3K is almost immediate. As far as maintaining speed and heading, the MCUs have a reaction time of 1/50 of a second. So, when the MCU needs to drive or turn a leg, it will perform the action almost immediately.

9.4: Battery Life

As stated before, the two six volt Optima Batteries provide 12 volts of power to the entire system. Minimal battery life expectancy is two hours, though RDB3K has run for four hours.

9.5: Detection of Objects

Based on a 30 degree camera angle and a 2.1 wide camera lens, the camera can detect obstacles nine feet away from the vehicle. The sonar can detect objects three ft away at a 45 degree angle.

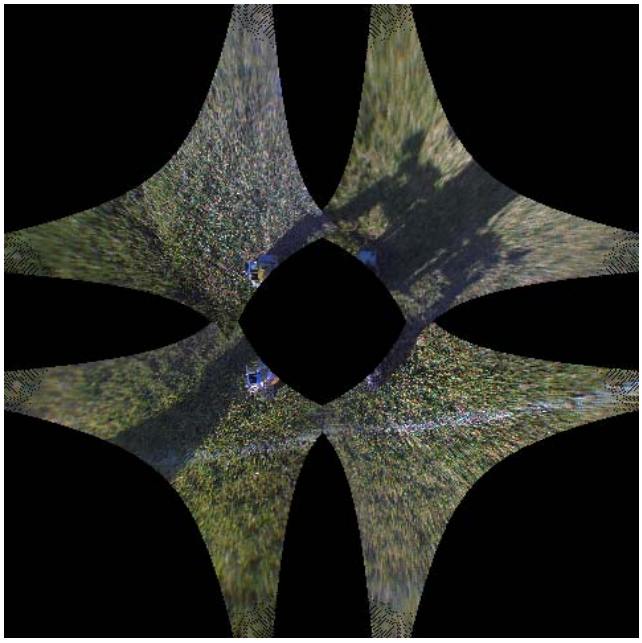


Illustration 11: Transformed Local Map Example (16 feet by 16 feet)

9.6: Dead Ends, Traps and Pot Holes

Because the system is reactive, it will initially enter into a dead end, however, once the scope of the dead end is found using the global map it will reverse and leave to a previously known location. Potholes are recognized by the vision system and seen as an obstacle the same way as a bucket or barrel would be seen, therefore, RDB3K will avoid potholes.

9.7: Accuracy of Arrival at Navigation Way Points

Based on a singular GPS, the accuracy is, without differential, is under three meters, but using dual GPS with filtering, accuracy is under a meter. Therefore, arrival at way points should not be a problem. Versus an independent accurate GPS system, our GPS system was 25 cm off.

10. Parts Cost

Part	Quantity	Cost to Lab	Actual Cost	Final Price
Battery	2	$135.00	$135.00	$270.00
Compass-KVH	1	$0.00	$795.00	$795.00
Compass-PNI	1	$0.00	$769.00	$769.00
Computer	1	$549.00	$549.00	$549.00
Computer Peripherals	1	$151.00	$151.00	$151.00
Covers	1	$256.00	$256.00	$256.00
Drive Motors	4	$105.00	$105.00	$420.00
Drive System (excluding motors)	4	$142.00	$142.00	$568.00
Frame Machine Work	1	$1,000.00	$1,000.00	$1,000.00
Frame Materials	1	$125.00	$125.00	$125.00
GPS-Garmin	1	$300.00	$300.00	$300.00
GPS-SIRF	1	$0.00	$200.00	$200.00
Misc Electrical Equipment (wires, fuses, etc.)	1	$180.00	$180.00	$180.00
Misc. Materials	1	$408.00	$408.00	$408.00
PCB Boards	5	$33.00	$33.00	$165.00
Steering Motors	4	$0.00	$120.00	$480.00
Steering System (excluding motors)	4	$75.25	$75.25	$301.00
Ultra-Sonic Range Finder	8	$15.00	$15.00	$120.00
Unibrain Fire-I Cameras	4	$120.00	$120.00	$480.00
Video Screen	1	$110.00	$110.00	$110.00
Video-transmitter	1	$79.99	$79.99	$79.99
Wireless E-Stop System	1	$0.00	$120.00	$120.00
Total Price				**$7,846.99**

11. Conclusion

Based on the requirements of the customer and design decisions made by the team, team RDB3K believes that the platform developed is innovative and satisfies the goals of the competition. The use of interchangeable parts and sensors reduces the expense and complexity that would have been necessitated on a traditional vehicle of this caliber. With omni-directional movement, coupled with panoramic vision, it is a creative step towards moving beyond conventional robotics. This ability to not only see in all directions, but to move in them as well, grants the capability to have a purely reactive system. Thus, RDB3K can make decisions, and act upon them, in a single step. Its use of induction, through training with a human, alleviates the constraints of a more rigid platform. Not only will it accomplish the IGVC tasks, but will also prove to be a successful test bed for further projects.