Design Report for the White Russian Unmanned Ground Vehicle

York College of Pennsylvania

Paul Kletzli, Ryan Muzii, Robert Carlson, TJ Hartlaub,
Peter Zientara, Travis Eichelberger, James Brown, and Daniel Chamberlin

Dr. Patrick Martin (pjmartin@ycp.edu), Dr. Kala Meah (kmeah@ycp.edu)

ABSTRACT

York College is please to present White Russian to the IGV Competition this year. Team members in charge of White Russian were to create a robot from scratch with only the part carrying over being the backup LIDAR. A tractor design was selected for the drive system while 8020 framing was chosen to make for easy assembly. The electrical system was divided into a high power system and low power system to separate the motor control from powering the other systems. Vision decided to use two cameras to give our robot a wider view for detecting white lines. The navigation team used multiple programs to direct the robot so each program only had one specific task. All of these innovations were used to make White Russian as seen in Figure 1.

Figure 1 - The completed picture of White Russian

INTRODUCTION

Team Organization

White Russian was created by eight undergraduate engineering students and two faculty advisors from York College of Pennsylvania. The eight students were divided between four sub-teams that included a mechanical team, a power team, a sensor team, and a navigation team. Table 1 shows the breakdown of which students focused on each portion of the robot.

Table 1- The layout of the team that created White Russian

Faculty Advisors Dr. Patrick Martin and Dr. Kala Meah			
Project Contact Paul Kletzli (ME)			
Mechanical Team	Power Team	Sensor Team	Navigation Team
Paul Kletzli (ME) Ryan Muzii (ME)	Terry Hartlaub (EE) Robert Carlson (EE)	James Brown (CE) Daniel Chamberlin (EE)	Peter Zientara (CE) Travis Eichelberger (CE)

Design Process and Integration

The design, integration, and fabrication was broken up into two semesters: summer of 2013 for design, and spring of 2014 for integration and fabrication. During the summer of 2013 our professor introduced three milestones spaced out across the summer to see how well our work was coming along. These milestones are simply a demonstration to the team and professors of the progress from design to implementation of the White Russian. These milestones not only helped keep progress moving along but helped with communication with different sub-teams working together.

Design Innovations

Since there are innovations to almost every sub-team, a brief overview will be discussed in this section, while each respective components system will go into more detail. A tractor design was selected for the drive system because continuous and zero-point turning. For the suspension system, a rocker bar was used to dampen the effects of the terrain to the image the cameras capture. A special triple mount was made to hold both of the cameras while it also holds the GPS. This mount has positions for the camera in 5 degree increments so the optimal position can be found and used.

The electrical system was divided into high and low power systems. High power system was used to power and run the motors with two 12 Volt sealed lead acid batteries. The low power system runs all of the sensors and on board computer with a different 12 Volt sealed lead acid battery. Our low power team implemented custom designed circuitry that allowed debugging the robot easier and user friendly. For example, an automatic switching circuit is used to switch between external and local power supplies internally as well as charge its local supply automatically. Battery monitoring circuits and emergency stop LED indicators were also implemented. The battery monitoring circuit allows battery life to be observed with a push of a button and LED indicators confirm the status of emergency stops.

The sensor systems main goal is to quickly and accurately provide information regarding the environment to the navigation team. Therefore, sensor systems were implemented in C++. C++ allows for lower level connections to the sensors, which provides increased communication speed. There was significant effort to improve the hardware being used in order to increase the accuracy of the data. More sensors were also implemented to broaden the range of information.The improved hardware allows for simplified data acquisition algorithms which makes the system more reliable and simplified.

The navigation team used multiple algorithms to direct the robot so each algorithm only had one specific task. Having each algorithm only perform one task allows the process of decision making to be quicker.

Safety

The Emergency stop systems are hardware based disconnects of both DC motors. The manual emergency stop is a 2 inch red push and lock button located in the back of the robot by the safety light. The wireless emergency stop is a radio frequency controlled relay that has a line of sight range of over 500 meters. When either emergency stop is engaged, power is physically disconnected from the 24 volt contactor coils. Once power is disconnected from the coil of each contactor, the contactors connected in series with the motor returns to its normally open state disconnecting power from the motors, stopping the motors. Figure 2 shows a schematic of the circuit for the manual and wireless emergency stops.

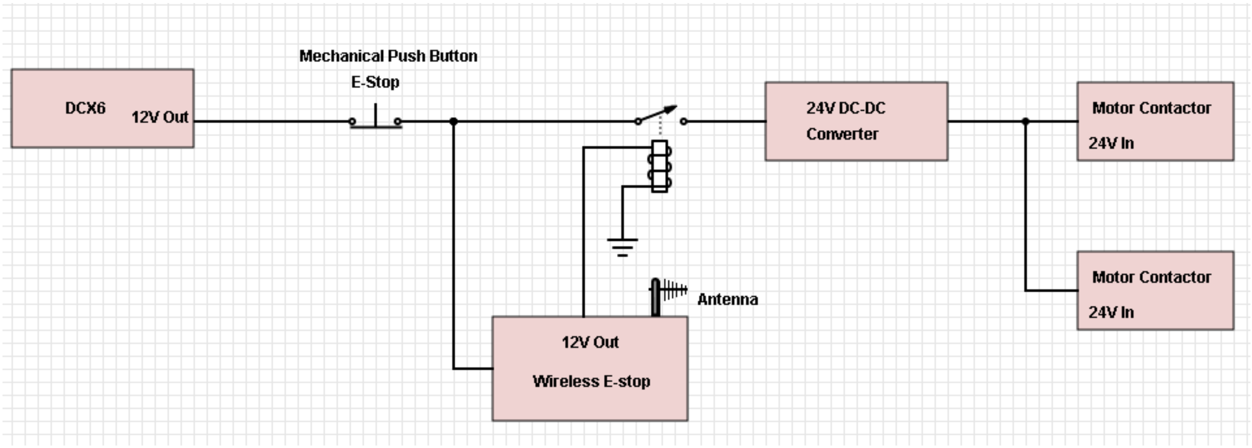


Figure 2 - Schematic of E-stop system

The safety light is a 6” by 2”, eight red LED light that is solid when the robot is on, and blinks every two seconds when the robot is in autonomous mode. The safety light is controlled through usb from the computer and driven by an arduino nano microcontroller. The microcontroller controls a 5 volt output that biases a N-Channel MOSFET to complete a 12 Volt circuit that turns on and off the safety light. Figure 3 shows the circuit for the safety light.

