Donatello

IGVC Design Report 2009

robotics @ MARYLAND

Team Members:
Antonio Busalacchi, John Garvey, Lucas Hedinger, Dave Holub,
Shaun McElhinney, Ryan Mukherjee, Rohit Ramesh, Rob Thomson,
Ken Tossell, Mark Walsh, Sam Winter, Alan Yang

Faculty Advisor:
Dr. Nuno Martins, Assistant Professor: Department of Electrical and
Computer Engineering and the Institute for Systems Research

Faculty Statement

I, Dr. Nuno Martins, assistant professor in the Department of Electrical and Computer Engineering and the Institute for Systems Research at the University of Maryland, do hereby certify that the engineering design of the autonomous ground vehicle Donatello has required significant work equal to that of a two-credit course.

Signed, Date

____
Nuno Martins

Table of Contents

Executive Summary

The University of Maryland's undergraduate robotics club recently formed a team of students to design and create an autonomous ground vehicle. The freshman-through-junior student team has been able to fund itself through progressive and active fund-raising. After raising funds, the team was organized into three smaller groups (Control Systems, Embedded Systems, and Structures) to design the robot in the spring of 2009.

Due to lack of experience, the team sought out advice from robotics labs, professors, and engineering sponsors for initial design advice and design review. These conversations led to the design of a robot chassis based on an all-terrain vehicle. Additions and modifications made to the chassis were designed with modularity, adaptability, and functionality in mind. The innovative changes to the structure include improved turn radius, autonomous control integration of drive and steering, and modular sensor mounts.

After identifying a base to create the platform, the Embedded Systems and Control Systems teams met to decide necessary sensory input to allow the robot to complete the objectives outlined in the IGVC rules. These sensors identified were stereovision cameras, SONAR modules, a GPS signal receiver, and a compass. Robotics@Maryland advisers suggested an addition of a LADAR scanner, though the team's initial budget was insufficient for its purchase.

With the addition of these sensors, the team created a power distribution system based on a 12V source and a 36V source that will provide four or more hours of runtime under load. The power system allows for safety and conservation of energy using a pair of circuit breakers and a switch board.

With sensors identified, the controls team was able to begin designing algorithms and systems that would integrate the data with autonomous logic and navigation. The team designed a three-stage system that receives raw data and compiles the data into obstacle and position information to store in a spatial data structure. The compiled data are then polled by a planner component, which finds the safest and most effective path to reach the next goal.

The team members have agreed that the design and integration process has been a unique and unparalleled experience within their academic careers. The team has benefited through application of knowledge from classes, business networking, experience within a large multidisciplinary team, and improved engineering ability.

