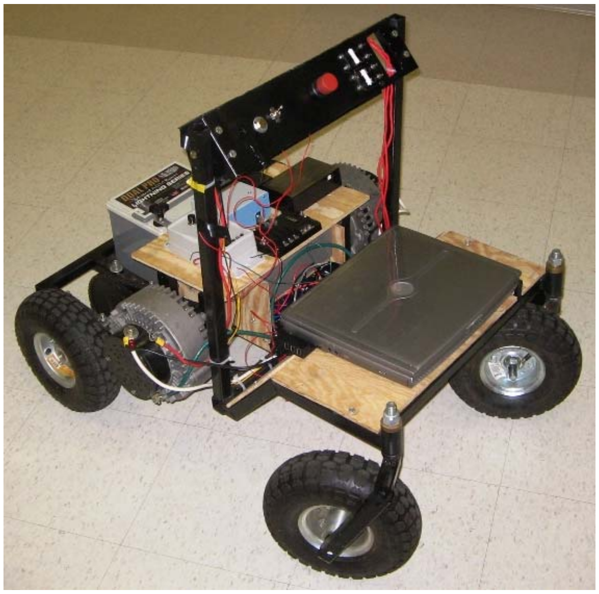
Design Report for

"Should Be Trivial"



"Should Be Trivial"

UMBC's Entry in the
13th Annual Intelligent Ground Vehicle Competition
Submitted by team: **Future SuperSeniors**

I certify that the engineering design in this vehicle by the UMBC student team has been significant and equivalent to what might be awarded credit in a senior design course.

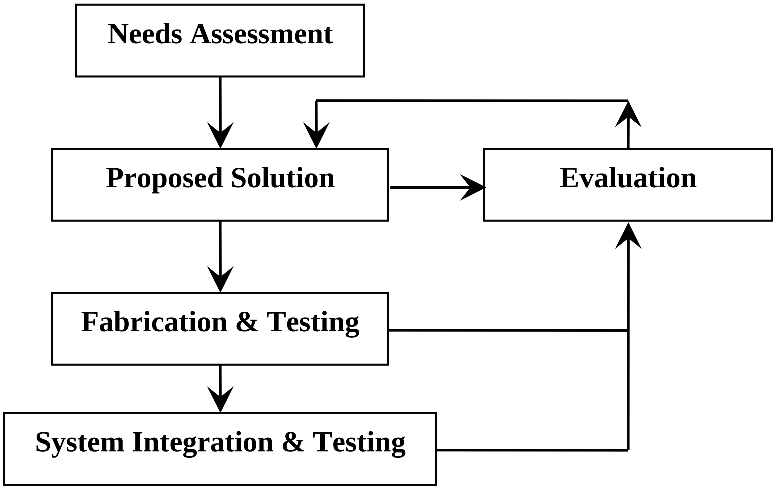
Joel Morris

UMBC Team Faculty Advisor
Computer Science and Electrical Engineering Department

Table of Contents

Introduction

This report details UMBC’s entry in the 13th Annual Intelligent Ground Vehicle Competition. The design team consists entirely of undergraduate students studying Computer Engineering, Computer Science, and Mechanical Engineering. The following diagram summarizes the design process of the project.



Design Process of “Should Be Trivial” Project

The Ground vehicle team was assembled and competition challenges were assessed in Fall, 2004. Solutions to these challenges were proposed, explored and evaluated for feasibility through December 2004. Fabrication and Testing of the robot components started in January 2005 and continues until the competition in June 2005. System Integration and Testing began in May 2005. The hours spent on the project are summarized in the following table:

Team Member	Hours		Team Member	Hours
Andrew Wilson	619		Patti Ordonez	40
Erik Broman	400		Holly Bennett	30
Albert Hsu	112		Lee Linkoff	30
Kevin Fisher	85		Solomon Mikael	30
Jeff Pierson	83		Lloyd Emokpae	30
Theo Meyer	80		Steve Sapp	20
Nate Price	70		Devin Burns	15
Emmanuel Aduroja	48			
			Total:	**1692**

The components of the ground vehicle are organized into three main categories according to function:

1. **Locomotion**: all systems that are responsible for the mechanical movement of the ground vehicle during operation
2. **Sensors**: all systems that collect data about the environment the ground vehicle operates in
3. **Data Processing / Navigation**: all systems that are responsible for the intelligent processing of sensor data to navigate the ground vehicle through its environment

Locomotion

The locomotion system addresses the following competition specifications:

- Maximum dimensions of 9'L x 5'W x 6'H, with a minimum length of 3'.
- E-stop with robot mounted and wireless triggers. Wireless reach of 50' minimum. Capable of stopping ground vehicle in no more than 6' on inclines up to 15°.
- Ability to climb 15° inclines, and drive through sand pits up to 3".
- Traverse course with lane's turning radius no less than 5'.

