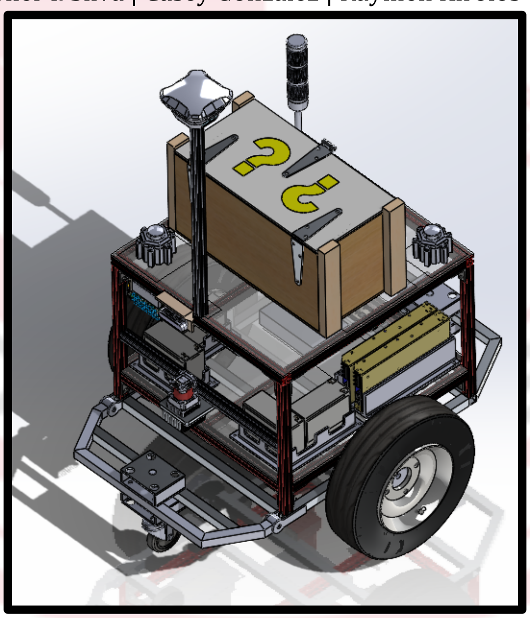
California State University Northridge 2024 - Spark-E

California State University, Northridge
Department of Engineering & Computer Science
Advisor: Dr. Vidya Nandikolla | vidya.nandikolla@csun.edu

Team Captains: John Harold | John.harold.935@my.csun.edu
Christopher I. Silva | christopher.silva.960@my.csun.edu

5/15/2024

John Harold | Michael Hartigan | Juan Aguilar | Nick Camaratta | Alec Weinstein | Van Exel Vispo | Rachel Alano | Carlos Aguilar | Tristen Harry | Pierceson Yu | Brandon Garcia de Alba | Christopher I. Silva | Casey Gonzalez | Raymen Kirolos | Ashley Frisch



STATEMENT OF INTEGRITY:

I certify that the design and engineering of Spark-E by the currently listed student team has been significant and equivalent to what would be awarded credit in a senior design course at California State University Northridge.

_________Vidya Nandikolla____

Table of Contents

1 INTRODUCTION

California State University, Northridge's Intelligent Ground Vehicle (IGV) team proudly presents Spark-E. Breakthroughs from mechanical and software perspectives in the 2024 academic year led to today's innovative design. Spark-E's main objective is autonomously navigating a static obstacle course while maintaining and delivering a 20 pound payload. It does so using various sensors and integrates them with the robot's mechanical components.

2 ORGANIZATION

Hardware Team	Software Team
Christopher I. Silva*	John Harold*
Nick Camaratta	Carlos Aguilar
Ashley Frisch	Juan Aguilar
Casey Gonzalez	Rachel Alano
Michael Hartigan	Brandon Garcia de Alba
Van Vispo	Tristen Harry
Alec Weinstein	Raymen Kirolos
Pierceson Yu	

3 DESIGN PROCESS

3.1 Mechanical

The intelligent ground vehicle is a fully autonomous vehicle that must be able to navigate a static course with obstacles and follow lines on the ground in a prescribed amount of time. The team has made many innovations from previous years, including a spring-loaded roller wheel to help overcome ramps and compensate for rough terrain. Also, aluminum plates were swapped with acrylic plates to reduce internal heating issues. The later sections explain the important design choices in detail.

3.11 Frame Design

The IGV 2023-24 team worked on the chassis shown in figure 1. Spark-E moves with center-wheel drive and has two guide wheels. This frame design was ideal, given the size restrictions and ability to make tight turns. The main benefit of the center wheels providing power is that the driver wheels takes most of the robot's weight. The typical four-wheel chassis struggles with fitting tight turns without a steering mechanism.

