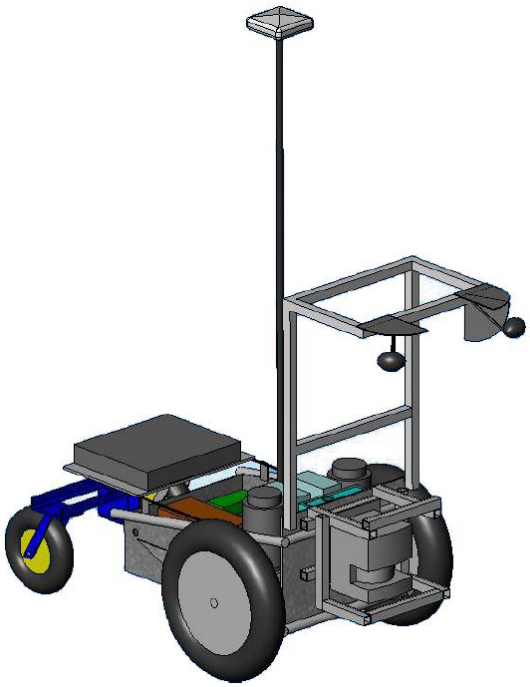
Q 2010 Design Report

Adam A. Wright '10, Nathan M. Swaim '10, Orko Momin '10, Paul Wortman '10
Rahul R. Shakya '11, Young Ho Shin '11, Prasanna Gautam '11
Steve Petkovsek '12, Adam Norton '12

Faculty Advisor: Dr. David J. Ahlgren

May 14, 2010



Faculty Statement This is to certify that Q has undergone significant redesign in both hardware and software from last years IGVC entry. The Q team members worked on the robot as an Independent Study project and received 1.0 credit (3 credit hours) per semester. This project is significant and has led to many senior design projects in both Computer Science and Engineering.

-Dr. David J. Ahlgren, Karl W. Hallden Professor of Engineering, Trinity College

Contents

1 Introduction

The Trinity College Robot Study Team presents its fourth iteration of Q, an autonomous vehicle for the Intelligent Ground Vehicle Competition (IGVC). Based on an all-terrain wheelchair chassis, Q is a rugged robot. This year the team has continued to improve Q, with a greater focus on usability, speed and safety. The team has also turned Q into a fully embedded system with the addition of a National Instruments Embedded Vision System (EVS).

2 Innovations

The innovations this year were focused on two main concerns: robustness and quick reaction time. The Q 2009 system consisted of a mixture of embedded systems and one Dell laptop. The central controller of the robot was a National Instruments CompactRIO (RIO stands for Reconfigurable Input and Output), which contains a 533MHz real-time controller and an FPGA arranged as co-processors. The cRIO was used to read from sensors and control motors. The laptop was connected to the cRIO using an Ethernet connection. Camera data acquisition and image processing for lane following during runs was done on the laptop. This scheme made development relatively convenient, but caused the team to blur the line between development and production. The Windows operating system on the laptop also became very cumbersome and was the source of many bugs in the system. As a result, the system was not very reliable or robust.

2.1 Embedded Vision System

This year, the laptop has been replaced by the National Instruments Embedded Vision System (EVS) (See Figure 1). This system has a dual core 1.66 MHz Intel processor with 1GB of RAM and a 1GB solid-state hard drive. The EVS runs LabVIEW Real-time under the VxWorks real-time operating system. By moving the line detection algorithm to the EVS, Q has become a fully

Figure 1: Embedded Vision System (EVS). There are several ports for Ethernet, a display output, USB storage, as well as IEEE1394 ports for cameras

embedded system. Now the team has adopted a development-production cycle, where code is tested until it works, and then finalized and downloaded to the real-time targets (the cRIO and EVS) on the robot. The overhead associated with the Windows operating system has also been eliminated, speeding up image processing.

2.2 Parallelized Software Architecture

The main software architecture in the cRIO that controls the robot has been rewritten to use a parallel architecture. Last year's software all ran in one finite state machine and a lot of processing time on the cRIO was not being used. To take full advantage of this embedded system, each component of the software has its own process. If one part of the algorithm at this iteration is waiting for sensor data or some other piece of information, the algorithm at the next iteration can begin running. This will speed up the execution of the robot, resulting in faster reaction times to obstacles. This system is also easier to debug and trouble shoot, since parts of the software are separated into threads. The software can be loaded with the main control process along with a debug/logging process, or with all of the sensor processes, or with only specific processes that must be tested.