Steering

Two steering mechanisms were compared to provide the ground vehicle with adequate maneuverability: Ackerman (car style) Steering and Differential (tank style) steering. Differential style steering was chosen because it matches or surpasses Ackerman style steering in these respects:

	Simplicity to Implement	***Cost***	***Turning Radius***	***Meets or Exceeds Specs***
Ackerman				X
Differential	X	X	X	X

Drive System

Motors

Both DC electric motors and combustion engines were considered to provide mobility for the ground vehicle. Combustion engines were quickly disregarded as a non-viable option as they lack native forward/reverse functionality. The following formula was used get a ballpark figure for power needed to drive the ground vehicle up a 15° incline.

$$P_{motor} = (\mu mg\cos\theta + mg\sin\theta)v$$
$$P_{motor} = mgv(\mu\cos\theta + \sin\theta)$$

P = power (hp)
m = mass (kg) =~150lbs*.4536 = 68.04kg
g = gravity = 9.8m/s^2

v = velocity (m/sec) = 2.232meters/sec
θ = degree of incline = 15°
μ = coefficient of friction = 0.80

Solving this equation for P gives peak requirement of 2.1hp. This shows that motors from a previous UMBC project (Briggs and Stratton E-Tek motors: 15hp peak, 6hp continuous) could be re-used on the ground vehicle to save cost Only one motor was available for use on the ground vehicle, so one motor was purchased. A gear-down ratio of 4:1 was used to lower the maximum wheel speed which in turn gives greater speed control to the PWM motor controllers.
10" wheels were selected to give enough clearance between the ground and the vehicle to drive through sand pits of 2-3" depth.

Battery

In order to keep vehicle sensors' electronics isolated from voltage transients generated by the motors, a separate battery was selected for the drive system. In an effort to save space, only one battery was used (12V 33Ah deep cycle Sealed Lead Acid).

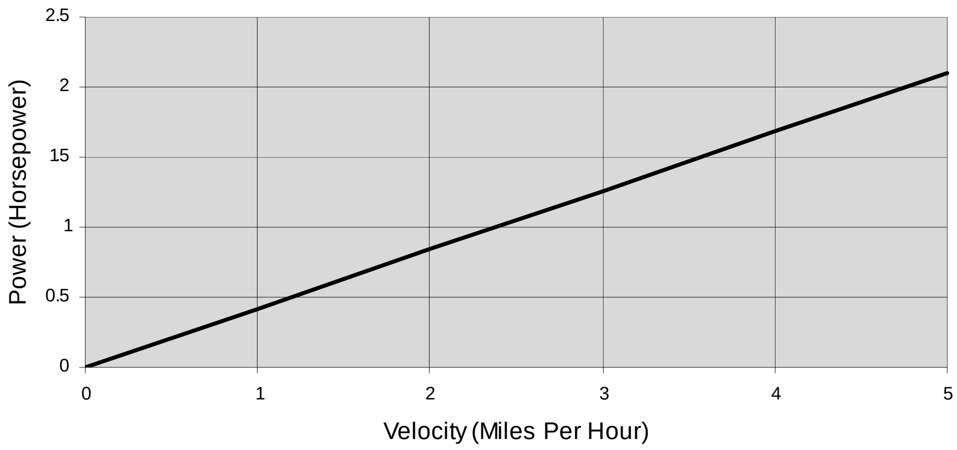
Motor Controller

Initially, two 12V 40A PWM motor controllers (Innovation First Victor 884's) were installed to power the E-Tek motors. Unfortunately, these failed quickly. The RoboteQ AX2500 was selected as a capable replacement. The AX2500 is capable of driving two motors at 120A each via a 5V Pulse-Width-Modulation (PWM) digital signal. Unlike the Victors, the AX2500 has no fans to power (or accidentally destroy). Heat is dissipated through its extruded aluminum body which acts as a heat sink. Additionally, the AX2500 features advanced options such as current monitoring via a serial (RS-232) interface and current limiting. Current is limited to 40A per motor in order to keep 8AWG and 10AWG chassis wiring from overheating. Two Hall Effect sensors are mounted near the drive sprocket to detect actual motor speed. Actual drive speed will be monitored and controlled in software.