Team Overview

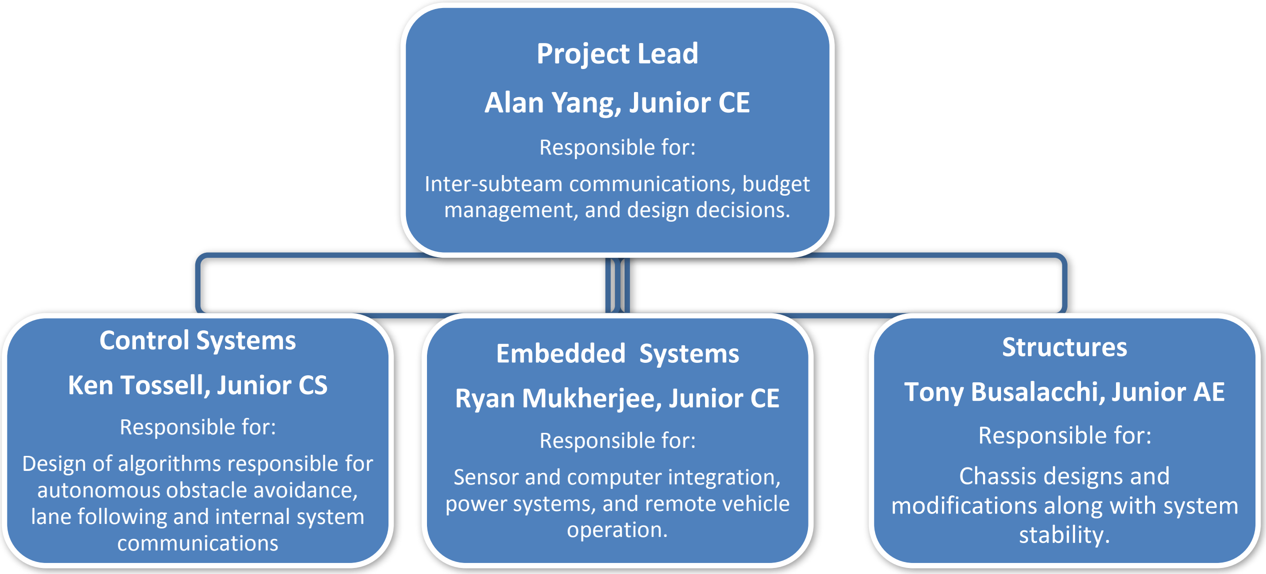
Introduction

Robotics@Maryland (R@M), a University of Maryland-sponsored organization, consists of two teams: the autonomous ground vehicle (AGV) team and the autonomous underwater vehicle (AUV) team. The AUV team has been internationally recognized as the winner of the summer 2008 AUVSI competition. Its success spawned a desire to expand R@M to incorporate a competitive land vehicle team. In the spring of 2008, an AGV team was formed; the next fall, it placed second in a local autonomous ground vehicle speedway competition. The young team has now decided to participate in a larger, more renowned competition, the IGVC.

The AGV team consists of undergraduate students who range from first-year to third-year, all of whom have at most moderate robotics experience. The latter characteristic has allowed for a unique learning experience for the team in terms of design, implementation, and testing.

Budget Acquisition

Being a completely student-organized club with a single faculty advisor, Dr. Nuno Martins, budget acquisition was a responsibility placed on the club and project leaders. Through successful demonstration of creativity and substantial project design work in combination with a history of past successes, the club was able to acquire sufficient funding for both of its teams.



Project Lead

Alan Yang, Junior CE

Responsible for:

Inter-subteam communications, budget management, and design decisions.

Control Systems

Ken Tossell, Junior CS

Responsible for:

Design of algorithms responsible for autonomous obstacle avoidance, lane following and internal system communications

Embedded Systems

Ryan Mukherjee, Junior CE

Responsible for:

Sensor and computer integration, power systems, and remote vehicle operation.

Structures

Tony Busalacchi, Junior AE

Responsible for:

Chassis designs and modifications along with system stability.

Figure 1 - Team Structure (CE= Computer Engineering, AE= Aerospace Engineering, CS= Computer Science)

Team Structure

To facilitate the AGV project, a team leader and three subteam leaders were chosen. To determine an appropriate team structure (described in Figure 1), the team decided an analysis of objectives and goals was necessary. Identification of objectives and goals were performed by first researching the competition rules, followed by a second iteration of the objectives. The second iteration allowed the team to identify internal goals for each major objective of the competition. These goals were then placed into a matrix (Figure 2) based on time and difficulty. Tasks identified in the high-difficulty, low-time category include objectives members had experience with. This matrix allowed team members to prioritize and distribute their time among the objectives.

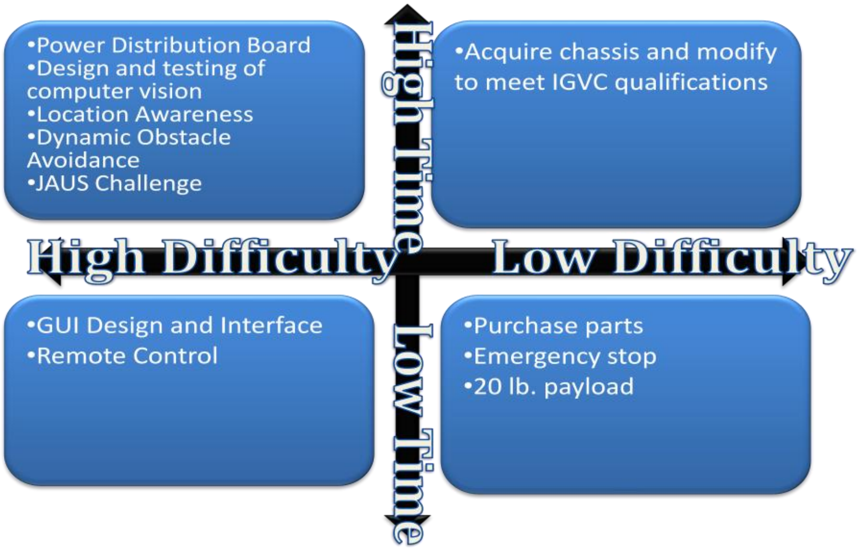


Figure 2 - Matrix of Objectives

Design Process

With a newly formed team, design was the biggest challenge. Only a couple of the members had any robotics experience, so the team sought advice from faculty. The team met with the Robotics Lab of the University of Maryland to inspect their AGV chassis. A majority of the vehicles were modified electric wheelchairs with modified differential drive. Unfortunately, a new wheelchair chassis was out of budget, and a used wheelchair was difficult to obtain. After additional research, the team identified an electric ATV within the budget and sufficient for its needs. After obtaining a base chassis, the team drafted a preliminary design of sensors, electronics, and structural modifications. The following were identified as initial conceptual suggestions:

Structure

- Custom computer case with false floors and sides for batteries and wires.
- High torque servo for steering
- Minimal modification to base ATV chassis

Sensors

- Sonar for object registration
- Stereo vision for line detection
- Compass and GPS for directional/location navigation

Electronics

- Sensors powered through motherboard
- DC-DC Power supply supported by 11.1V LiPos

These basic concepts were communicated to the Robotics@Maryland AUV team and General Dynamics Robotic Systems engineers, who provided great feedback and suggestions. The final design is documented in the following sections.

Chassis

In choosing a chassis, the team weighed several factors. These factors were based on the team's ability, options' costs, IGVC qualification requirements, modularity, and adaptability. Through research of successful AGVs and advice of faculty members and sponsoring mentors, the team identified three chassis opportunities and weighed each against a list of criteria (shown in Table 1).

Criteria	Electric ATV	Electric Wheelchair	Robotics Chassis[1]
Can hold at least 100 lbs	✓	✓	✓
Ease of modification	✓		✓
At most 1.5 m turning radius		✓	✓
Less than $600	✓		
Meets dimension requirements	✓	✓	✓
Minimal modification needed	✓		✓
Scale 15% gradient	✓	✓	✓
Transportable	✓	✓	✓
Travel at least 5 mph	✓	✓	✓
Adaptable for future use	✓		✓

Table 1 - Structures Criteria

After weighing these factors, the team was able to determine that an electric all-terrain vehicle would be an optimal choice. The only limitation was the turn radius offered on a stock

---

[1] This category included considerations of premade robotics chassis such as the RobotShop Robot Rover and Segway RMP-400

ATV. The initial turn radius was about 3 meters, but through removal of pivot barriers placed on the axle, Donatello's turn radius was decreased to 1.7 meters. Because the course will have a minimum five-foot turning radius (about 1.5 meters), the team determined this was a sufficient reduction.

The ATV has a wheel base of 67 cm, and it is 1.04 meters long, with a height of 63 cm (excluding the camera mount). The most significant benefits the ATV offers are its modularity, its adaptability, and its low cost. An ATV is designed to travel across many types of terrain with various types of riders and purposes, so it has an inherent need for adaptability. Also, the ATV offers a chassis that is built solidly, with few barriers for adjustments and additions. Finally, the cost of the ATV was a third of that of the other options. These three advantages outweighed the minor challenge of adjusting the turn radius.

Sensor Mount

The innovation of the structure is introduced in the addition of the vehicle's autonomous "eyes." The required sensors needed to be interfaced with the base ATV chassis in a manner that would allow future disassembly, modification, and adaptability. In addition to these general criteria, each part of the design factors in stability and weather tolerance as additional design constraints.

Optical Sensors

The housing for the cameras was designed with three main factors in mind: weatherproofing, modular design and stable alignment of the two cameras. The team needed this sensor to achieve stereovision, so having a design that locked in a pre-calibrated system was most efficient. The Embedded Systems team calibrated the stereovision through use of a checker board diagram and provided angles necessary in final placement. The cameras are locked into position on the optical sensor block and sealed under a plastic shield. The shield is sealed with weatherproof glue. By locking the cameras on all axes, Donatello can maintain camera calibration.

LADAR and SONAR

These two mounts have a very simple but effective design. The primary goal was to allow for manual and automatic adjustments of the position of the SONAR modules and the LADAR ranger. These components are moved using a servo attached to each module.

Electronic Systems

Introduction

The electronic systems on Donatello were designed to be modular, extensible, robust, cost-effective, and easy to debug. Each system can be individually monitored via a wireless debugging client or operated standalone without the presence of the integrating controls software. This form of module isolation provides ease of debugging as well as a framework for the development of additional systems. Costs are identified in Appendix A.

Sensors

Donatello uses a variety of sensor systems to provide its artificial intelligence software with a clear picture of the world around it. While each individual sensor system might not be among the most accurate, each is determined based on its cost effectiveness and its ability to supplement other sensors. This allows the team to reduce costs significantly while still providing information about the environment equivalent to that of much more expensive systems.