2.3 Simplified Power System

The power distribution system for Q in the 2009 competition drew power from seven different batteries, including the laptop battery. This year, Q only uses two 12V batteries connected in series to provide 24V. Electronics are used to distribute the power to the various systems on Q.

2.4 More Usable Mechanical Design

Q's mechanical system has been overhauled to allow for easy access to batteries and allow for quick and easy servicing of the robot.

2.5 Control Panel

In addition to improved accessibility, a new hardware control panel used for operating the robot has been constructed. The panel facilitates operation of the robot by merely manipulating four switches and one GO button. Previously, to operate the robot, the operator had to make sure all seven batteries were charged, boot the laptop, and start the LabVIEW program in Windows. This process was cumbersome and made working on the robot difficult at IGVC last year.

3 Design Process

3.1 Team Organization

The team was organized into four broad groups based on the various task families within the scope of Q. These groups are defined as follows:

1. **Management** This group was responsible for leading the team and making all key design decisions. The group also managed all logistics and the planning for IGVC.

2. **Hardware** This group was responsible for maintaining and improving the mechanical and electrical power system of Q.

3. **Software** This group was responsible for software development. All programming from initialization, motor control, communication, to low level FPGA programming was done by this group.

4. **Testing** This group was responsible for testing the algorithms and performance of Q. The group was also responsible for creating diagnostic programs to expedite debugging.

Members were divided into these groups based on their expertise, interests and workload associated with each group. Most members contributed to more than just one group and occasionally rotated projects, this allowed members to gain an overall understanding of Q. All members were required to work together for a minimum of 3hr per week as well as attend weekly meetings to give progress reports and discuss their projects.

Every year the Q team loses several key senior members and welcomes new members many of whom are new of robotics. To help new members' integration into the team, a mentoring system has been implemented. Every new member is grouped with an incumbent member and work together in a specific project for Q. Furthermore, regular workshops were held on topics such as FPGA programming, Solidworks modeling etc to help members master wide variety of skills.

3.2 Design Methodology

The team followed an iterative design process to improve Q. The process started with a detailed failure analysis of Q's performance in the IGVC 2009. After understanding the faults of Q and analyzing IGVC rules, a detailed list of requirements was created. Based on these requirements, strategies were proposed and continually tested to reach a finalized implementation.

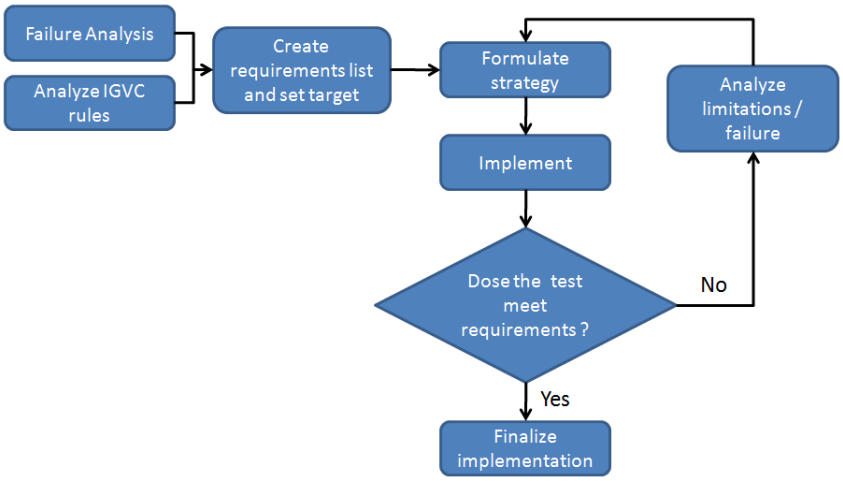


Figure 2: Iterative Design Process

4 Hardware

4.1 Chassis and Drivetrain

The physical platform of Q is a modified Permobil Trax all-terrain wheelchair. This frame can support a payload of over 250 lbs [1] and has a small footprint of 40" by 26". It features a differential front wheel drive system - a pair of 500W Leroy Somer MBT1141 motors - and a pair of rear mounted casters. The motors are geared with a 26:1 ratio, providing sufficient torque. Additional sensor and payload mounting frames were constructed using 80/20 extruded aluminum channels. The use of these channels allowed for quick and easy component layout without compromising mechanical strength. The camera mount was designed to fold down to reduce Qs volume during transportation.

The laptop mount used on Q last year partially covered up the underlying electronics and batteries, making it difficult to change batteries or troubleshoot electrical problems. This years mechanical design features a sliding tray on which both processors are mounted. The tray can be latched in two positions: The default position gives the user access to the batteries, and the alternate position gives the user access to the electronics. This way, the batteries can be easily accessed when running the robot and the electronics can be serviced without much more effort.