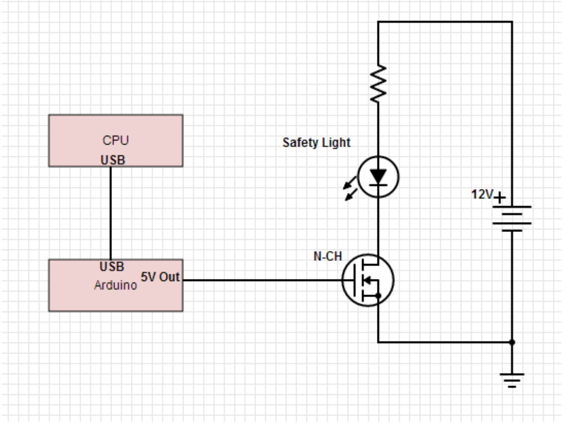


Figure 3 - Safety light schematic

Mechanical safety features of the robot include a bumper for the LIDAR, a metal box for the electrical components, and a waterproof enclosure. A simple metal bar was attached as a precaution to protect the LIDAR if our robot were to hit an obstacle. The electrical components were put in a metal box to prevent any effects of emf from the motors that are directly below the circuitry. Along with the metal box, the electrical components are also protected by a plastic enclosure. After the rain from the competition last year, it was determined that a waterproof enclosure was needed. Even the lid of the enclosure has a foam seal to prevent rain from coming in.

MECHANICAL TEAM

Frame

The frame was made out of 8020 Aluminum framing due to its modularity. This modularity would allow us to take parts of the frame apart easily if something on the robot needed changed or moved. A profile of 1515 was selected for the 8020 because it had the most options for attachable brackets and it provided us with the desired structural rigidity. The shape of the frame was all based on the theoretical components that needed attached to it.

Drivetrain and Suspension

The drivetrain configuration chosen for this year's robot was two drive wheels in the back with two castor wheels in the front. This option was primarily chosen because of the power having the drive wheels in the back could provide. Having the drive wheels in the back provides the most power for the two wheel configuration robots because it puts a majority of the weight from the robot on the back tires. Having the drive wheels in the back also allows for the robot to perform continuous turning. This can be done by either having one motor turning faster than the other or one wheel driving while the other wheel stays stationary. The importance of continuous turning is so that the robot can more easily achieve the competition goal of going over one mile per hour on average. A chain connects the drive shafts and the motors through a 1:1 gear ratio as shown in Figure 4. This ratio was selected because the max torque needed to perform any of the tasks, of 233 ft-lbs, could be achieved by the motors used.

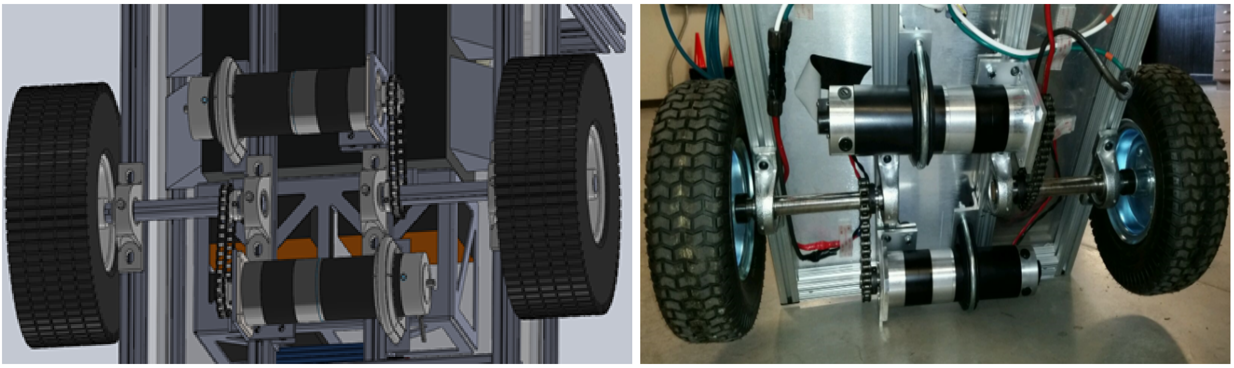


Figure 4 - The Solidworks model of the designed drivetrain

Air filled tires were used for the drive wheels since the pressure in the tires could be adjusted to dampen the effects of the terrain. With the sensors being on the front of the robot, the focus of the suspension was directed toward the front of the robot. The same concept with the air-filled tires was applied to the front portion of the robot as pneumatic casters were used. With the frame being as rigid as it was, the robot still did not handle well when traveling at more than 30% power. Therefore, it was determined that a suspension system of some kind was necessary. Instead of using a generic spring and damper system, we decided to use a system that is similar to what is used on a tractor to go over terrain that is very similar to that of the field used for the IGVC competition. The rocker bar system uses a bearing to rotate both pneumatic castors about so the actual effect of the bumps on the rigid frame is minimal. This system also guarantees that there will be at least three points of contact on the ground if there was really would terrain that we were to encounter. The rocker bar in Figure 5 shows the rocker bar as it's attached to the robot.

Figure 5 – The suspension system designed to stabilize the vision of the cameras

Mounts

The mounts are the supporting members or housing unit of specific components. The LIDAR mount (see Figure 6.1), is a simple fixed mount that is made out of 0.5 inch by 0.5 inch 6061 aluminum bar stock. The mount holds the LIDAR 13 inch high for obstacle detection, with a clearance of 6 inch from the ground, and protrudes the LIDAR out front to have a clear 270° view. The Tri-mount (see Figure 6.7) is a complex adjustable mount made out of ⅛ inch thick 6061 aluminum sheet metal that is placed on top of the tower. The Tri-mount holds the gps, two camera's, and a rain guard. The gps is placed at the top of the mount for no obstruction the gps may come into contact with. The camera's can be adjusted in two different planes, each plane has 18 different positions at 5° increments. Which gives each camera a total of 324 different position the camera can be placed. The IMU mount (see Figure 6.4) is a simple fixed mount made of high temperature resistant plastic. The mount is made to elevate the IMU away from the small electromagnetic fields from the computer and other circuits within the electrical box enclosure. The Payload and Battery support brackets (see Figure 6.2 and 6.3, respectfully) were made to hold different components but both were bent from ⅛ inch thick 5052 aluminum sheet metal. Each one of these support brackets holds one or more item(s),where the accessibility of the item(s) is necessary. But not to the point where these item(s) will be able to be thrown around during operations. Thus, an unique tolerance was made for each of these support brackets. These brackets was made with the use of bending sheet metal techniques, which allowed for porefessional aesthetics and a rigid structure.

