# WIRED

## Wisconsin Robotic Exploration Device

**IEEE Robot Team**
**University of Wisconsin - Madison**

**Required Faculty Advisor Statement**
I certify that the engineering design of the new vehicle, WIRED, described in this report, has been significant and equivalent to what might be awarded credit in a senior design course.

Professor Nicola Ferrier
Department of Mechanical Engineering

# 1. Introduction

The IEEE Robot Team from the University of Wisconsin – Madison would like to present a newly designed robotic platform for entry into the 14th annual Intelligent Ground Vehicle Competition. The **Wi**sconsin **R**obotic **E**xploration **D**evice (WIRED) will represent the team's third appearance at the competition.

# 2. Innovations

In creating WIRED, the team sought to improve on the weaknesses of last year's entry, Whitespace, while maintaining the many successful features. The team developed a brand new robust chassis for more logical component distribution, added protection and easy maintenance. New high-torque motors were used in order to improve speed and control characteristics of the vehicle. A redesigned power system featuring long-lasting batteries and dedicated power supplies was incorporated. The introduction of a laser range finder expanded our sensory abilities, while a newly developed embedded system provided a flexible interface between the computer and sensors. Lastly, a foldable mast was chosen over last year's tripod support for ease of transportation.

# 3. Design Process

In the development of WIRED, the team followed the basic process as detailed in Figure 3-1. We began by identifying the requirements for the competition as well as evaluated the previous years' design strengths and weaknesses. The team researched other approaches to the given problems and used the information to complement our own experience. The initial brainstorming was done by a small group of members who attended last year's competition and later discussed with the full team. We selected approaches that have worked well in the past, augmenting them with new solutions for previous shortcomings. The team used a top-down approach in designing systems, specifying a general structure first and subsequently working out specific details. When each design was sufficiently analyzed, a prototype was developed allowing preliminary testing of each system. Though careful planning and thorough testing, each system was ensured to function correctly before integration into the final product.

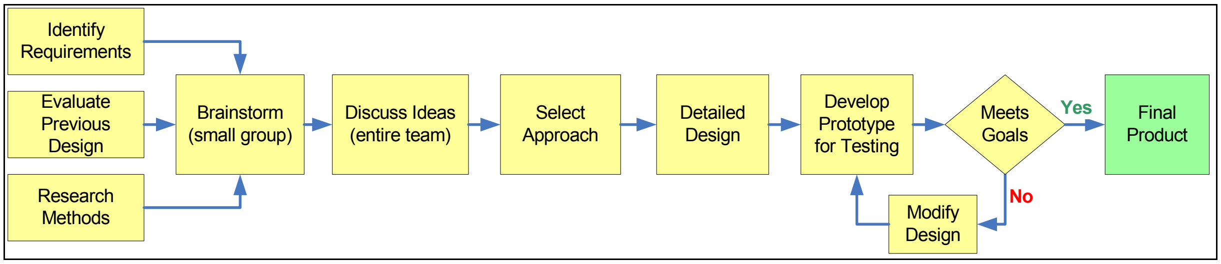


**Figure 3-1 Design Process**

### *3.1. Team Organization*

To accomplish our tasks efficiently the team was divided into four subgroups (mechanical, power, embedded electronics and software) as illustrated in Figure 3-2. The hierarchical structure of leadership allowed groups to focus on specific sub-projects, while leaders coordinated their efforts to ensure compatibility and functionality of all robot systems. Each group followed a strategy similar to the overall design process, but on a more detailed level.