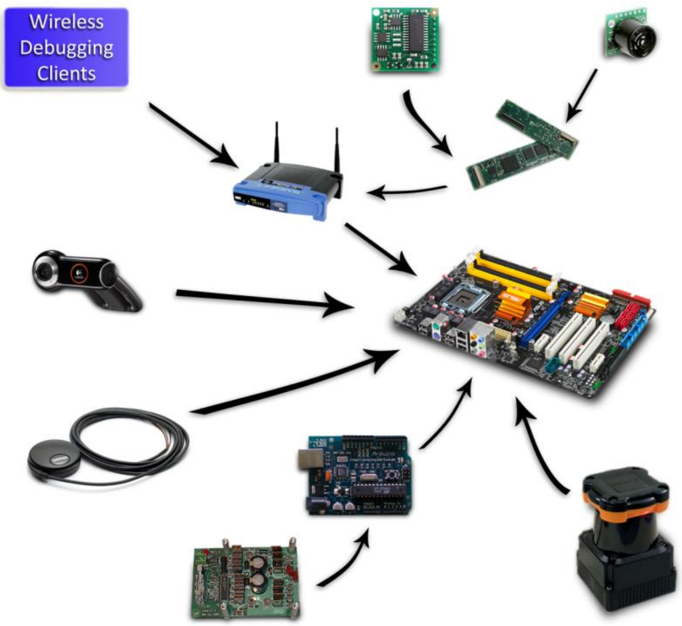


Figure 3 - Sensor Integration

Logitech QuickCam Pro 9000

Logitech's QuickCam Pro 9000 was chosen based on its relatively low price of $99, its ease of integration using a standard USB port and Linux drivers, its high resolution support (frames up to 1600x1200 pixels) and the accurate image capture possible using the built-in Carl Zeiss lens. Donatello uses Logitech's QuickCam 9000 cameras in pairs to provide stereo vision support for the artificial intelligence component. Stereo vision allows for more accurate color information, glare reduction as a result of multiple camera angles, and obstacle distance information that supplements the data from other sensors.

Hokuyo UTM-30LX

Hokuyo's UTM-30LX laser range finder was chosen as a superior alternative to the commonly used SICK range finders. Hokuyo's UTM-30LX has a scanning frequency of about 40Hz, an effective range of 30 meters at 270°, a maximum measurement error of ±50mm at 30m, and relatively low power consumption of 1A at 12VDC. Hokuyo's laser range finder also offers an easy-to-use serial interface, further complementing its competitive price. The team uses the laser range finder for the primary obstacle detection system.

Maxbotix LV-MaxSonar-EZ1

Maxbotix's sonar modules were chosen based on their accuracy, narrow beam width, and ease of integration with other systems. Each module provides output in multiple forms as well as continuous or triggered operation for maximum compatibility. Donatello uses four Maxbotix sonar modules mounted on rotating servos above each wheel to maximize close-object detection and minimize the possibility of object collision.

Garmin GPS 18x LVC

Garmin's GPS 18x LVC receiver was chosen because it provides accurate GPS positioning data with a relatively high refresh rate of 1Hz at a low cost. Garmin's built-in Wide Area Augmentation System (WAAS) capabilities bring accuracy to within 3 meters, and use of a Kalman filter further increases the accuracy of the positioning data to provide reliable and accurate position information for the Navigation Challenge.

Devantech CMPS03

Devantech's CMPS03 compass module was chosen for its low cost, high accuracy, and ease of integration. Devantech's compass has an accuracy of about 3° after calibration and is easily interfaced via I^2C. Donatello uses the compass to supplement its GPS positioning data in the Navigation Challenge and to prevent the robot traveling backwards in the Autonomous Challenge.

Remote Control

Texas Instruments MSP430F2013

To handle PWM decoding for remote control of the vehicle, Donatello uses two MSP430F2013 low-power, low-cost microcontrollers. Isolated from other sensor and control modules, the MSP430s decode PWM input from an RC receiver and determine whether or not to stop/override artificial intelligence control of the vehicle.

Futaba 2-Channel FM Pistol Radio Transmitter/Receiver

To remotely drive the vehicle, the team uses a standard Futaba 2-channel FM pistol radio, which is generally used by hobbyists for remote-controlled cars. This provides an easy to use interface for driving the vehicle while also taking advantage of reliability and range developments in the RC field. This setup gives us more than enough range to meet the wireless e-stop requirements.