Enclosures

There are two enclosures built for this robot: the polycarbonate panels and lid, and the electrical box. The polycarbonate panels and lid (see Figure 6.5) are the main enclosure protecting components on the inside from rain and dust. The polycarbonate panels and lid have a 'umbrella' style of shielding the components on the inside, with enclosure vents that allow airflow in the top of the enclosure (these vents have a special membrane that block water but allow air to pass through), four louver vents that block rain from entering but allow air to enter with the use of filters to block dust or mist from being draw in. With the use of rubber seals and caulked edges the polycarbonate panels and lid provide an adequate protection from the environment but at the same time provides airflow for cooling. The electrical box (see Figure 6.6) is a box bent from ⅛ inch thick 5052 aluminum sheet metal in a shape of 22 inch by 24 inch by 9 inch. All electrical components are placed inside this box with our innovative modular system. This modular system uses high temperature resistant plastic to back each circuit board and on the bottom of these backings is dual lock which allow for quick, interchangeable, adjustable, and stable components. Each components is on one plane to increase air flow and result in better cooling from the four 140mm fans, where two are blowing in and two are blowing out. the combination of these two enclosures provide protection and cooling for the components inside as well as gives an aesthetic appealing style.

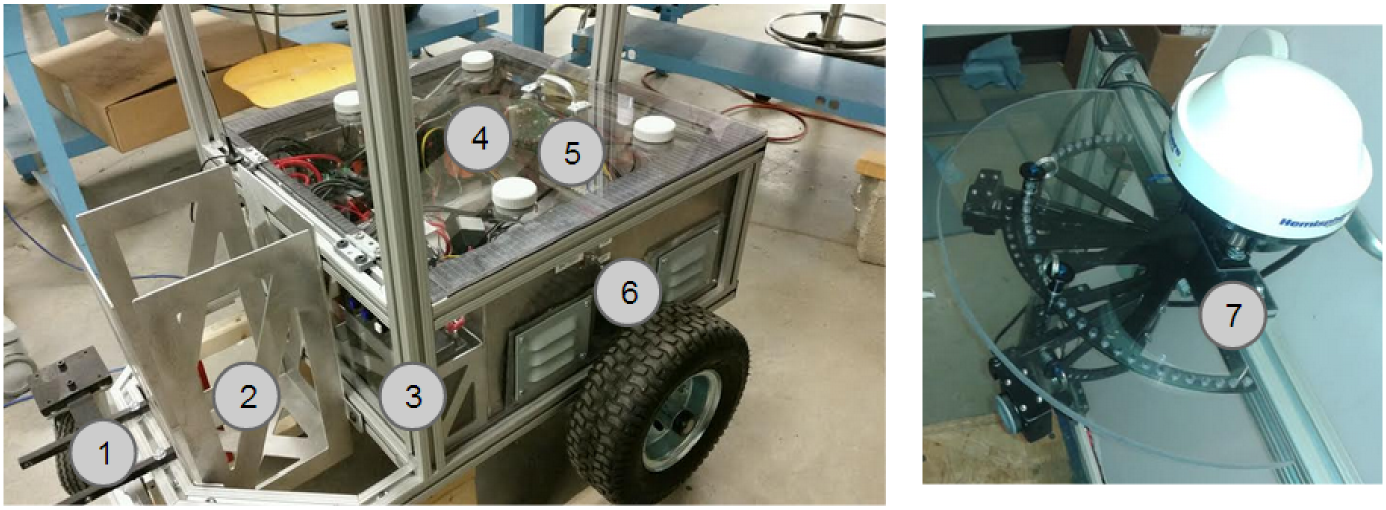


Figure 6 – Above is pictures of the Mounts and Enclosures made for the White Russian.

POWER TEAM

The electrical power subsystem is divided into two parts: the high power and the low power.The high power system objective is to control and supply power to motors that drive the robot through the course. The low power system objective is to supply power to an onboard computer and sensor components. The two power systems are powered by their own battery sources. The High power system is supplied by two 12 Volt, 18 amp hour batteries. The battery sources are placed in a series configuration to supply the motors with 24 volts when discharging and placed in a parallel connection when recharging. The Low power system is supplied by one 12 Volts, 20 amp hour battery. Both systems are completely isolated from one another to allow each system to be designed separately and prevent adverse transients on sensors from irregular current flow of high power system.

High Power System

The high power system goal is to control and supply power to motors that drive the robot through the course. This is accomplished with two Midwest Motions 24 Volt DC Motors. The motors are capable of spinning at 138 rpms and outputting 233 in-lbs of torque. With them the robot is able to easily traverse the course terrain. Supplying those motors are two 12 Volt 18 Amp hour batteries which allow normal operation for over an hour. Controlling the amount of power to the motors is the RoboteQ HDC2450 Motor Controller. All communication is done through the 25-pinout on the motor controller. Below in Figure 7 is the wiring for the 25-pin connector. Using a 9-pin serial connector, the computer is able to send commands to the motor controller. The motor encoders are also wired directly to the motor controller for speed feedback.