Expected Performance

In saving on motor cost and battery space, efficiency is sacrificed because the E-tek motors run most efficiently at 48V. In order generate 1hp per motor (2hp total) at 12V, nearly 80A per motor is required. Because of the current limiting, this is not possible. To verify that the motors will still work with these limitations the equation used above is plotted with respect to velocity, and inspected.

Power Required as Function of Velocity up 15° Incline



Clearly the E-Tek motors are still quite capable of driving the ground vehicle up a 15degree incline, albeit at a lower speed.

In reality, the motors will not be operating at these extreme conditions all the time (up hill & with a large friction coefficient). It is expected that motors will not draw more than 12amps average. This 24amp total continuous draw gives projected run-time of just over an hour.

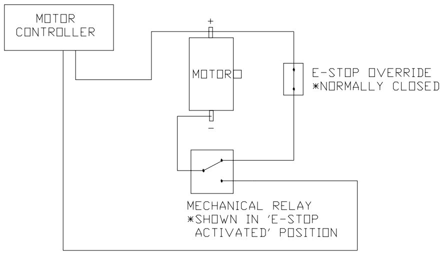
Chassis

The ground vehicle chassis was designed to accommodate the E-Tek motors, battery and all other system components. The chassis measures: 37"L x 27"W x 29"H. It is constructed from 1" steel box tubing which was available as free scrap material. By mounting the motors on the outside of the robot chassis, rather than the inside of the chassis (conventional mounting), more chassis space is available for electronics. Low cost and easily-machinable plywood panels were used to mount components and protect them from the elements.

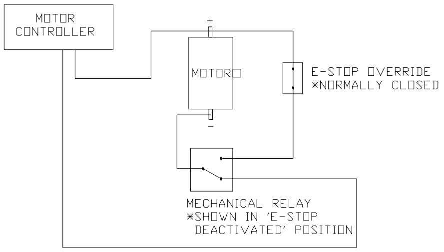
Emergency Stop Mechanism

Braking is achieved via “dynamic braking” by mechanically switching the connection to the motor coils from the motor controller to a load resistor. This slows the motor to a stop within 6feet on up to 15 ˚inclines.

The following diagrams show the two states of the e-stop circuit: e-stop engaged, e-stop disengaged. The e-stop is engaged by turning power off to the relays. This means that when there is a loss of power to the robot, the emergency stop circuit will keep the robot from moving. An override switch installed on the switch panel allows the robot operator to disengage the e-stop when pushing the robot to a new location.



Dynamic Braking E-stop Active



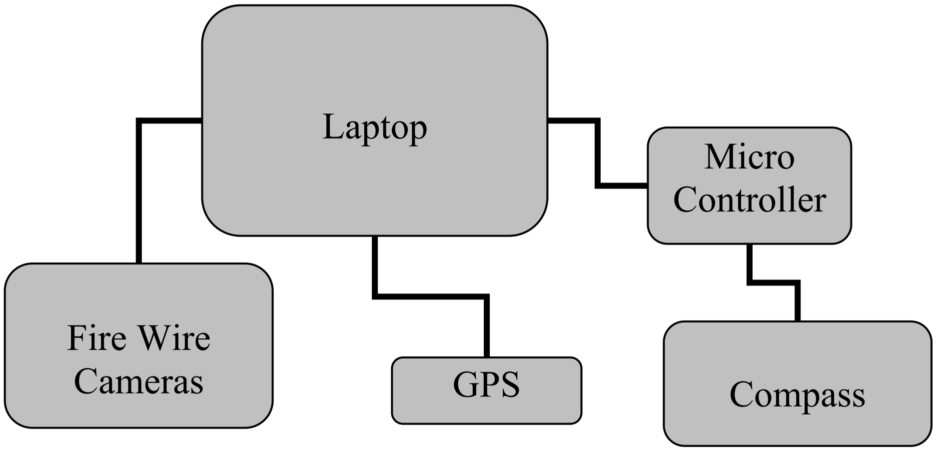
Dynamic Braking E-Stop Deactive

As per competition rules, the E-Stop can be engaged via a wireless remote control, and a red pushbutton mounted the back side of the robot. The e-stop may also be activated by a TTL signal generated by the computer. As an additional safety measure, the radio receiver (433mhz) mounted on the robot is in constant communication with the transmitter. If the receiver ever loses communication with the transmitter for more than a second, it triggers the e-stop automatically. The e-stop must be manually reset by a pushbutton mounted on the RF receiver enclosure.