Power

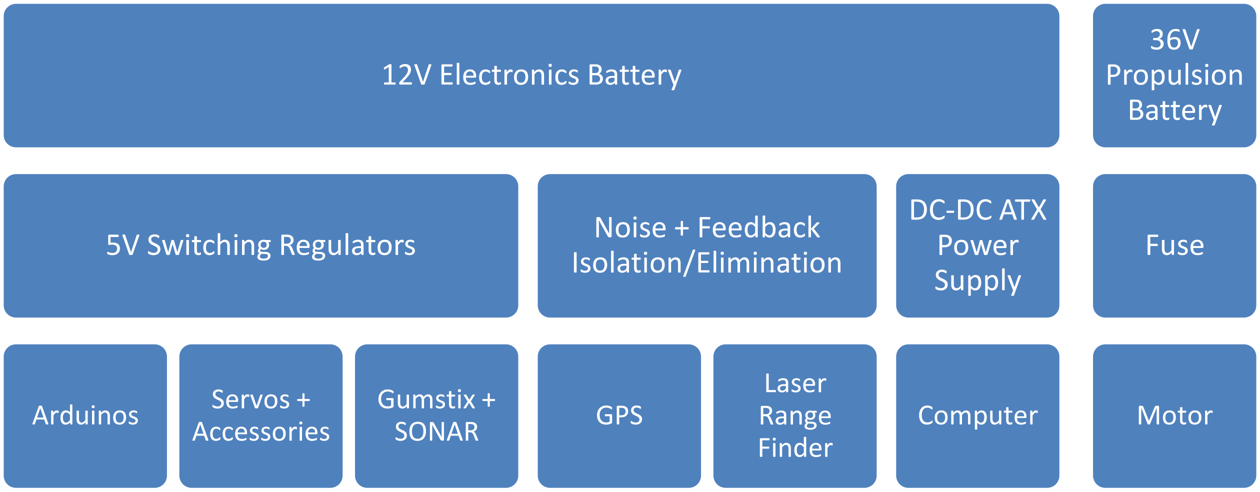


Figure 4 - Power Distribution

To power the necessary components on the vehicle and provide maximum power isolation, Donatello makes use of two power sources. One 12V high-capacity deep-cycle battery is used to provide power to the computational-based electrical systems (e.g., the computer, microcontrollers and sensors), and three 12V medium-capacity lead acid batteries are combined in series to provide 36V to the propulsion systems. Under extreme load, the electronics battery should last about 5 hours, while the propulsion battery will allow Donatello to travel about 15 miles.

To prevent unnecessary power dissipation and accidental short-circuiting during testing, Donatello's regulating circuits have the ability to be manually disabled via switches. Several inline fuses have also been included between various key systems to prevent unlikely damage, and circuit breakers have been placed in locations where excessive current draws are more probable (to reduce the likelihood of not having the required fuse during operation).

National Semiconductor LM22670

To provide regulated power to the various electronic systems on Donatello, the team uses several LM22670 Buck type switching regulators. Each regulator provides one 5V output and can support a maximum load current of 2A. At about 92% peak efficiency, the LM22670s help increase Donatello's runtime.

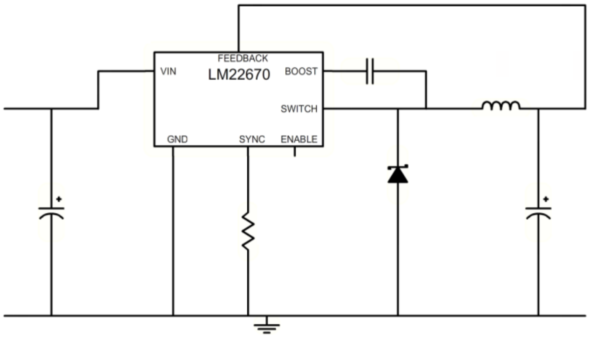


Figure 5 - LM22670 Voltage Regulator Circuit

To minimize noise between each of the various electronic components as well as to provide all necessary current, the team used five separate voltage regulators.

West Mountain Radio Super PWRgate PG40S

West Mountain Radio’s Super PWRgate allows the team to effortlessly and instantaneously switch between wall power and battery power. While connected to the wall, the PWRgate recharges onboard batteries and supplies power to the electronics, and once it is disconnected from wall power, the robot runs solely off batteries.

Mini-box.com M4-ATX

Mini-box.com’s M4-ATX DC-DC power supply allows the team to provide power from the 12V electronics battery to the onboard computer. In doing so, the M4-ATX provides active monitoring of the battery’s voltage, allowing the team to prevent deep discharge.