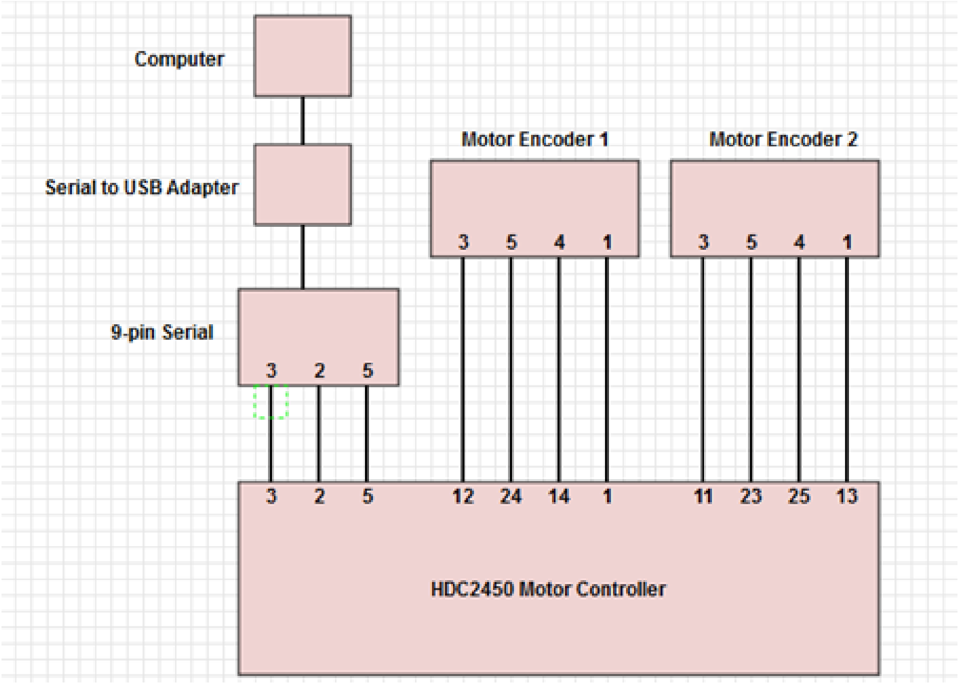


Figure 7 - Motor Controller Serial Connections

Due to the high currents seen in the high power system extra protection is needed for the components. The high power circuit is shown below in Figure 8 between the switches and the motor controller. This small circuit contains 35 Amp fuses, resistors to prevent switch arcing, and diodes for back emf protection.

The high power system also has a charging system that is separate from the low power system for safety purposes. Charging is done on-board and only requires a power connector and the batteries in parallel and connected to the charger via the two high power DPDT switches.

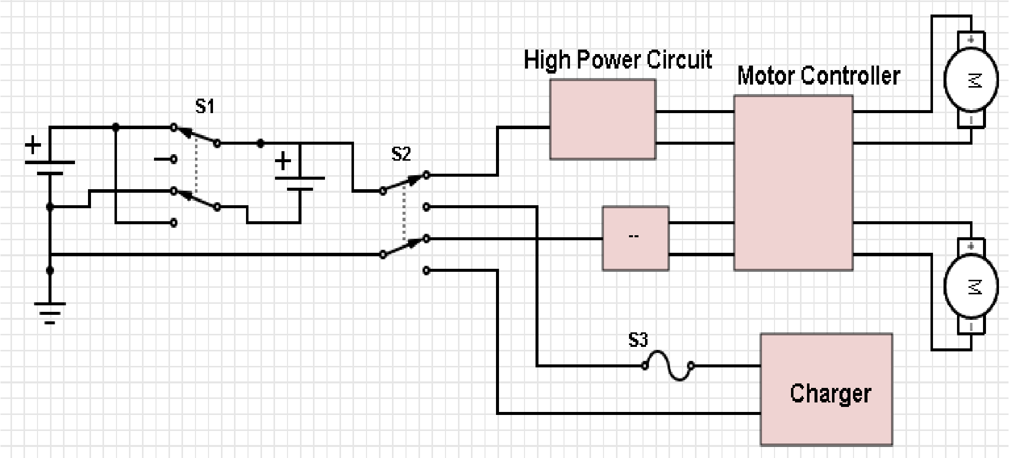


Figure 8 - High Power System High Level Diagram

Low Power System

The low power system goal is to safely supply power to an onboard computer and external sensors. Sensors supplied by the low power system include LIDAR, GPS, and Cameras. The system has two power sourcing options, local and external. The local power supply is one 12 Volt, 20 amp hour battery and external supply, any 110 Volt AC power from a wall outlet. By using an external source to power the low power system, the local power source is not discharged at any time other than when the robot is being driven. The external power supply allows for battery runtime to not be a factor when coding or debugging software. When the LPS system is driven off of an external power supply, the local power supply is also recharged automatically. All switching between power sources and battery charging is done automatically internal of the robot when the external source is connected and disconnected. The automatic switching is done by P-Channel and N-Channel MOSFET's controlling the current paths between source and load. A PCB for the switching circuit was laid out in Eagle PCB software as seen in Figure 9.

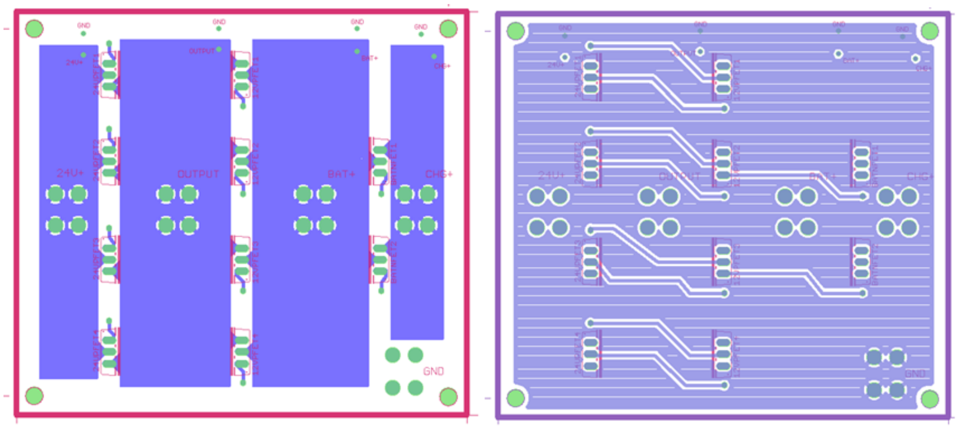


Figure 9 - PCB of Auto Switching Circuit

The low power system powers an onboard computer as well as external systems with voltage regulation by Opus Solutions DCX6. The onboard computer uses standard ATX compliant powering protocols with the DCX6. Using ATX powering protocols allows the computer to be powered on and off with the flip of a switch. The toggle switch to turn on and off the computer connects and disconnects an input signal to the DCX6. The DCX6 then acts as a slave device to the computer, having the computer system control its own power requirement prevents inadequate startup and shutdown times that can cause memory loss and data corruption. Since the computer acts as a master to the DCX6, the external sensors are only powered when the computer is on and vice versa when it is off.