Vehicle Sensors

The following sensors were selected to allow the ground vehicle to detect the following obstacles, as per the competition specifications:

- 5-gallon White Pails
- Orange and White Construction Drums
- Cones, Pedestals and other Barricades used on roadways and highways
- Yellow and/or White, Solid and/or Dashed 3” painted boundary markers
- GPS waypoints (within 2 meters)



Environmental Sensors Interface

Vision

The ground vehicle's primary environment sensor is a stereoscopic pair of *Unibrain Fire-I Board* Firewire cameras used for both object detection and range finding. Stereoscopic video cameras were chosen for their inexpensive cost relative to the popular SICK laser rangefinder modules. The cameras are mounted in the middle of the ground vehicle, 25" above the ground. The cameras have a horizontal field of view (F.O.V.) of 80°, and a vertical F.O.V. of 65°. This gives the robot a horizontal F.OV. of approximately 4.2 feet across, immediately in front of the robot (smaller than two full size road barrels) and a F.O.V. of approximately 25 feet across at 15 feet. Frames are captured in color from both cameras by the laptop and processed to find location and distance information for obstacles.

Coordinates

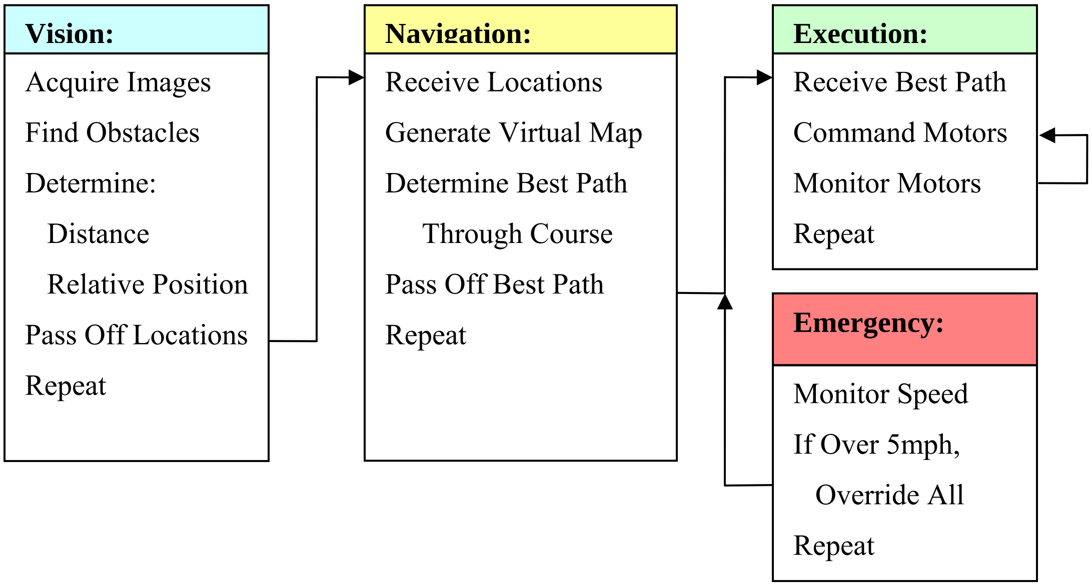
The GPS unit (*Laipac TF30*) is used to determine global position of the ground vehicle. With WAAS (Wide Area Augmentation System) signal correction, accuracy of the unit is within 3 meters 95% of the time. Coordinates are updated once per second. As per the competition guidelines, the ground vehicle successfully "reaches" a waypoint when it travels within 2 meters (4 meters diameter) of the target coordinate. While this GPS receiver cannot guarantee 100% waypoint realization, it is expected to work at least 70% of the time. This GPS system will be replaced at a later time to increase accuracy to within 2 meters needed by the waypoint challenge. The GPS module is connected to the ground vehicle laptop via a USB interface.