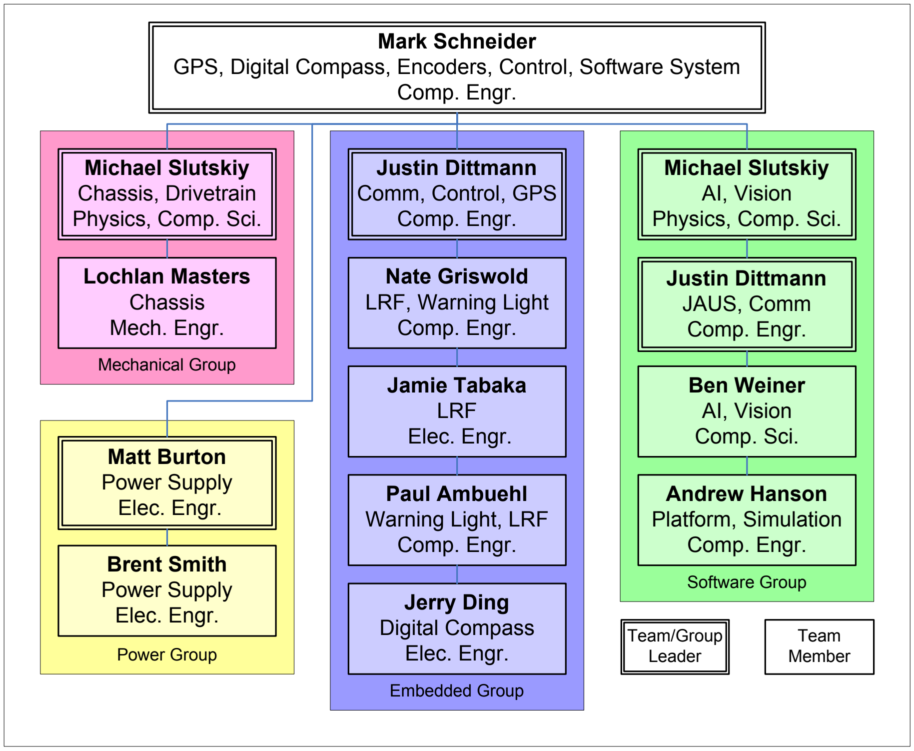


**Figure 3-2 Team Organization**

# 4. Mechanical Design

Learning from past experiences, the team chose to design the mechanics for the new vehicle from the ground up. Among the many modifications and improvements from the previous years' robots, WIRED's frameless chassis and reliable drivetrain are notable. As in past years, special care was taken to ensure safety, reliability and value in every step of the design. Before manufacturing, a full 3D model of the robot was completed using SolidWorks software and component failure analysis was performed using COSMOS Express to ensure a minimum Factor of Safety of 10. Computer-aided design enabled the team to plan and layout every mechanical feature, greatly reducing production time and costs.

## 4.1. Chassis

WIRED is built on a custom designed chassis that is logically divided into five compartments: left and right motor housings, battery compartment, laptop drawer, and electronics enclosure. The chassis is a hybrid combination of an exoskeleton body and an internal frame developed for superb rigidity and unobstructed access to even innermost components.

The main body sections are housed between two 0.125-inch thick steel sheets that function as the main structural members for the body with lateral reinforcement provided by square aluminum tubing. The front, rear and bottom walls of the robot are made out of thin lightweight sheet metal providing excellent protection for the internal components without significantly increasing the weight.

### 4.1.1. Main Compartments

The main body contains three sections that house the majority of the robot's internal components. The largest compartment contains two lead-acid batteries and is located in the forward half of the robot. This places the bulk of the weight slightly behind and below the drive axles of the robot providing excellent traction and balance. The compartment is recessed, sitting merely 6 inches above the ground, in order to keep a low center of gravity while maintaining sufficient ground clearance. The batteries can be charged through the dedicated weatherproof charging port, and can also be easily accessed for maintenance or removal via the top door.

Above the batteries, mounted on two 18-inch slides, is the laptop drawer. The enclosure is a modified aluminum display case with reinforced corners and dual latches for added protection. The lid of the case is made out of transparent Plexiglas, which in the future could be used with a Tablet PC to provide monitoring of the software without opening the enclosure. The drawer design was chosen to facilitate quick access to the computer without compromising the safety of other components. The enclosure retracts fully into the body of WIRED, stowing the laptop in a safe and space-efficient manner. The charging port described earlier also facilitates simple charging of the laptop battery making removal (though trivial) unnecessary in most cases. To allow proper ventilation of the enclosure cooling fans were added, while cable ports simplified connection of the laptop's USB, serial and power cables.

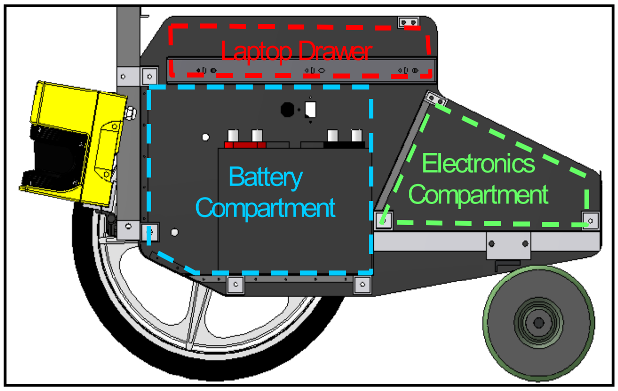


**Figure 4-0 Main Internal Compartments**

The rear compartment of the main body houses the power supplies, speed controllers and embedded electronics. The components sit several inches above the battery section for more convenient user access and to give ample space for the caster wheel mounted beneath. The electronics can be monitored through the transparent Plexiglas lid while maintaining a weatherproof seal. If the need arises, the lid opens freely allowing full access to any component.