Control Systems

Primary Information Sources

Donatello obtains its knowledge of the outside world using a combination of passive sensors, locally active sensors and geocentric orientation sensors. Information from these sources is communicated through a logic system as described and summarized in Figure 6 and the subsequent sections.

Our passive sensors include a roll measurement device as well as color vision for secondary motion registration, following marked paths, and identifying obstacles that cannot be discovered through depth detection. Our locally active sensors consist of a 2-D planar laser range finder for medium-to-long-distance object detection along with a SONAR array whose main task is the quick and reliable identification of nearby obstacles. Finally, Donatello uses a GPS location device for waypoint navigation and a digital compass for local direction assistance during waypoint-based navigation.

Localization and Mapping

The mapper, an independent component, is responsible for storing obstacle position information, for answering spatial obstacle queries, for determining where on the map to place new objects, and for resolving conflicts between old and new sensor data. Each point of information Donatello collects is evaluated and registered in the mapping system before it may be processed by any goal-oriented planning or navigation component. The vehicle's data entry points, such as SONAR contacts and compass headings, offer frequent, unrequested raw observations to the mapping system. Upon receipt of one of these raw observations, the mapping system estimates the vehicle's position based on prior scene and motion information, inserting the new datum into the map and modifying its conception of the course with respect to the system's existing knowledge[2].

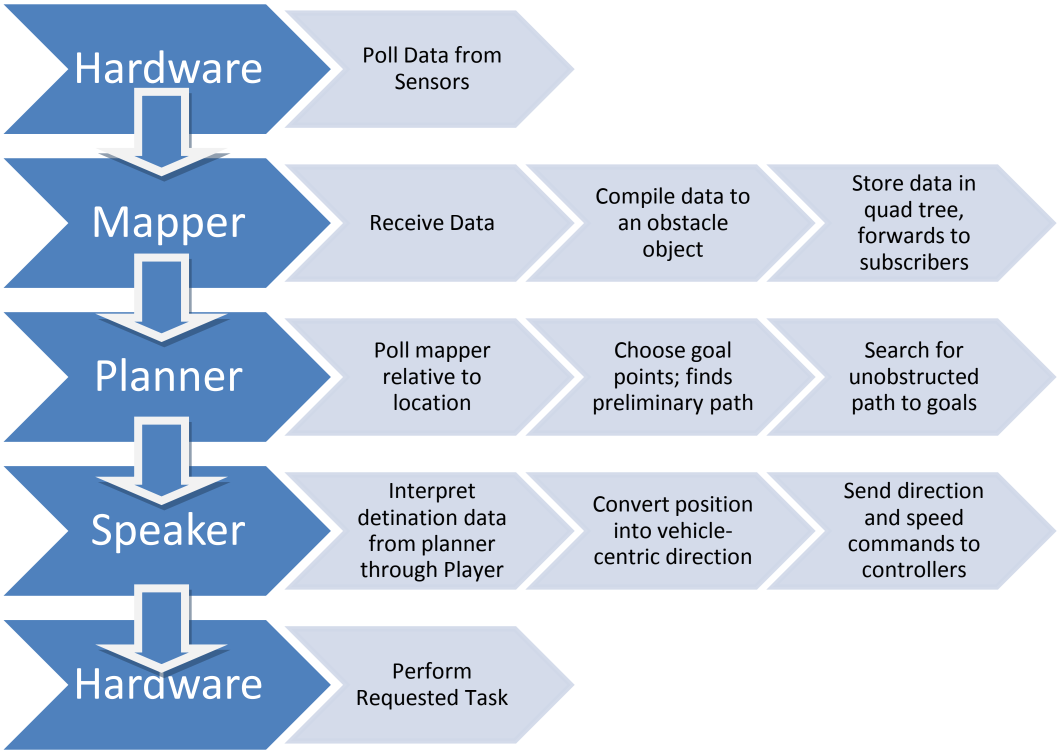


Figure 6 - Control Logic

The map is composed of 100 cm^2 regions, each of which contains a list of obstacles that intersect the given region. The vehicle occupies about 70 regions (100 cm by 70 cm). The map is represented with sparse data in a quadtree, which allows us to store all possible landmark points while still being able to quickly and efficiently search the set of occupied locations. Upon insertion and deletion, for questions of leaf node splitting and consolidation, the mapper compares the leaves' sorted lists of intersecting objects.

Laser Range Finder

The laser range finder provides a list of distances to objects from the sensor, indicating the position at every 0.25 degrees, covering a 270-degree view. Donatello's laser processing software removes the object-free points and extremely nearby points (which often indicate measurement error). Donatello converts the points' angles to positions on the map and stores groups of points as obstacles.