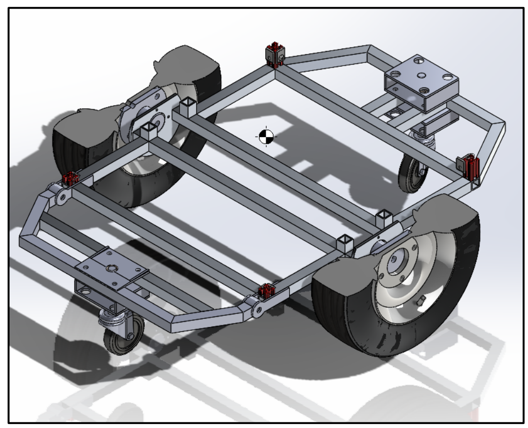


Figure 1: Spark-E's chassis

3.12 Motors

The main drive motors chosen were AmpFlow E30-400 DC motors with 32:1 planetary reduction gearboxes, model number E30-400-24-PR32. The motors were selected due to their ease of control, ability to provide sufficient torque for our application, and low cost. The motors chosen are easy to control due to their internal structure and use of brushes with an applied voltage to induce rotation, unlike brushless DC motors, which require electronic commutation to induce rotation, which is more complicated and expensive to implement. The planetary gearbox mated with the motor decreases the rotational speed to a more usable value of approximately 160 RPM. The gear reduction increases torque, aiding the IGV over ramps in the course.

3.13 Weatherproofing

Weatherproofing the IGV was accomplished using weather stripping as a prudent measure to safeguard its contents from the elements. Acrylic, while durable and transparent, may not provide the same level of protection against moisture and dust as other materials. A tight seal was created by applying weather stripping along the mating edges of the acrylic and IGV frame, preventing water, dirt, and debris from infiltrating the interior. This is especially crucial for outdoor applications or environments prone to humidity fluctuations, where moisture ingress could compromise the integrity of sensitive items housed within the box.

The process of weatherproofing the IGV with weather stripping was straightforward yet effective. First, we cleaned the surfaces where the weather stripping was to be applied to ensure maximum adhesion. Next, we carefully affixed the weather stripping along the edges of the acrylic, ensuring a snug fit. The compressible nature of the stripping allows it to conform to irregularities in the surface, creating a tight seal when the acrylic is mounted to the frame. This barrier acts as a defense against moisture and airborne particles, maintaining the clarity and functionality of the IGV.

4 ELECTRICAL

4.1 Overview

The electrical system of our robot consists of a primary power source, several power distribution boards (PDB), various sensors, custom circuit boards, a microcontroller, a central computer, and a motor controller. All these systems work together to help the robot navigate its environment. Figure 2 describes the robot's electrical system in its entirety.