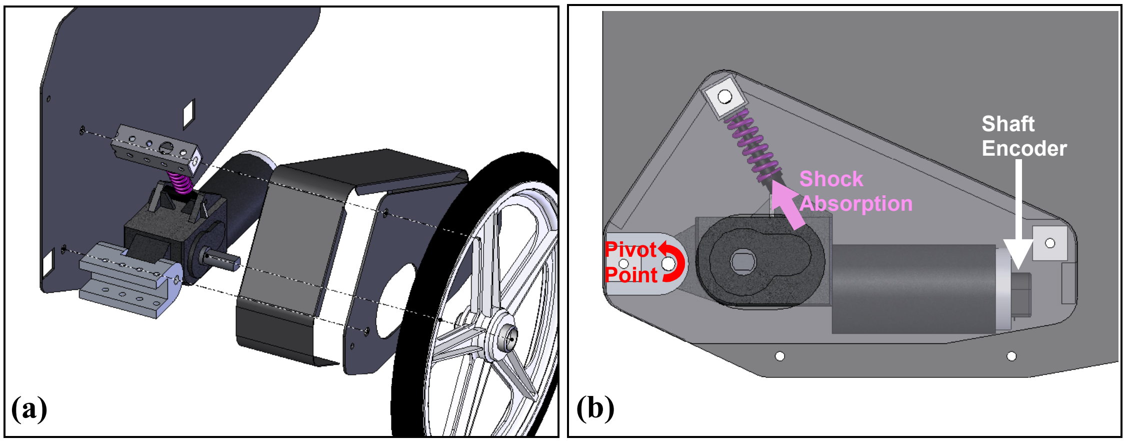


Figure 4-1 Left Motor Compartment (a)Exploded view (b) Side view

Each motor is mounted in a protective compartment sharing a steel wall with the main body (Figure 4-1a). Rather than being rigidly attached to the robot, the motors are mounted using a spring suspension that drastically reduces shock and vibration from terrain roughness (Figure 4-1b). The motors are intentionally placed exterior to the main section since they do not require the same level of protection from the elements as other components. Furthermore, isolating actively moving components is an added safety measure for the system.

### *4.1.2. Mast*

The foldable mast assembly made out of dual rectangular aluminum tubes provides a rigid support for the camera, GPS and digital compass, placing them just below the 6-foot height limit in a weatherproof “crow’s nest” compartment. This provides a large field of view for the downward facing camera: at an approximate angle of 30° from the horizontal the FOV is approximately 10 feet in front of the robot. Additionally, the compass’s location protects it from fluctuating magnetic fields generated by the DC motors while the GPS is positioned high to improve signal quality. A warning light is located in the midsection of the mast and is easily visible from both the front and rear of the vehicle. The SICK laser range finder (LRF) is placed at the base of the mast approximately 10 inches from the ground. The mounting mechanism allows the LRF’s angle to be adjusted if the need arises.

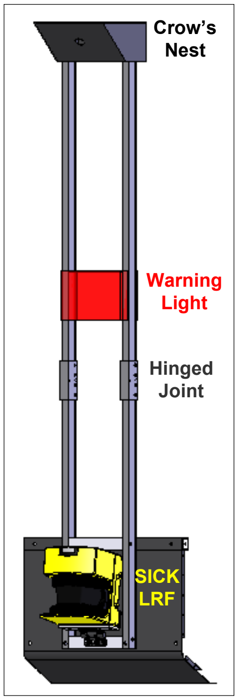


Figure 4-2 Mast Assembly

### *4.2. Drive System*

WIRED is propelled by two 20-inch MAG wheels driven by powerful 24V right-angle gearmotors similar to the type used on many commercial wheelchairs. A 7.5-inch caster-wheel is located in the rear of the vehicle to provide a third point of stability. The built-in gearbox on the motors allows for a compact and weatherproof drivetrain with a direct connection between the motor and drive-wheel, requiring minimal maintenance. The vehicle is differentially steered by controlling each wheel individually with feedback provided by the attached optical shaft encoders (Figure 4-1b, see Section 6.3 for a detailed description of control). This drive system has proved successful in the past and provides excellent power and maneuverability. WIRED can easily climb steep inclines, move quickly and precisely, and carry over one hundred pounds of payload. These and other vehicle statistics are given in Table 9-1 WIRED Statistics.

## 5. Power System

The power system design process began by determining the voltage and current requirements of our systems. Three DC voltage levels are necessary for the various components: 24V, 5V and 3.3V (see Table 5-1). Since power for WIRED is supplied by two 12V deep-cycle lead-acid batteries, a dedicated step-down supply was sought to produce 5V and 3.3V. Although the batteries in series can provide the 24V requirement, current draw from the motors and battery charge depletion would produce instabilities in the supply. Therefore, a regulated 24V supply was essential to power the LRF.

Since battery lifetime is critical for a mobile vehicle, linear voltage regulators (an inexpensive and common solution for DC-DC power regulation) were not used this year because of their low efficiency and high heat dissipation. Instead, the team developed high-efficiency switching power supplies in order to meet WIRED's needs (detailed in the following table).

| **Power Supply** | **Component** | **Current (mA)** | **Power (W)** |
|---|---|---|---|
| **Regulated5V** | GPS-RadioShack DigiTraveler | 75 | 0.375 |
| | AVR Microcontroller (4) | 6 | 0.030 |
| | HCTL2017 Quadrature Decoder (2) | 3.4 | 0.017 |
| | HEDS5540 Encoder (2) | 19.1 | 0.096 |
| | Receiver | 11.20 | 0.056 |
| | RS232 Transceiver | 13.85 | 0.070 |
| | LCD | 1.20 | 0.006 |
| | Voltage Display | 16.20 | 0.081 |
| | Compass-PNI Vector 2x | 10 | 0.050 |
| **Regulated 3.3V** | TLC5940 (LED Driver) (2) | 12.0 | 0.040 |
| | S-Flux LED (32) | 28.48 | 0.094 |
| **Regulated 24V** | SICK LRF | 800 | 30 |
| **24V Direct Battery Connect** | Victor 883 Speed Controllers (2) | 10 | 0.2 |

**Table 5-1 Power Requirements**

# 6. Electronics Design

## 6.1. Sensors

### 6.1.1. Camera

The camera selected by the team, Creative WebCam Live! Ultra, features a 640x480 CCD sensor capable of 30 frames per second. The images are transmitted directly to WIRED's computer over USB2.0. While the camera is a very versatile sensor for a wide variety obstacles (line markings, sand pits or objects), it can be detrimentally affected by shadows and changing light conditions and often needs to be supported by other sensors.