The *Devantech IIC Digital Compass* is used as an additional navigation reference. It is calibrated before the competition with respect to a constant heading. The compass data is passed to the computer through the *Brainstem GP* microcontroller board. The *Brainstem GP* is also connected to the laptop via a USB interface.

Data Processing & Navigation

The software program is split into four significant processes: vision, navigation, execution and emergency conditions. These four processes are designed to run concurrently.

Four Main Software Processes:



Four Main Software Processes

Images are captured from the two video cameras via a Firewire interface. Images are then processed by code written in C/C++ utilizing the OpenCV open-source image processing libraries. At the submission time of this report, the image processing code is still in the fabrication & testing stage, thus the proposed design follows:

Vision

Barrel, Pothole Detection

Images will be processed to find contours using the OpenCV FindContours() function. These contours are found between two immediately adjacent areas of high contrast. This is effective to detect edges of obstacles, however it does not discern between objects and gradients of shading. To remove garbage contours (areas of shade, blades of grass, paterns) the area inside the contours will be calculated and compared against a range of "valid obstacle" areas. These "valid obstacle" areas will be determined experimentally by deciding an effective range for the vision system, then capturing images of obstacles at varying distances in this range and calculating their contour area. Contours without "valid obstacle" areas will be discarded and not inspected further. The contours will then be passed through an open source function that detects square shapes from contours.

This will allow the detection of road construction drums, and barrels. The contours will also be passed through the OpenCV findEllipse() function to detect elliptical shapes. This will allow the detection of pot-holes. Finally, color of the areas bounded by the remaining contours will be compared to a range of “appropriate color” values which correspond to the colors found on road construction barrels, and buckets. These “appropriate color” values will be found experimentally, accounting for varying lighting conditions. Position of the obstacles within the image frame will be recorded and referenced against compass heading. This data will be combined with the distance found below and passed to the Navigation algorithm.

Line Detection

To detect lines, an OpenCV Hough map transform is performed on the contour map. This algorithm will search and flag all contours with a distinct heading for a distance. The lines are compared to lines found in previous frames. Points on the line will be sampled, distance to sample found, and sent to the Navigation algorithm for consideration.

Range finding

Distances to obstacles will be calculated by using a stereo disparity map. The two captured images are passed into the OpenCV FindStereoCorrespondence() function. The result is a 256 shade grayscale image with pixel values from 0 (black) to 255 (white). This gives 256 units of distance. The real length of these units is variable, and depends entirely on the effective range of the stereo correspondence. The effective range is dependant on both the distance between the cameras, and the angle in which the cameras point towards each other. The optimal spacing and angle configuration is found experimentally based on ability to detect and react to obstacles. To sample distance to obstacles, the contours for each obstacle are overlaid on the disparity map image and points inside the contour are selected at random and averaged. If the cameras are configured for optimal stereo performance at 5-15ft, this gives a resolution of approximately 2”.

Navigation

The navigation code is based on an implementation of “Potential Field” path planning. A potential field path is similar to the concepts of potential fields, namely the notion of a field which surrounds an object. An attribute that most fields have in common is that they get weaker with distance. Our path planning algorithm will simulate this weakening of fields by having resistive values set for obstacles. The algorithm set paths so as to move through the least resistance around obstacles and avoid lines. The path of least resistance is defined here as the area around an obstacle where the field has the least amount of strength or value. To accomplish this path planning the navigation thread will create a two dimensional potential field map and asses a best path through the map.

Potential Field Mapping

A two dimensional map will be created based on the obstacle distances from the vehicle. This distance data is obtained from the vision thread. A potential field consisting of impedance states will overlay the map with each pixel having a default impedance state. Every obstacle and line detected from the vehicle’s current position will be given a high impedance state value. To create a field-like potential, each pixel on the map will be given a degree lower impedance state the obstacle or line adjacent to it, and recursively each adjacent pixel to the previous pixels will have another lower degree of impedance. This pixel impedance mapping will continue until the next adjacent pixel is already at a higher impedance state, the current pixel’s impedance is at its lowest state, or the boundary of the current map has been reached.

Path Assessment

When pixel mapping is completed a path will emerge which will consist of pixel impedance contrasts. Every pair of pixels that have contrastingly higher or lower impedance levels will become a safe path for the vehicle. The paths will approximately be in the middle of obstacles and lines which will give the vehicle enough mobility and clearance. This path will be sent to the execution thread.