---

[2] The vision system requires already-perceived scene data in addition to raw observations; this is implemented in such a way that there is an information cycle between the vision system and the mapping system. The vision processor receives scene information developed from its own observations as well as from those of other sensors.

Vision

For the IGVC Autonomous Challenge, Donatello is required to find its way through a course with boundaries that are marked with white paint lines. Donatello collects images from two cameras that are pointed at 30 degrees left/right from forward and 30 degrees down and extracts the white areas. Next, the vision system identifies connected white components and evaluates the components' dimensions: If a component's size is smaller than a set value, it is ignored for the remainder of the boundary detection process (it will be stored and later considered as a possible pothole). If a component is narrow enough, it is treated as a boundary segment. If it is not narrow and it appears to touch both sides of the course, it is registered as a sand area that must be crossed. Such a segment is then broken into three: The left-side connected to the boundary path and the right-side are submitted for further processing as boundaries, while the center area is ignored as it is not important for navigation.

Once all boundary segments have been identified, the vision system identifies nearby boundary ends and stitches them together. When discontinuities remain, the system will bridge gaps if they are small enough and close enough to be sure they do not represent valid paths. If the laser scanner has identified a standing barrier at or above the position of a discontinuity, the vision system assumes there is a segment hidden below the barrier and connects the segments beside it. After segments have been finalized, the vision system sends the boundary data to the mapping system along with the locations of any potholes that were detected.

Planning

Donatello's planning system is split into three components: a high-level planner, a mid-level, and a low-level planner.

The high-level planner considers remaining checkpoints (or other goals) and chooses the best goal to pursue based on preset goal ranks as well as goal proximity. In doing so, it determines the map position of the vehicle's destination. The high-level planner does not use any obstacle information.

The mid-level planner attempts to find a path to the newly found destination that avoids any obstacles that may be listed on the map. In most cases, the mid-level planner's path is the path the robot will take.

The low-level planner uses data received directly from the sensor array (not through the mapping system) and watches for close-range obstacles. If any obstacles appear, the low-level planner may reroute or stop the vehicle; its main purpose is to avoid collisions in the case of main sensor or planning failure.

Controls Hardware

All medium and high-level controls computation is done on a single computer, which is equipped with a 2.4 GHz Core 2 Quad CPU and 4 GB of RAM. The operating system and controls software are stored entirely in memory at runtime, preventing damage to the persistent storage device. The logging functions operate on a remote laptop, with primary data (e.g., camera and ranger output); secondary data (e.g., map contents, planner output and raw or calculated position information); and run metadata (e.g., CPU usage, temperature, network usage and processing speed), all being transferred to a Player-subscribed log receiver.

Communication Layer

Donatello’s communication systems are organized around the Player robotics communication framework. We have implemented a position-control driver that exposes Donatello's speed and direction controls to any Player-aware component. The driver accepts "position2d" commands from the planner via Player, and forwards relevant position data to drivers operating the Motion Mind and OSMC controllers over serial.

The planner is a Player client program, and it subscribes to the mapping system's "nearby obstacles" data feed as well as the output of the mapper's combined position awareness function (based on GPS readings, output odometry and visual object recognition).