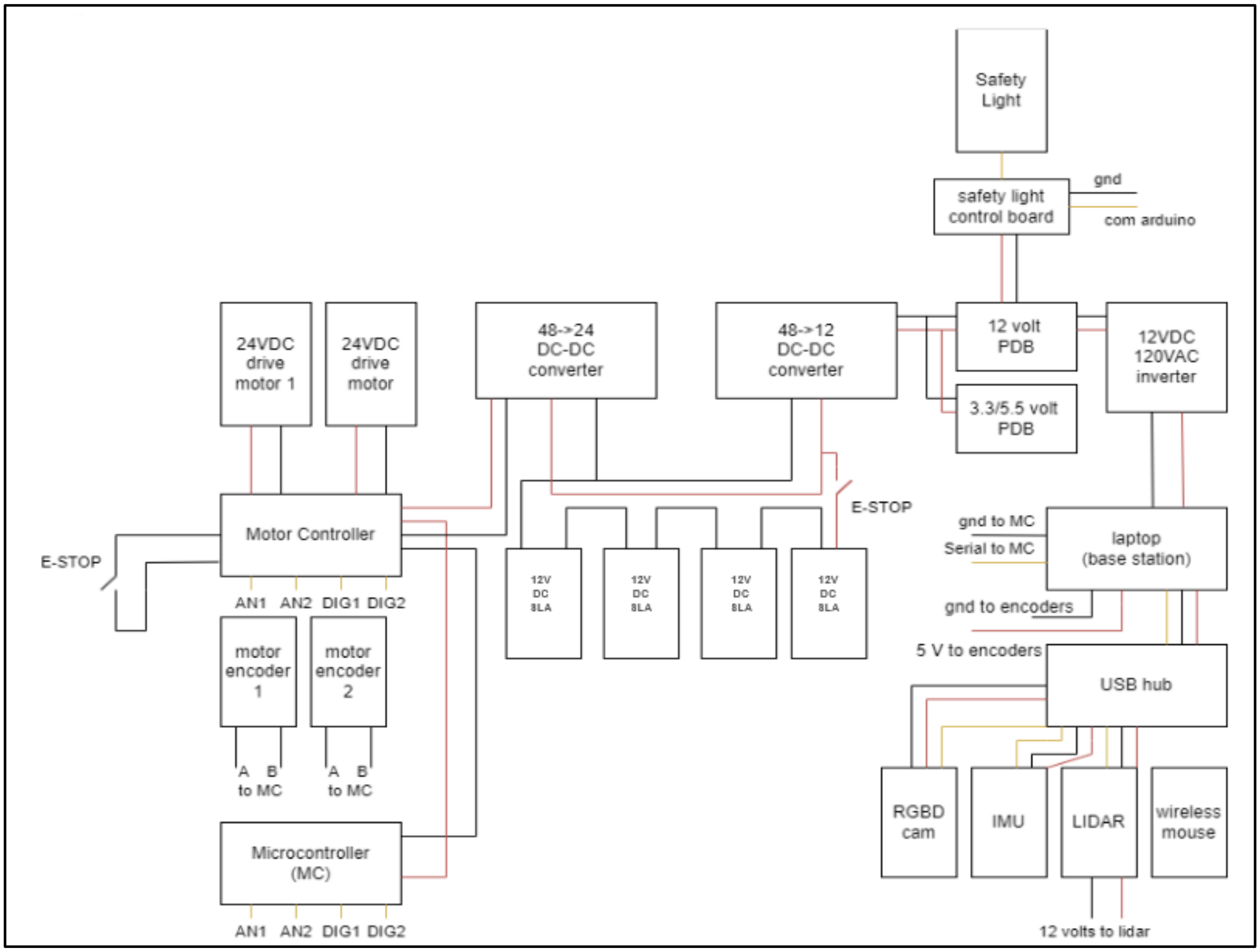


Figure 2: Overall Electrical Diagram

4.2 Power and Power Distribution

The main power for the robot consists of four 10 amp hour 12V SLA batteries connected in series. The main power connects to the two DC-DC converters in a parallel configuration; one steps down the voltage from 48V to 12V and the other from 48V to 24V.

The converter that steps down the voltage to 12V connects to the 12V PDB, which powers auxiliary devices like the LiDAR sensor, the TIP-120 robot state light strip, 12V to 120VAC inverter, and secondary 3.3V/5V PDB. The 12V to 120VAC inverter powers the laptop running ROS (base station). All sensors are powered by the USB hub connected to the base station except for the LiDAR, which is powered by the 12V PDB.

The 48V to 24V DC-DC converter powers the electronic speed controller. The speed controller provides variable power to the motors. A diode is in line with the positive lead from the DC-DC converters to the motor controller to prevent back EMF from shutting down the converters. The base station powers the microcontroller (Arduino Mega) and motor encoders. Two emergency stop (E-stop) push buttons are wired into the electrical system. One E-stop is connected in line with the positive lead of the primary power source. The second E-stop is connected to the motor controller's on/off switch terminal block and can be controlled wirelessly over 100 feet.

4.3 Motor Control and Sensors

4.31 Motor control

The motor controller chosen for the robot was a Cytron SmartDriveDuo60. The Duo60 can provide a continuous current of 60 A to each motor and 100 A for 1 second. The motor driver also has overcurrent protection, which means if either one of the motors is subject to a stalled state where the maximum current is drawn, the motor controller will shut itself off.

4.32 Light Detection and Ranging

An essential component of Spark-E is the Light Detection and Ranging (LiDAR) sensor for which the Hokuyo UTM-30LX was used. The LiDAR measures the distances to objects by emitting laser pulses and measuring the time it takes for the light to bounce back from those surfaces. The LiDAR returns units that describe the surrounding area. LiDAR data is known for providing highly accurate spatial data. Despite its incredible utility, Hokuyo UTM-30LX's field of vision is limited to the 2D plane level to where it is mounted.

The LiDAR's primary utility is obstacle detection and mapping. The Hokuyo UTM-30LX scans up to 30 meters within its 270-degree field of vision and scans every 25ms, making it ideal for outdoor navigation, obstacle detection, and mapping.

4.33 Inertial Measurement Unit

The inertial measurement unit (IMU) used is the Vectornav 200T. The IMU provides data on Spark-E's orientation, velocity, and acceleration. As the robot's internal compass, the IMU helps interpret sensor outputs by providing a true global coordinate system (GCS). The GCS simplifies navigation and displacement calculations as all numbers can be referred back to Spark-E's starting orientation.