The low power system requirements include powering the system by the local power supply for an extended period time while the robot is on the course. In years past the minimum requirement for IGVC teams required a runtime of 45 minutes. A runtime of 45 minutes is enough time for immediate debugging and multiple trials on the course before having to swap or recharge the local power supply. To determine a worst case scenario runtime of the low power system a load jammer was installed on the computer and a maximum load of 42 watts was driven from the 12 Volts output from the DCX6. A worst case scenario current requirement of the system is 15.4 amps, 184.8 Watts. From the local power supplies battery discharge curve it can supply this amount of power for approximately 53 minutes. The low power system current requirement during a typical load scenario is 7.3 amps, 87.6 Watts, and can be supplied for over 90 minutes. Figure 10 shows the schematic for the low power systems circuit.

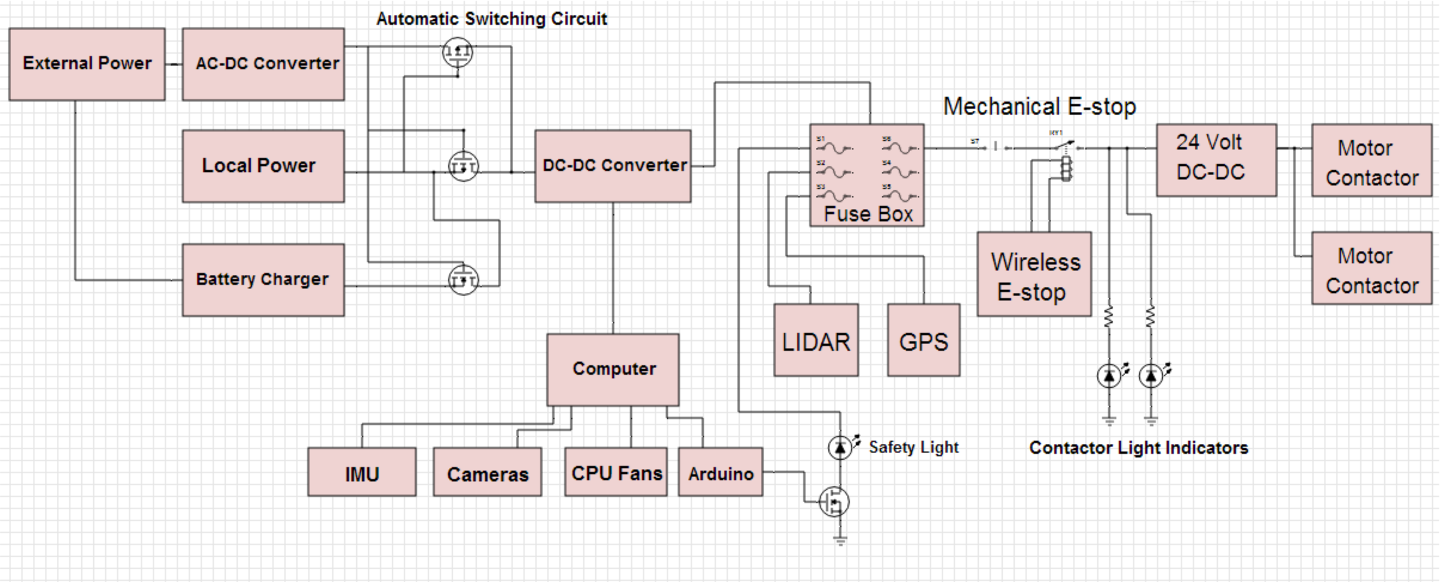


Figure 10 - Low Power System High Level Diagram

SENSOR TEAM

Localization

The localization part of this years robot is designed to give a general idea of where the robot is in compared to where it started and where it needs to go. The robot must have an idea of where it is along the course if it is to make the best decision about its next move and how to best execute it. In order to navigate to the GPS waypoints, a GPS receiver is used to collect the GPS data. The GPS receiver operates by calculating the time difference from when the satellite sent the signal to when the receiver actually receives the signal. The receiver needs a clear view of four or more of the satellites in order to return a value. An IMU (Inertial Measurement Unit) is also used to give the robot an idea of direction, using several on-chip sensors to compute various data values, the most important of which being the magnetometer. Both units data is collated and sent to the Navigation Team for processing and decision making.

The GPS unit in question is a Hemisphere A352 SMART Antenna, which has an accuracy of ~1m, with the website putting it at .6m. The unit has an onboard processor, and understands a library of commands that can be sent over a serial port or CAN connection. In this case, the commands are sent over a serial port and read back when the unit returns data. The algorithm starts by sending the command to the GPS unit to start data collection. Once the GPS returns the relevant data, the string received back is parsed to separate out the substrings associated with the latitude and the longitude. These strings are then converted to doubles and made available for the Navigation team to access. As with all the sensors, the GPS program was implemented as a stand-alone library with methods to initialize and begin the data collection loop. It then spawns a separate thread to allow the Navigation team to grab data from the collection loop without interrupting the processing or acquisition of new data points. There is no filtering done on the acquired data, besides the initial filtering that the GPS does automatically. The GPS unit filters out the most extreme effects of multipathing. Multipathing is best described as a signal echo that is often associated with tall structures or dense upright surroundings. With what the GPS filters out and the locale of the competition, it was deemed unnecessary for further handling of this issue. The GPS is also noisy, but from the tests that were done involving a stationary and moving unit, the largest error seen was 1.2m off. The actual observed average was closer to the range of around .4 to .7 m off. With the requirements of the competition and the time restraints on developing the programs, it was decided to forgo a filter on this data.

An IMU is also used to provide a basic compass bearing corresponding to the robots direction of motion. The IMU selected is a Phidget 1044 Spatial 3/3/3. It is a 3-axis board with several data collection units onboard, including a magnetometer and gyroscope. The only data used from the system is the magnetometer data. The IMU returns the relative strengths of the magnetic field in each axis, and then simple trigonometry is used to find the compass bearing. Once the bearing is calculated from the data it is made available to the Navigation team, so that it can be accessed from any of the threads that it is needed. Similarly to the GPS unit, the IMU is implemented as a stand-alone library that the Navigation team can initialize in a main thread and then access the data from the threads that the program needs.

Object Avoidance

The object avoidance sensors are used to search for and map obstacles surrounding the robot, and allows the Navigation team to avoid said obstacles. For the competition, the main obstacles are large construction barrels and an assortment of unspecified large objects that are inserted randomly throughout the course. In order to detect these obstacles, the robot uses a LIDAR system. The LIDAR uses a high-intensity infrared laser and shoots it out in a 270 degree arc. It then measures the time for the laser reflection to return to the lens and calculates the distance from this information. The LIDAR returns the distances, and the angles that these distances are at is implied based on the order they come in and the resolution of the LIDAR.