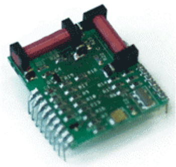
### 6.1.2. SICK Laser Range Finder

An addition to our sensor array this year is a SICK PLS101-112 laser range finder (LRF), which interfaces directly with the computer via an RS232 serial port. The device provides 180° single-plane sweep of the area in front of the robot with 0.5° angular and less than 70 mm (2.75 inches) radial resolution. Although the maximum scanning range of the SICK is almost 50 meters (164 feet), the scan radius is restricted to 10 feet coincide with the camera's field of view. Although the LRF is a much more reliable sensor when it comes to detecting solid bodies, it is completely blind to lane boundaries or potholes.

### 6.1.3. GPS

The GPS unit the team is again using this year is the RadioShack DigiTraveler. Although the unit is not as accurate as many higher priced systems, it supplements the other positioning sensors greatly reducing accumulated error.

### 6.1.4. Compass



The digital compass the team selected is the Vector 2x from PNI. Being inexpensive, low power and having a resolution of 1 degree at 2.5Hz, the Vector 2x is an ideal solution for our needs.

### 6.1.5. Encoders

Agilent HEDS-5500 optical shaft encoders were chosen for motion control feedback and positioning data. These feature two-channel quadrature output with 1024 CPR (counts per revolution) resolution. Two 16-bit quadrature decoders (Avago HCTL2017) interpret signals from the encoders into meaningful data. The primary function of this sensor is to provide responsive motor feedback ensuring precise motion. Additionally, the data is used to supplement the GPS/Compass positioning data.

## 6.2. Computer

In the past, the team has used custom-built computers to maximize computing performance. The key disadvantage with these machines is their high power consumption resulting in poor battery life. As a result, we recognized the necessity for an alternative, lower power solution. The team chose a Dell Inspiron 600m with a 2.0GHz Pentium M Processor with 1GB of DDR memory for its processing needs.

## 6.3. Embedded System

To interface the computer with the low-bandwidth sensors an embedded system was designed. Although we initially considered a single processor solution, a distributed system was chosen for better modularity and extensibility. Using the low-cost, high-performance Atmel ATmega8 series microcontrollers (AVR) for each embedded task provided a flexible interface between necessary devices and the computer. The AVRs feature hardware-supported pulse-width modulation (PWM) used for motor control and standard communication protocols ($I^2C$, SPI) for easy component interconnectivity.

In order to simplify development, libraries were written to establish a stable and modular code-base common to all the networked AVRs. These libraries included communication routines to send and receive data over the Serial Peripheral Interface (SPI) used with the digital compass and the warning light drivers, RS232 for receiving GPS data and communicating with the laptop, PWM generation code to set motor speeds and routines to read the encoder data. Once this common library was created, the microcontrollers could be easily programmed further for their specific tasks.

To coordinate the data flow through the network, a custom protocol was developed to identify the data structure of the packets and to establish a unique address for each microcontroller. A dedicated AVR, the network master, functions as a transparent link between the computer (via an RS232 serial port) and the network slave devices (over $I^2C$, the Inter-IC bus). The goal of the master is to provide a reliable interface at a low latency without impeding the primary operation of the other AVRs on the network. The master routes requests from the computer to the appropriate microcontroller and returns the sensor data to the software (Figure 6-1).

In particular, WIRED’s network consists of three slave AVRs responsible for motor control, data acquisition from the compass and GPS, and warning light operation. The motor controller AVR converts heading and speed specified by software into appropriate PWM signals for the speed controllers. Additionally, a closed feedback loop from the data supplied by the encoders is used by this AVR to maintain a constant speed and to orient the vehicle during turns. The second slave AVR parses the ASCII data stream from the GPS unit and polls the digital compass for positioning data. A supplementary function of this microcontroller is the interface to a character LCD screen used for debugging purposes. Lastly, a third AVR operates the current monitoring LED drivers over an SPI-compatible interface. The driver ICs control the S-Flux LEDs used in the warning light.