4.34 RGBD Camera

The camera mounted to Spark-E's front is an Intel RealSense Depth Camera D435 with a depth sensor and a wide field of view up to 10 meters. It serves a similar function to the LiDAR in that it aids with object detection. Additionally, the camera allows Spark-E to detect and follow lanes as well as potholes.

While the LiDAR is excellent for accurately measuring simple objects with uniform thickness, the RGBD camera can see things that the LiDAR might miss. For example, potholes and lines will go undetected by the LiDAR; however, the camera, which is angled slightly towards the ground, would pick up on this. The LiDAR might see just the tip of a cone, while in reality, a cone is much larger than its tip. Spark-E can count on the camera to correct the false judgments made by the LiDAR.

5 SOFTWARE

5.1 Overview

The software we use to operate the robot is the Robot Operating System (ROS). Within ROS, we utilize the ROS Navigation Stack and implement additional code for line detection. Overall, software architecture comprises obstacle avoidance, localization, and decision-making subsystems highlighted in figure 3. Real-time sensor data is collected for obstacle avoidance and localization. Within the ROS Navigation stack, this data is used to populate local and global costmaps, indicating areas of high and low difficulty for the robot to navigate. Information from the global costmap is then used in the global planner of the ROS Navigation Stack to generate a high-level path from the robot's initial position to a goal. At the same time, the local planner refines this path based on immediate surroundings and dynamic obstacles, resulting in a more localized and calculated path. The culmination of this process is the generation of a command velocity, which is sent to the microcontroller. The microcontroller then translates this velocity command into motor control signals (PWM), effectively controlling the speed and direction of the robot's motors and ultimately enabling it to traverse the planned path.

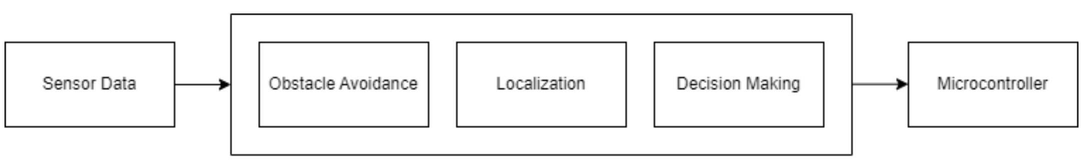


Figure 3: General Software Architecture

5.2 Obstacle Avoidance and Mapping

The Intel RGBD camera and Hokuyo LiDAR collectively handle obstacle avoidance. Data from these two sensors are fed to the navigation stack to store information about obstacles in the world and generate 2D cost maps on a global and local level. The global cost map creates long-term plans for the entire environment, while the local cost map is primarily used to help the IGV avoid obstacles in its immediate surroundings.

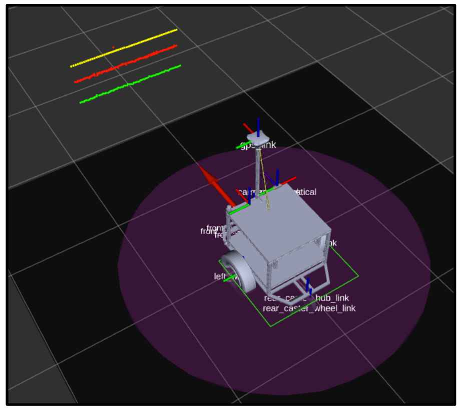


Figure 4: Object Detection using RVIZ

Our mapping process relies primarily on our LiDAR and is implemented through Hector SLAM. Hector SLAM is a laser scan data-driven Simultaneous Localization and Mapping (SLAM) algorithm, meaning it only requires input from a LiDAR sensor to generate a 2D map (figure 4), highlighting its simplicity. It also excels with LiDARs that exhibit a high update rate, a quality our LiDAR possesses, making it an optimal choice. The mapping aspect is critical for our global cost map as it functions as our robot's memory of its overall environment. Figure 5, highlights Hector SLAM's functionality in creating a global cost map in conjunction with the LiDAR.