This year, a SICK LMS100 was used. It is accurate and reliable, and is a solid and reliable tool. The system was set up as a Java client, communicating back to the server controller. The controller, which is implemented as a library, is called from the main program and is used to set up the LIDAR for data collection. Once the LIDAR unit is ready, the program sends a command to start the data collection. The LIDAR client collects a sweep, then waits for a response so the next sweep can begin. In this fashion, the data is collected, processed and stored, and then new data is collected. This helps to avoid collisions where the program is attempting to store the data while new data is queued up behind waiting to be written in, essentially locking up the array. On the other side, the Navigation team also acts as a client, requesting access to the data as soon as it is safe. In order to ensure that data is always available, a two array safety feature is added in. This ensures that whenever the navigation team requests data, it will have access to the most recent complete set. These data arrays are only mutable from within the library itself, meaning that the Navigation team does not have the permission or ability to alter or delete this data, preventing corruption or data loss.

Vision System

The Vision System is responsible for locating distinct objects surrounding the robot and determining its distance relative to the robot. These objects include white lane lines painted in the grass, and red and blue flags. To detect the objects in the environment, a camera system is implemented to capture images of the robot's surroundings. An algorithm is then used to pinpoint objects depending on pixel colors.

The key of this year's Vision System was to implement a hardware solutions as opposed to a software solution to allow for a simplified algorithm while still maintaining the same level of reliability. In order to maximize the effectiveness of the Vision System, a wide field of view is necessary to be able to detect as much objects as possible. Thus, two cameras will be positioned on the robot to provide maximum coverage and suitable vision surrounding the robot. In order to maximize the horizontal range of the camera system the two camera view will minimize overlapping. This configuration will maximize the range that each robot can detect obstacles by increasing the camera view of area to left and right of the robot. The Firefly MV is affordable line of low resolution cameras produced by Point Grey. The Vision System will be implementing two Firefly MV 0.3 MP Color USB 2.0 camera (P/N: FMVU-03MTC-CS) shown in Figure 1. This is a small camera (24.4 x 44 x 34 mm) that provides low resolution (640 x 480) images at 60fps. This camera also comes equipped with a global shutter that captures the image all at once instead of a rolling shutter that captures an image over time. The camera also is able to internally improve image quality. The camera also comes equipped with a standardized CS lens mount that will allow for implementing the Tamron 13VM2812ASII CCTV lens. The lens offers customizable parameters to maximize its effectiveness for various situations. This includes varying focal length, focus, and aperture. All three of these parameters can be controlled independent control rings that can be locked during operation. Also, the lens comes standard with a multi-coating on the lens to reduce ghosting and flares caused from over exposure to light and will allow for the truest possible image colors, even under dynamic lighting conditions. Both the camera and lens choice divert the strain of handling the varying light conditions from the software to the hardware.

The algorithm to accomplish object detection will take advantage of the computer vision libraries of OpenCV. OpenCV is an open source computer vision and machine learning software library that focuses on computationally efficiency for real-time applications. This allows the algorithms to take advantage of computationally efficient filters. The algorithm will also implement the FlyCapture SDK provided by Point Grey. The FlyCapture SDK uses a simplified object-oriented interface to connect directly to the Firefly MV cameras. To interface with both OpenCV and the FlyCapture SDK, C++ was employed. Figure 11 depict the object detection algorithm that can be customize its threshold values for any object.

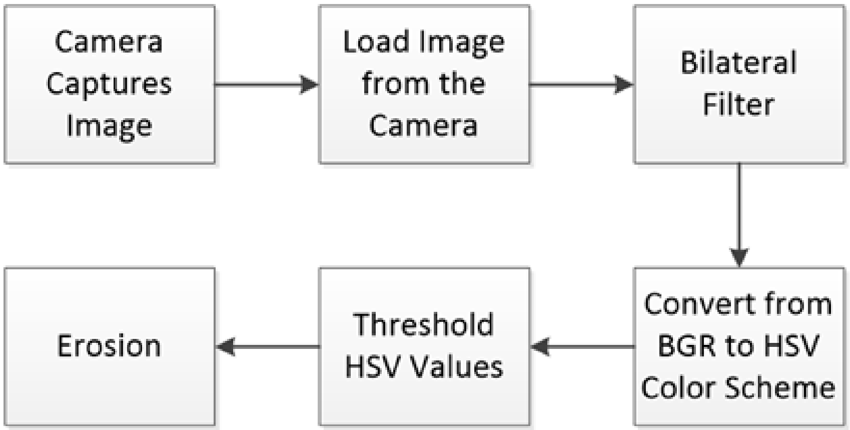


Figure 11 - Object Detection Algorithm

The main innovation of this algorithm is implementation of a bilateral filter as a means to reduce image noise in the captured image while retaining the distinct colors in the picture. A standard Gaussian blur takes a pixel and computes the average pixel quantity of pixels around it. The amount of surrounding pixels used to compute the average can be modified to increase or the decrease the amount of blurring. Then the average pixel quantity is applied to the original center pixel. The issue with this method is that it can cause an overall blurring effect on distinct lines. The bilateral filter takes the filter to the next level by adding a second component to the computation of the average pixel value. It adds a range filter that takes the photometric similarity between pixels. If a pixel's value ranges significantly from that of the center pixel, it is ignored. This allows the bilateral filter to maintain edge preservation while smoothing. This allows for a distinctive HSV range to be targeted for effective thresholding. Figure 12 shows the effects of the object detection algorithm designed to search for white lines.

Figure 12 - White Line Detection - Original Image (Left) and Thresholded Image (Right)

Once the pixel locations of the detected objects are determined through the camera image, they are then translated to their corresponding to a real world distance in respect to the robot. These real world locations are the information that needs to be passed to the navigation team. This is achieved by passing the pixel locations of detected images through a homographic matrix, calibrated specifically to each camera, that will convert the pixel locations into a list of real world vectors including the x and y distances in respect to the robot. These real world locations are sent to the navigation team.

NAVIGATION TEAM

Navigation System Architecture

The navigation system of the White Russian consists of three algorithms running in parallel that continuously run using the latest data from the sensors. There is also a weighted averaging system that sends commands to the motors based on the weighted average of the decisions of the different algorithms. The weights used to calculate motor commands change based on where the robot is on the course and if sensor data is considered reliable.