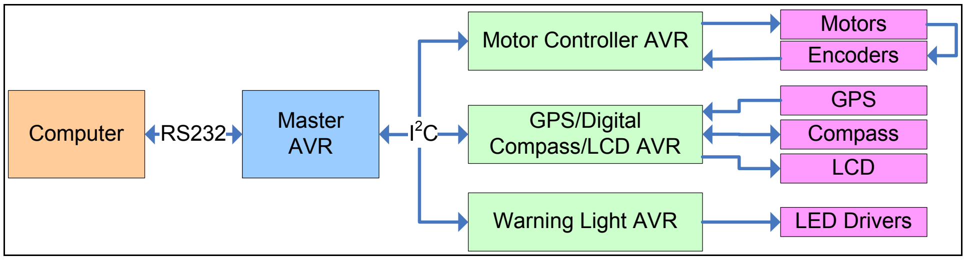


**Figure 6-1 Embedded Network Structure**

### 6.4. Safety Measures

Safety was a consideration in every step of the design, and, in addition to thoroughly testing every system on WIRED, several measures were implemented. First, a front and rear warning light alerts people of the robot's presence. Using an array of 32 ultra-bright S-Flux LEDs, the light provides both high visibility and low power consumption. The device is programmable allowing different output configurations to make it more noticeable and for debugging purposes. Also, we have integrated two emergency stop (e-stop) circuits into WIRED's systems: one hardwired pushbutton located at the rear of the vehicle and one wirelessly activated relay switch. Both e-stops physically break the connection between the batteries and the speed controllers bringing the robot to a quick halt. The wireless e-stop adds security and convenience when the onboard e-stop is out of reach. Lastly, to comply with IGVC rules and for general safety, the vehicle's speed is restricted to a maximum 5 mph at all times by the motor control AVR. These safety precautions guarantee that the vehicle remains under full control of the operator without posing a danger to bystanders or itself.

## 7. Software Design

### 7.1. Overview

Our main control application was developed in Java and based on the open-source Robot Simulation and Control Lab (RSCL) framework. The main advantages of using Java are its cross-platform compatibility and its ability to allow for modular design. Building on the software developed over the past several years, we created a consistent system for obstacle avoidance and navigation that is easily extensible for new tasks.

### 7.2. Robot Simulation and Control Lab

RSCL is an open source robotics framework that spawned from the team's work in previous years. It provides the necessary backend components to simulate and control a robotic vehicle including a convenient graphical interface for debugging, a 3D simulation structure for extensive testing and modular organization for ease of programming. This software platform makes it easy to concurrently program as a team and provides a reusable and reliable framework.

## 7.3. Sensory Obstacle Detection System

### 7.3.1. Vision

Following previous years' successes, the team again chose to utilize the Java Media Framework (JMF) and Java Advanced Imaging API (JAI) for consistent and efficient image acquisition and manipulation respectively. In particular, the JAI native libraries provide numerous compiled (and therefore fast) image-processing operations that were vital in developing the vision system.

Prior to data extraction, the raw image undergoes several filters and transformations that attempt to take advantage of the multitude of sensory information present. Noting that many obstacles are distinctly colored, the software passes the image through a filter that singles out particular hues and converts them to white or black depending on whether they must be avoided or not. After much experimentation, we chose the HSI (Hue, Saturation, Intensity) color space for such filtering. HSI is more convenient than RGB when lighting conditions vary since mostly the saturation and intensity values are affected. Next, the image is converted back to RGB and the blue plane is extracted. Taking into account overall lighting conditions, a threshold value is calculated dynamically for each frame. The image is then converted into a binary (black and white) map. In order to eliminate speckling and "salt and pepper" noise, an edge preserving median filter is applied. Lastly, to compensate for the angled perspective of the camera, the image is transformed into a top-down view.

At this point the image is ready to be analyzed to determine areas free of obstacles. A duplicate binary image is also used for line detection, described in more detail in Section 7.5.3. Through lengthy calibration and fine-tuning of the algorithms the vision system has become a reliable means of detecting almost any area that WIRED must not venture into.

### 7.3.2. Laser Range Finder

Because our vision system does not differentiate between solid objects and ground markings, this year a SICK laser range finder (LRF) was added to the obstacle detection system. The software communicates with the LRF using the Java Communications API (javacomm). The LRF transmits data continuously, and the software complements the visual information with the LRF data after each scan. The data provided by this sensor is analogous to the data we extract the images. However, the process is simplified greatly since no preprocessing is required. The sensor provides reliable physical data, with the downside of not detecting potholes or lane markings. When combined with vision data, this increases the reliability of the obstacle detection system and enables WIRED to determine whether there is a physical object in front of it or not.

### 7.3.3. Sensory Data Fusion

In order to reconcile possibly conflicting information, we developed a paradigm where readings from an arbitrary number of obstacle sensors are combined into a single set of allowed paths. The only requirement is that sensor data conforms to a specified format: in our case, gaps between obstacles in the sensor's field of view. After raw data from each sensor is converted to this format, a moderator system considers each set of gaps and outputs a more reliable combined version of the input data. The fusion process is analogous to a logical AND operation: if all the sensors agree that an area is clear of obstructions, the gap is added to the output set. On the other hand if any sensor detects an obstacle in an area, that section is considered an impasse for WIRED.