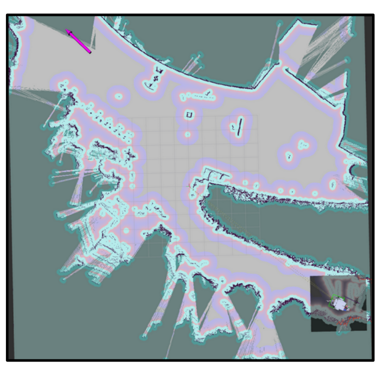


Figure 5: *Generated Global Cost Map*

5.3 Localization

Localization is the process of determining the position and orientation of the robot relative to its environment. Localization of the robot is handled by the motor encoders, IMU & GPS. All these sensors are fused using the extended Kalman filter (EKF) to determine where the robot is relative to the world around it. The EKF employs a dynamic model that describes how the robot moves when subject to a control input and external forces. In other words, the EKF takes sensor data to calculate the robot's position continuously.

5.4 Decision-making

The decisions made by the robot involve a six-step process critical to the robot's ability to navigate autonomously (as seen in figure 6):

1) The camera and LiDAR acquire sensor data for obstacle avoidance and from the IMU, GPS, and motor encoders for localization and an optimal path.
2) Sensor data fusion occurs to localize the robot concerning its environment and improve perception.
3) The global cost map is generated based on sensor data, which will help the robot determine how traversable its environment is.
4) Once the global cost map is generated, the global planner generates a high-level path based on the robot's starting point, intermediate waypoints, and goal.
5) The local costmap is generated with continuously updated incoming sensor data.
6) Finally, the local planner generates feasible trajectories and ultimately chooses an optimal path for the robot to navigate through, issuing velocity commands to the microcontroller.

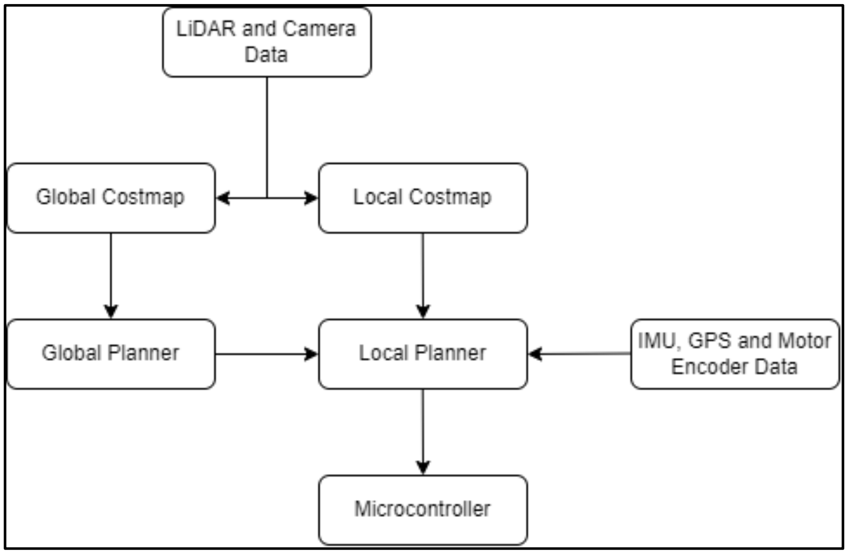


Figure 6: Decision Making Architecture

Continuously changing data from onboard sensors is fed into the global and local costs to generate a map of cost intensity. The global planner utilizes this information to develop a high-level path from start to finish. In contrast, the local planner refines this path based on immediate surroundings and dynamic obstacles, resulting in a more localized path.

6 SYSTEM INTEGRATION

Streaming data from our physical components to our software was achieved seamlessly through packages readily available on GitHub. Moreover, these packages could be easily modified if specific parameters needed to be adjusted. All the decision-making was completed in ROS, which outputs command velocities.

The Arduino motor controller code continuously waits for command velocities from ROS. The command velocities were converted to PWM signals that the Arduino would send to the motor controller. This was achieved by calculating the rpm of the motor at full duty cycle, which came out to be 160 rpm. Multiplying the rpm by the tire's radius would yield the vehicle's linear velocity. A maximum and minimum PWM value was calculated to ensure Spark-E stays between two and five miles per hour. Finally, all command velocities received from ROS were converted to their respective PWM values within the Arduino code.

7 NAVIGATION

7.1 Lane Following

The lane following relies on the Intel Realsense D435 Depth Camera. The process begins with the camera capturing an RGB image, which undergoes image processing to filter the white colors from the rest of the image. The binary image, which highlights the white areas, is used with contour detection to differentiate between all white surfaces and the white lines. With these white lines, the depth image is used in conjunction to get the point cloud representation of these contours. This facilitates smooth integration of our lines to the move base framework. Allowing the white lines to be treated as obstacles ensures the robot stays within its designated path.