Our team chose to use multiple algorithms running in parallel rather than a single algorithm because it allows each individual algorithm to be more tailored to a specific purpose. As a result of multiple single purpose algorithms, each is simpler and therefore easier to debug and understand. This structure also allows for greater threading capabilities which makes each iteration of the navigation system happen faster which allows the robot to react to changing conditions quickly. The architecture of the navigation system with the correlation between the program used for each task is shown in Figure 13.

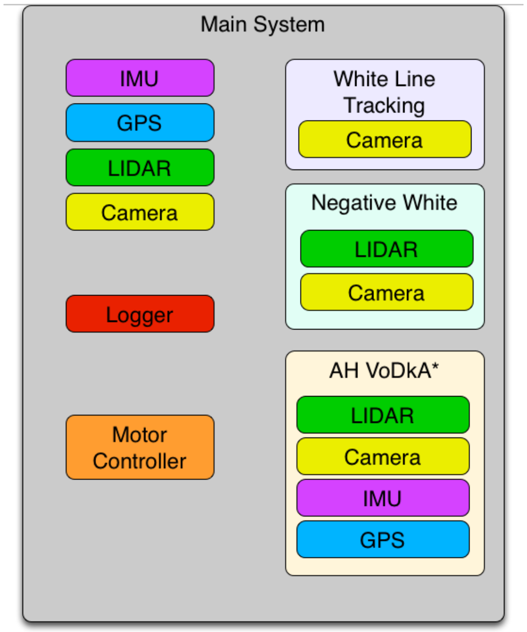


Figure 13 - Navigation System Architecture (Different boxes indicate different threads, same color indicates same object)

Algorithms

The first of the three algorithms that our robot uses for navigation is white line tracking which works by treating each white pixel from the cameras as an attractive force that will result in the robot wanting to stay centered between the white lines. An important part of the white line track algorithm is that it makes white points seen further away have a greater effect of how much the robot turns then white points close to the robot which keeps the robot from turning straight toward a white line right next to the robot and driving over it. The white line tracking algorithm also treats white points on the on either side of the robot as slightly closer to center than they actually are to keep the robot from driving directly on top of a white line if only a single line is seen.

The second algorithm negative white works by giving obstacles a repulsive force. This algorithm works with our white line tracking algorithm to allow our robot to track white lines while avoiding obstacles. This is done by using LIDAR detections within a certain distance as a force much like how white line tracking algorithm works, but pushing rather than pulling.

The third algorithm in our navigation system is AH VoDkA* which is an acronym for a homebrew version of Dijkstra's and A* works by checking the distance from every LIDAR detection to every other LIDAR detection to analyze where there are gaps big enough for the robot to fit through. Once the robot knows where there are gaps large enough to fit through it then analyzes which gap will take the robot closest to the current goal. This is operation is demonstrated below in Figure 14.

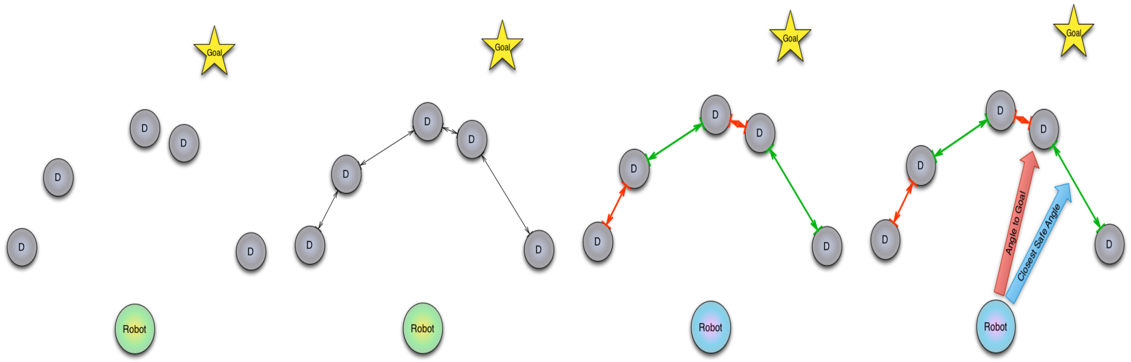
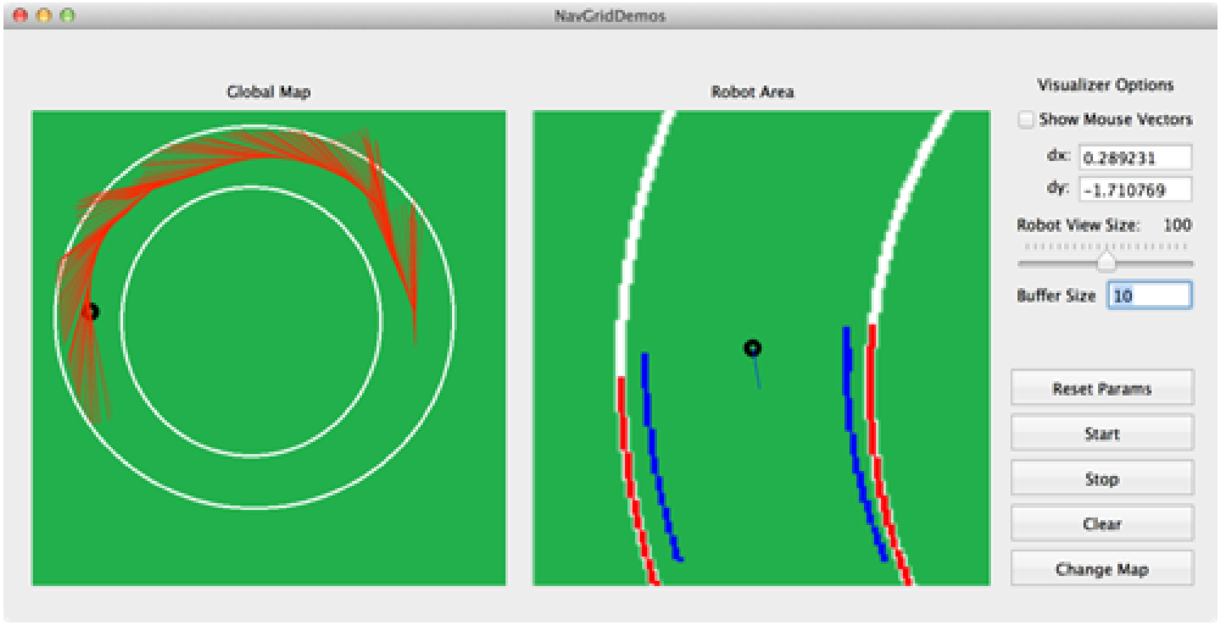