Since there are situations where sensors may not agree on any open areas a deadlock prevention mechanism is necessary. Our solution was to assign each sensor a priority corresponding to its perceived reliability. In case of conflict between two sensors, the moderating system defaults to the higher priority data. In our scheme, the LRF can only detect solid bodies, while the camera is capable of also detecting lines, potholes and sandpits. Therefore, the camera is deemed as providing more reliable obstacle data. The priority of the sensor systems could be changed dynamically by the program during runtime to compensate for varying external conditions that may render a sensor ineffective. This technique is used in the Navigation Challenge when the robot is sufficiently close to a waypoint to ignore the white circle marking the location. Although the vision would not allow the vehicle to cross a line, its "suggestion" is intentionally ignored in this case.

## 7.4. Positioning System

### 7.4.1. Strategy

WIRED relies on the GPS and digital compass in conjunction with encoder data to determine its position and heading accurately. Since GPS and compass data is fairly straightforward, no additional software processing is required. A dedicated system continuously updates the change in position based on the encoder counts using a circular trajectory approximation for the robot's motion. This information is subsequently combined and used in motion planning and mapping. The sensor redundancy allows the combined data to minimize error inherent to the sensors and such approximate calculations.

### 7.4.2. Positioning Data Fusion

The strategy for meaningful reconciliation of complementary data comes from observations about each of the sensors' weaknesses. The GPS, for example, is not as precise over small displacements as the encoders and has a much slower update frequency. Thus, a natural solution is to use encoder data for short intervals and when the GPS data is unavailable (due to lag or signal loss). To eliminate a common problem of error accumulation from wheel slip, the position is periodically "reset" by the GPS as the data becomes available. This allows us to take advantage of both sensors reducing error and response time of the system.

Similarly the heading as determined by the compass can be verified and complemented by processing encoder counts from each wheel. This helps in situations where the compass may lag or be affected by inclined terrain. The redundancy of data makes the overall system more reliable, and, in using the encoders as a backup to the compass, the team truly used the sensor to its fullest potential since the principal reason for their addition was motion feedback for embedded controls.

## 7.5. Robot Navigation

### 7.5.1. Strategy

In developing the strategy for the IGVC challenges, the team sought and successfully developed a comprehensive solution that requires minimal modifications between Navigation and Autonomous modes of operation. It was established that data acquisition does not differ significantly in the two modes as objects must be avoided and accurate position must be determined regardless of the robot's goals. Therefore, the techniques described in the two preceding sections are employed identically in both challenges.

Initially, it was thought that the motion planning was entirely disparate for the two modes. However, later an innovative technique was developed allowing lane following to be cast as waypoint navigation. This method greatly simplified software design as the Navigation Challenge code was reused for the Autonomous Challenge. The following sections describe in detail the software strategy for waypoint navigation and the modification necessary for lane following. A visual summary of the system is offered in Figure 7-1.

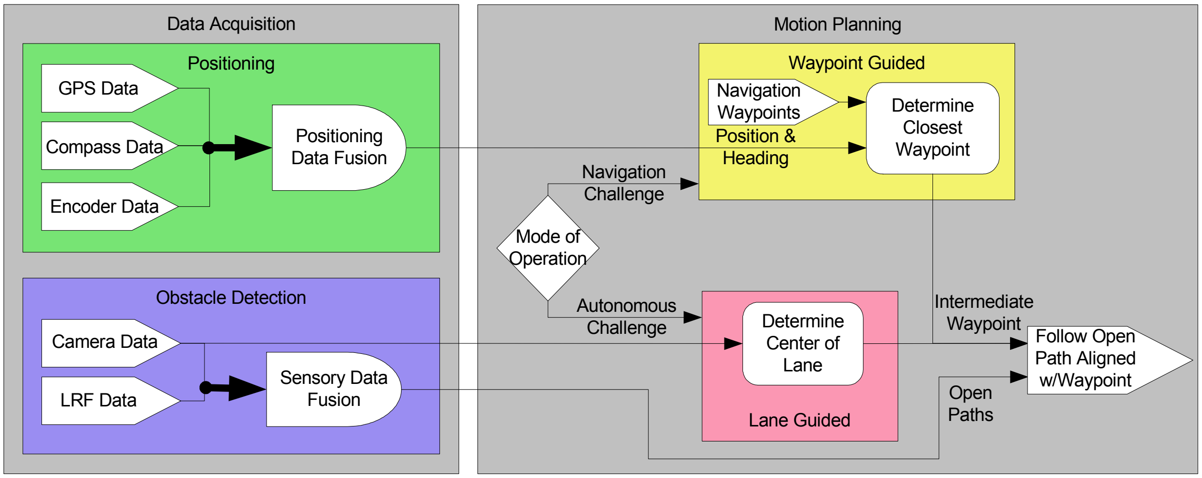


**Figure 7-1: Functional Software Structure**

### *7.5.2. Navigation Challenge*

Past experiences have shown that oftentimes a simple general strategy works exceedingly well in the majority of circumstances and deadlock cases are often resolved naturally due to the randomness inherent to the surrounding world. For example, it is, for all intents and purposes, impossible to approach an obstacle in such a way that the open path to one side is identical to the other side. Even if for an instant the robot perceived such a situation, the smallest action would force the robot out of this unstable equilibrium and down one of the two paths. Of course, cases can be constructed where such a naïve approach will fail, but let us not worry about these yet (in Section 7.5.4 we will present a strategy for handling unusual situations).