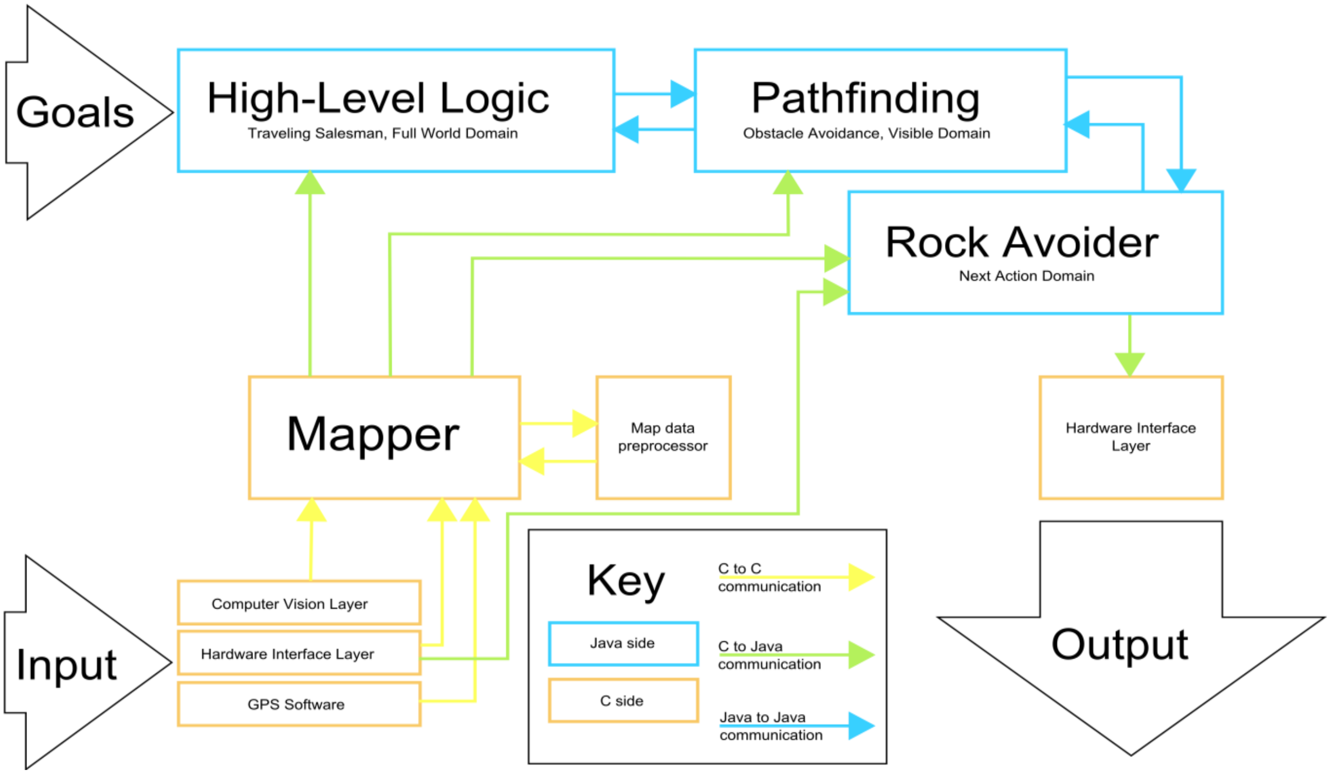


Figure 7 - Internal/External Communication

The vision processing component receives image and position data using the "camera" interface, and it implements the "ranger" interface to provide other components with course boundary and pothole information. The mapper accepts laser ranging data over Player, and in addition to its feed output, it can be queried for obstacles in a specified area.

It is also important to note the communication of internal software. Each module of the overall control logic passes its information described in Figure 7.

Appendix A – Team Expenses

Part	MSRP	R@M's **Cost**
Structure		
XA-750 ATV	$725	$500
Lockbox	$210	$210
Material Cost	$500	$500
Machining Costs	$1,000	$700
Electronics		
Hokuyo UTM30LX	$5,569	$4,775
Logitech QuickCam Pro 9000	$165	$165
Maxbotix EZ-1	$120	$120
CMPS03	$60	$0
Garmin GPS 18x LVC	$65	$65
Power Distribution Board	$100	$100
West Mountain Radio Super PWRgate PG40S	$165	$165
Texas Instruments MSP430F2013	$30	$30
Futaba 2-Channel FM Pistol Radio Transmitter/Receiver	$90	$0
Open Source Motor Controller	$220	$220
MotionMind Controller	$85	$85
Computer	$450	$450
12V, 80Ah Lead Acid Batteries	$270	$270
Misc. Hardware	$200	$200
Total	$10,054	$7,885

Table 2 – Expenses

Thank You to Our Sponsors

We would like to thank the University of Maryland's Electrical and Computer Engineering Department, Aerospace Engineering Department, Computer Science Department, and Institute of Systems Research for their generous funding. Also, special thanks to our corporate sponsors (Hokuyo, Lockheed Martin, Northrup Grumman, and GDRS) who have provided great advice as well as funding.

Appendix B – Team Member List

Team Member	Major	Subteam
Alan Yang	Computer Engineering	Control Systems
Ken Tossell	Computer Science	Control Systems
Lucas Hedinger	Computer Engineering	Control Systems
Rohit Ramesh	Computer Engineering	Control Systems
Rob Thomson	Computer Engineering	Embedded
Ryan Mukherjee	Computer Engineering	Embedded
Shaun McElhinney	Computer Engineering	Embedded
John Garvey	Electrical Engineering	Embedded
Tony Busalacchi	Aerospace Engineering	Structures
Dave Holub	Electrical Engineering	Structures
Sam Winter	Civil Engineering	Structures
Mark Walsh	Aerospace Engineering	Structures
Total Hours Contributed	2288 (Approx. 143 hours/week)	

Table 3 - Team Members