Figure 14 - Demonstration of AH VoDkA* algorithm

Simulation and Testing

The navigation system on the White Russian was tested before being applied to the robot using a simulator written specifically for this robot architecture.



dx: 0.289231
dy: -1.710769
Robot View Size: 100
Buffer Size 10

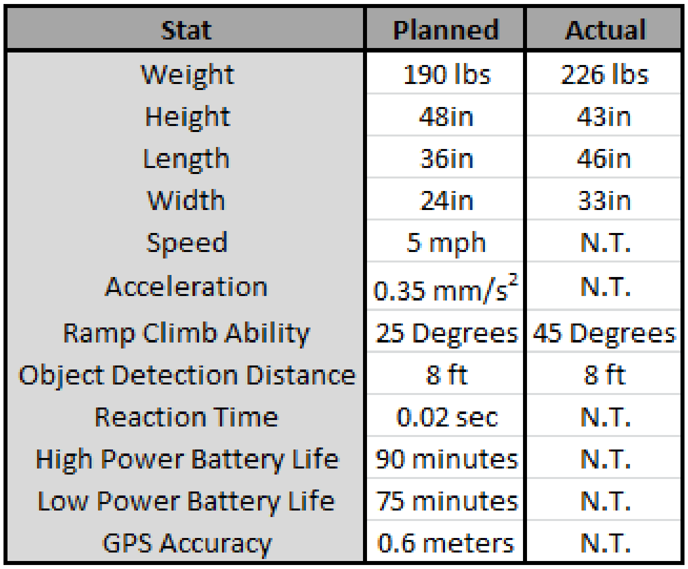
Figure 15 - Simulator Example

Using simulation to test the navigation algorithms was a decision made out of the desire to test the algorithms separately from the robot so that potential problems with algorithms could be identified separately from potential issues in the robot design. The structure of the simulator created closely resembles that of the architecture of the robot. This similarity in structure allowed for a easy transition to the robot. Figure 15 above shows an example of our simulator running our white line tracking algorithm where the image on the left is a global view of a test course and the right image shows what the robot can see. The red lines in the picture on the right show the portion of the white lines that can be seen by the robots camera where the blue lines represent a buffer system that keeps the robot at a set distance away from the white lines.

ROBOT CAPABILITIES

Table 2 compares the theoretical performance characteristics of White Russian to its actual capabilities. Some characteristics were not recorded yet due to unexpected problems that delayed testing. These included not testing things from in the chart, and how the vehicle deals with complex obstacles including switchbacks and center islands dead ends, traps, and potholes.

Table 2- The performance capabilities of White Russian (N.T. means Not Tested)

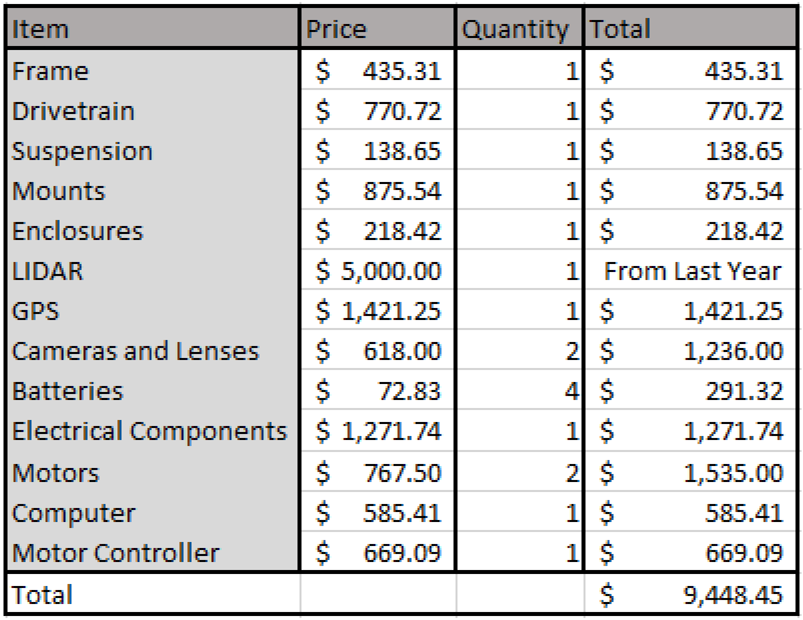


Stat	Planned	Actual
Weight	190 lbs	226 lbs
Height	48in	43in
Length	36in	46in
Width	24in	33in
Speed	5 mph	N.T.
Acceleration	0.35 mm/s^2	N.T.
Ramp Climb Ability	25 Degrees	45 Degrees
Object Detection Distance	8 ft	8 ft
Reaction Time	0.02 sec	N.T.
High Power Battery Life	90 minutes	N.T.
Low Power Battery Life	75 minutes	N.T.
GPS Accuracy	0.6 meters	N.T.

TEAM BUDGET

The summary for the cost of the components on the White Russian was documented in Table 3. Total cost for White Russian would have been $14,368.36 if all of the parts had been bought this year; however, our team is using the backup LIDAR from last years team. Since that was the largest expense, the new total cost of our robot is $9,368.36 for all of the parts used.

Table 3- Cost report for the money spent on the White Russian



Item	Price	Quantity	Total
Frame	$ 435.31	1	$ 435.31
Drivetrain	$ 770.72	1	$ 770.72
Suspension	$ 138.65	1	$ 138.65
Mounts	$ 875.54	1	$ 875.54
Enclosures	$ 218.42	1	$ 218.42
LIDAR	$ 5,000.00	1	From Last Year
GPS	$ 1,421.25	1	$ 1,421.25
Cameras and Lenses	$ 618.00	2	$ 1,236.00
Batteries	$ 72.83	4	$ 291.32
Electrical Components	$ 1,271.74	1	$ 1,271.74
Motors	$ 767.50	2	$ 1,535.00
Computer	$ 585.41	1	$ 585.41
Motor Controller	$ 669.09	1	$ 669.09
Total			$ 9,448.45

CONCLUSION

Through regimented milestones, White Russian was built and moving with plenty of time to perform testing to prepare for the competition. Although all of our testing has not been completed to date, we have confidence that White Russian will do well at this years IGV Competition. On behalf of all the people involved on this project from York College, we would like to thank the judges for their time and consideration.