Keeping with this philosophy, it was decided that there are three prerequisites for successful waypoint navigation: effective obstacle avoidance, accurate positioning and sufficient perseverance. Since the first two were taken care of by the sensory systems described earlier (Sections 7.3 and 7.4 respectively), all that remained was to “encourage” WIRED to maneuver to specified target locations instead of randomly driving around. Although given sufficiently accurate positioning and control it is not difficult to simply drive to a specified coordinates, this goal must never override the obstacle-avoidance algorithms. Clearly, the robot must only follow unobstructed paths in reaching its final destination, yet it must not be entirely thrown off-course by an obstacle.

This reasoning directly guided the development of our Navigation Challenge strategy. First, WIRED greedily prioritizes the list of waypoints based on their distances from its current location and chooses the closest as the next target. As the vehicle moves and its position changes, the priorities are continuously updated. In order to avoid indecision when two values are close, a nominal amount of hysteresis is introduced favoring the current target. Once the immediate goal is determined, WIRED evaluates each available path in its direct vicinity (as provided by sensory obstacle detection) and selects the one that will lead most directly to the intended target. Once a waypoint is reached, it is removed from the list and the algorithm only considers the remaining set, returning the robot to the start when no targets remain. This approach is simple, yet easily adapted to other prioritization methods.

### 7.5.3. Autonomous Challenge

Last year, our strategy for this part of competition was simply to let the robot move forward as it deemed safe. Since lines, virtual and real obstacles were treated equally, there was no danger of the vehicle leaving the course and the random nature of the sensor data naturally resolved all the situations we encountered in the event. In some ways this was a fortunate coincidence, as we simply did not face the few scenarios we knew might not be handled correctly. In particular, there was no strong impetus for the robot to move solely in the forward direction. The software would not intentionally go in backwards, but a sufficiently devious course layout could coax the robot to turn around and not be aware of the mistake.

What we needed was a means of persuading the vehicle to continue in a certain direction without overriding its other necessary functions. Upon further consideration, it was realized that the general goal should be to continue in the direction of the lane as defined by the left and right line boundaries. This could be accomplished with the strategy developed for the Navigation Challenge by creating a pseudo-waypoint directly in the middle of the lane some distance (approximately twice the range of the object detection range) ahead of the robot's current location. Assuming that the lane is generally smooth over short intervals, a continuation can be extrapolated using the Hough transform to calculate the equations of the lines. The visual data is scanned to determine whether the right or left line is visible and the center point is predicted accordingly. As the robot progresses, this target is overwritten by newer estimates.

Ideally, WIRED never reaches the pseudo-waypoint, but instead continues following it rather like a mule guided by a carrot on a stick. Of course, nothing is ever ideal and several measures were taken to prevent mistakes. First, the waypoint is "aged" and eventually becomes invalid. This prevents WIRED from getting stuck in an attempt to reach an impossible goal. Additionally, if the boundary lines cannot be detected the vehicle will continue following the predicted path until the latest target is reached. Under our assumption, the robot should not choose a gap in the dashed line marking as it would likely lead away from the predicted center of the lane. Lastly, instead of stopping if the target is reached as in the Navigation Challenge, the robot simply reverts to exploring the world around it in a manner affected only by the obstacles it encounters until it is stopped or locates a lane boundary to follow. Since the lanes are usually narrower than the distance to the pseudo-waypoint, it is probable

WIRED will see a line while still on its way to the target (in the correct direction) and so chances of turning around are minimized. The figure below summarizes the software systems.

### *7.5.4. Mapping and Special Situations*

The mapping system uses open path information, current position and heading to keep track of WIRED's progress through the environment. By storing the intermediate waypoints chosen by the navigation system and the available paths not taken, a tree-structure is created. In case of a dead-end, WIRED can retrace its movements to the last junction in the tree and choose a different path to follow. This system can also be used to learn about the surrounding world and repeatedly follow successful paths. Since the waypoint data is used only as a guide for the motion, the vehicle can correctly adjust to new obstacles in a previously open path and safely avoid collisions.

## *7.6. 3D Simulation Environment*

To ensure that our software would be operational when our mechanical design was complete for the robot, 3D simulation software was written to replace real world inputs and outputs with simulated ones. The simulation software used the Xith3D Java extension that allows creation of an OpenGL virtual world complete with the models of a varied terrain, 3D obstacles and line boundaries similar to those that the autonomous and navigation competitions would present the robot. This simulated world was used to generate artificial vision, laser range finder, GPS, compass and encoder data for the robotics application and responded appropriately to the control signals that were calculated from this data input.

# 8. Joint Architecture for Unmanned Systems (JAUS)

WIRED was designed from the ground up with JAUS compliance in mind. Using OJJAUS, an open-source project which implements JAUS in Java, the software is able to send and respond to JAUS-compliant messages over any Ethernet or 802.11 interface. To implement JAUS, a dedicated team member studied the JAUS reference documents in their entirety to obtain a thorough understanding of the protocols and methods involved and to convey the necessary ideas to the rest of the team. Each sub-group then developed their systems with JAUS-compliance in mind. For example, the software uses the coordinate system defined by JAUS and many inner structures are developed to be analogous to their JAUS counterparts so there is no information loss in translation to JAUS specified format.