7.2 GPS Mapping

The Emlid Reach RS+ is our choice of GPS for the robot, which offers high-precision location tracking to navigate complex environments and autonomously create maps. We input specific coordinates into the system as waypoints, which guide the robot through a dynamic map while avoiding obstacles. Custom code was written to extract raw data from the GPS module, which is crucial for achieving accuracy in positioning. This code also integrates with a graphical user interface (GUI) that was developed. The GUI, shown in figure 7, monitors the robot's location in real-time, inputs new waypoints and adjusts paths as needed. The combination of accurate GPS data, strategic waypoint placement, and real-time data visualization through the GUI enables the robot to navigate and map its environment effectively and autonomously, adapting to new challenges.

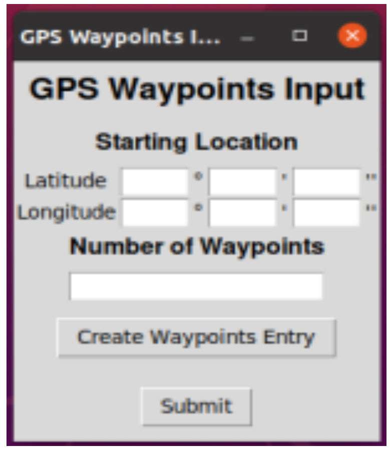


Figure 7: GUI Initialization Window

8 SIMULATION

8.1 Overview

Simulation and 3D visualization tools have provided a means to test and validate algorithms, behaviors, and designs in a virtual environment. Gazebo was the 3D simulation environment primarily used during this project. RVIZ was the 3D visualization program used to test the robot in Gazebo and in real life. Testing in a simulated environment saved the team hours of potentially downtime when the robot could not be tested.

8.2 RVIZ

Robot Visualization (RVIZ) is a 3D visualization tool in ROS. It allows the user to visualize the robot and the sensor data that it reads in real time within a virtual 3D world. During test runs, the team uses RVIZ to observe the inputted data from the camera and LiDAR and interpret the data to determine how accurately the robot depicts the surrounding real-world or simulated environment. It has been an essential tool for testing and troubleshooting the robot's navigation system.

8.3 Gazebo

Gazebo is an SIL virtual environment simulation testing process. This was used to create a 3D dynamic replica of the course that the vehicle would eventually navigate. The simulation was especially beneficial in testing the vehicle's lane-following capabilities. This program was necessary because the vehicle was sometimes available to run because of the mechanical maintenance that was constantly applied.

9 OVERALL SYSTEM PERFORMANCE

The 2024 IGV, Spark-E is a fully functioning and working vehicle. Spark-E can take inputs from its various sensors and finds its pathway in outdoor and indoor environments. The software team is calibrating some parameters to ensure Spark-E can complete the course in under five minutes.

10 TOTAL COST ESTIMATE

Mechanical Components			
Item	Quantity	Unit Price	Total Price
Wheel Hubs	2	$ 54.99	$ 109.98
Steel-hub Pneumatic Tires	2	$ 32.56	$ 65.12
Pneumatic Casters with Rubber Whe	2	$ 47.90	$ 95.80
Miscellaneous	-	$ 1,602.45	$ 1,602.45
Total Price Estimate			$ 1,873.35

Electrical Components			
Item	Quantity	Unit Price	Total Price
Motor Controller	1	$ 216.90	$ 216.90
Motor	2	$ 275.95	$ 551.90
Encoder	2	$ 70.03	$ 140.06
Arduino	1	$ 33.25	$ 33.25
Camera	2	$ 602.50	$ 1,205.00
Laptop	1	$ 2,409.00	$ 2,409.00
IMU	1	$ 3,614.98	$ 3,614.98
LiDAR	1	$ 4,217.48	$ 4,217.48
GPS	2	$ 1,204.99	$ 2,409.98
E-Stop (Button)	2	$ 72.24	$ 144.48
E-Stop (Remote)	1	$ 60.19	$ 60.19
DC-DC Converter	2	$ 431.39	$ 862.78
Total Price Estimate			$ 15,866.00

Grand Total	$ 17,739.35