Execution

The execution code enables the vehicle to move and will directly control the motors' speed and direction. Each motor will be controlled separately. The implementation for movement will be for the vehicle to move to a location, stop and then turn and then move again to a new location. The path given by the navigation code will be curved, since it will outline the potential fields of obstacles. To make the path easier to follow, the curved lines will be approximated by straight lines. Each motor will be commanded through the *Brainstem GP* micro controller through 5 Volt PWM signals. This signal combines both direction and speed. Signals are sent within the integer range from 0-255. Forward motion are controlled through values from 148 – 255 where 255is full forward. Similarly reverse is controlled through values 0 – 106 where 0 is full reverse. The dead band rate for the motor controller is 16% so the ranges of values for stopping are from 107 - 147. As well as setting the motion of the vehicle, the execution thread will monitor the motors' speed. The monitoring of the motors while in motion assures that the vehicle is proceeding correctly to its destination. Other monitoring information will be retrieved from the emergency conditions thread. If the vehicle goes beyond a certain speed as determined by the emergency thread, the motors will be slowed down to adhere to that speed. This execution process will continue until the E-stop is activated.

Emergency Conditions

The emergency conditions thread will be a constant monitor for emergency stopping conditions. Specifically, this process monitors the vehicle's speed. The emergency process will receive constant feedback from the Hall Effect sensors relaying the RPM values from each motor. From the RPM values received, the code will calculate the equivalent MPH for each motor. The formula for converting RPM to MPH is as follows:

$$\frac{RPM * SecondsPerMinute * \pi * wheelDiameter(inches)}{InchesPerMile} \equiv MPH$$

Once the motors exceed the 5 mph limit, the emergency thread will constantly send an alert to the execution thread that signifies the vehicle exceeded the speed limit until the vehicle is below 5 mph. Once the execution code receives the alert signal, the vehicle will slow down to below 5 mph. Other emergency conditions are yet to be implemented.

Bill of Materials

A concentrated effort was made to keep costs to a minimum. The final cost analysis of components used on the ground vehicle follows:

Part	**Unit Cost**	**Quantity**	**Total Cost**	**Cost to Team**
Locomotion:				
Briggs and Stratton E-Tek Motor	$400.00	2	$800.00	$400.00
RoboteQ AX2550 Motor Controller	$495.00	1	$495.00	$495.00
E-Stop Circuit & High Power Relays	$300.00	1	$300.00	$300.00
Dual Pro LS 2200 Deep Cycle Battery	$133.00	1	$133.00	$133.00
Cherry GS100902 Geartooth Hall Effect Sensor	$40.00	2	$80.00	$0.00
60 Tooth Sprocket, #35	$29.68	2	$59.36	$59.36
1" Axle Pillow Block Bearing	$12.95	4	$51.80	$51.80
1"x1" Low-Carbon Steel Box Tubing (per foot)	$2.00	25	$50.00	$0.00
4 x 10" Knobby Hand Truck Tires	$36.00	1	$36.00	$36.00
Drive Wheel Hub	$14.99	2	$29.98	$29.98
15 Tooth Sprocket #35	$7.43	2	$14.86	$14.86
#35 Riveted Roller Chain, 3/8" Pitch	$2.21	6	$13.26	$13.26
1/4" Plywood (24" x 24" sheet)	$3.49	2	$6.98	$6.98
Sensing:				
Unibrain Fire-I Board Color Cameras w/ 2.1mm Lens	$139.00	2	$278.00	$278.00
GPS Module	$200.00	1	$200.00	$200.00
Brainstem GP Microcontroller	$81.00	2	$162.00	$162.00
Serial to USB Adapter	$40.00	1	$40.00	$40.00
Devantech IIC Compass Module	$51.00	1	$51.00	$51.00
Misc Wire and Wiring Hardware	$50.00	1	$50.00	$50.00
Component Boards and Enclosures	$50.00	1	$50.00	$50.00
Control:				
Dell Inspiron 1150 Laptop	$989.00	1	$989.00	$989.00
		Totals:	**$3,890**	**$3,360**

Sponsorship

UMBC Student Government Association

UMBC Computer Science & Electrical Engineering Department

Baltimore Section of the Institute of Electrical and Electronics Engineers

UMBC College of Engineering and Information Technology

Special Thanks

UMBC SAE Mini Baja Team

Corey Fleischer

John Keyser

Chad Richman