Although originally the team planned to use JAUS for every communications interface on the vehicle (RS232, $I^2C$, and SPI), complete JAUS support was dropped when our designs started to cater to JAUS at the expense unnecessary complexity and restrictions for the embedded system. Since a complete implementation of a JAUS interpreter would take well over four kilobytes of storage available on the ATmega8 microcontrollers, a two-protocol solution was initially proposed: JAUS would be used for communication between the computer and a communications master, while the master would communicate with the embedded systems network over $I^2C$ with a custom-designed protocol. However, the JAUS-to-$I^2C$ protocol converter was eventually dropped in favor of full support of the custom $I^2C$ protocol, similar to JAUS.

In accordance with IGVC requirements, the vehicle is programmed to handle three JAUS messages: Resume (Command Code 0004h), Standby (Command Code 0003h), and Set Discrete Devices (Command Code 0406h). Standby and Resume commands suspend and restart WIRED's operation, while Set Discrete Devices activates the warning light.

# 9. Predicted Performance

During the design process, numerous performance targets and vehicle constraints were established for the new robot both from the analysis of IGVC rules and characteristics desired by the team. Last year the vehicle's speed and battery life were markedly sub-par and, therefore, were a particular focus of efforts this year. Careful component selection, CAD design and extensive testing played a crucial role in meeting and exceeding requirements. The table below summarizes the team's goals and WIRED's measured performance in each category.

| **Tests** | Target | Measured |
|---|---|---|
| Vehicle Height | Max 78 inches (6 feet) | 77.5 inches |
| Vehicle Length | Min 36 inches (3 feet) | 39 inches |
| Vehicle Width | Max 36 inches (3 feet) | 35 inches |
| Battery Life | Min 1 hour | >3 hours |
| Laptop Battery Life | Min 1 hour | 1.5 hours |
| Maximum Speed | 5 mph | Restricted to 5mph |
| Stopping Distance (down 15% grade) | Max 6 feet | <3 feet |
| Ramp climbing ability | Min 15% grade | >30% grade |
| Obstacle detection distance | 10ft | Restricted to 10ft |

**Table 9-1 WIRED Statistics**

# 10. Other Design Considerations

Beyond safety and solid mechanical, electrical and software design, the team kept in mind a few other key design considerations, striving for a robust and rugged vehicle while keeping costs low.

## *10.1. Reliability and Durability*

Each system within WIRED was built with reliability and durability in mind. Reliability within the robot's software and electronics was achieved through rigorous testing of each system before final integration. Durability of the robot stems from the careful component selection and attention to strong mechanical design.

### *10.2. Cost*

The cost of creating a robot from scratch can be a substantial investment. Many of the parts and material our team used for WIRED were either donated or on loan.

| Category | Item | Qty. | Tot. Price | Team Cost |
|---|---|---|---|---|
| **Computer** | Dell Inspiron 600m | 1 | \$600 | \$600 |
| **Power** | 12V deep-cycle lead-acid | 4 | \$400 | \$0 |
| | Charger | 1 | \$300 | \$0 |
| | Power Supply components | 3 | \$45 | \$30 |
| **Chassis** | Sheet steel (1/8") | 20sq.ft | \$450 | \$0 |
| | Sheet metal (.04") | 20sq.ft | \$100 | \$0 |
| | Square aluminum tubing | 12ft | \$40 | \$40 |
| | Rectangular aluminum tubing | 16ft | \$60 | \$60 |
| | Wheels | 2 | \$100 | \$0 |
| | Caster | 1 | \$20 | \$0 |
| | Misc. Hardware | | \$300 | \$200 |
| **Motors** | 24V right-angle gearmotor | 2 | \$100 | \$80 |
| **Electronics** | Victor 883, 24V Speed Controllers | 2 | \$298 | \$0 |
| | Creative WebCam Live! Ultra | 1 | \$100 | \$100 |
| | GPS-RadioShack DigiTraveler | 1 | \$99 | \$0 |
| | Compass-PNI Vector 2x | 1 | \$50 | \$0 |
| | ATmega88 | 4 | \$12 | \$0 |
| | S-Flux LEDs | 32 | \$32 | \$0 |
| | Quadrature Shaft Encoders | 2 | \$120 | \$0 |
| | Quadrature Decoders | 2 | \$30 | \$0 |
| | SICK PLS | 1 | \$4,500 | \$0 |
| | Misc. Electronics | | \$100 | \$100 |
| **Totals** | | | \$7,856 | \$1200 |

**Table 10-1 Team Expenditures**

# 11. Conclusion

The IEEE Robot Team has brought together a diverse team of students to design an autonomous robot, WIRED, to compete in the 14th Annual Intelligent Ground Vehicle Competition. The team developed the robot to exceed all specifications while holding to our team goals. We feel that through participation in this project and competition we have expanded student knowledge and experience while creating a safe, robust and competitive intelligent vehicle.