4.2 System Integration

The cRIO is the central control unit of Q. All motor control, data collection, navigation algorithms and sensor interfacing (with the exception of vision processing and cameras) occurs on the cRIO. Our latest addition to the system is the NI EVS which interfaces with the two webcams and executes image processing for lane detection. To expedite debugging, a Linksys WRT-54GL wireless router has also been added. The complete system diagram is as follows:

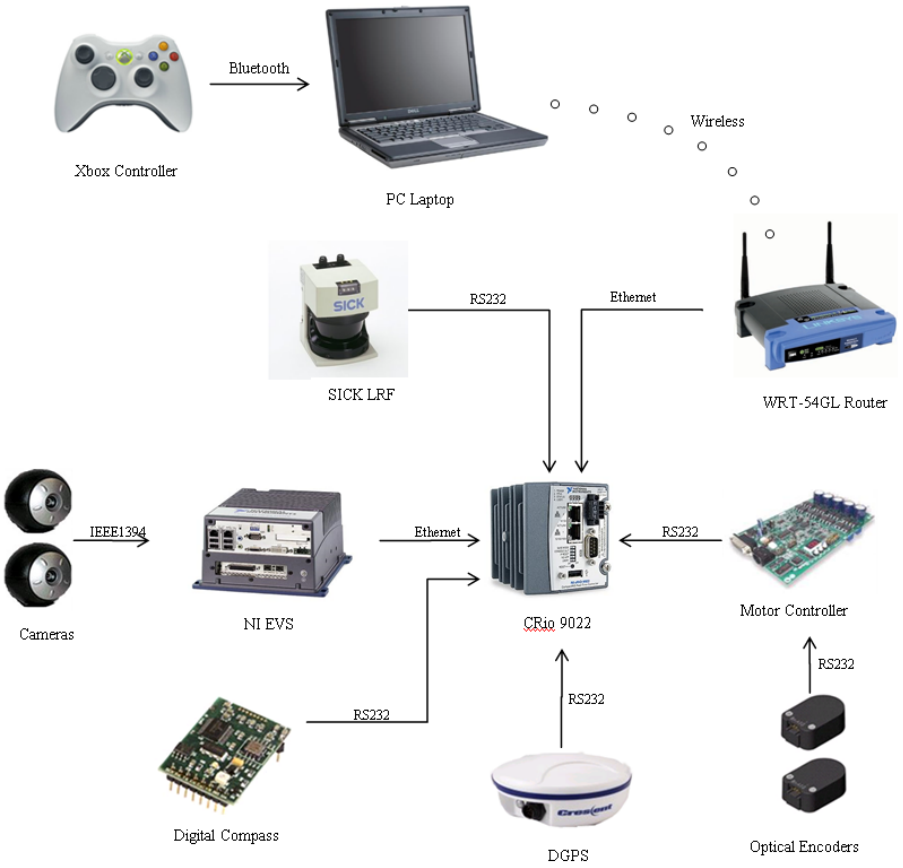


Figure 3: Diagram showing all the hardware components of the Q system and how they are integrated.

4.3 Control Panel

To allow easy control of the robot, physical hardware control panels have been added. A primary control panel allows easy operation of the robot when powered on. By simply initializing, turning off the motor safety and then pushing the GO button, the user is able to start a competition run.

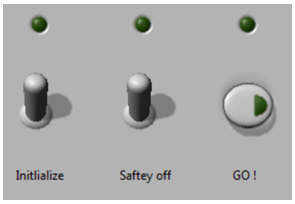


Figure 4: Primary Control Panel

An additional secondary control panel offers the user more options and also gives detailed dynamic feedback of Qs operation. Here, the user is able set competition challenge type and also switch to remote control and debug modes. An array of LEDs provides the user with real-time status of the

various sensors and the control program.

Figure 5: Secondary Control Panel

4.4 Power Supply

In the previous power supply system, 7 batteries were used to power the various components. This was a very inefficient system, since failure of any one battery could compromise an entire competition run. Thus the power supply was redesigned to use a minimal number of batteries. After analyzing the power requirements, a 24 V-22Ah battery pack was chosen as the power source for the robot. Under conservative estimates, this battery pack gives 1 to 2 hours of test time.

In addition to using new batteries, a protection circuit was also designed. It was designed to account for transients, high frequency feedback, and surges from the motors. The system consists of a 10 amp Radius Power filter model RP220-10-4.7. In addition to the filter, slow-blow fuses were used to further protect each device.

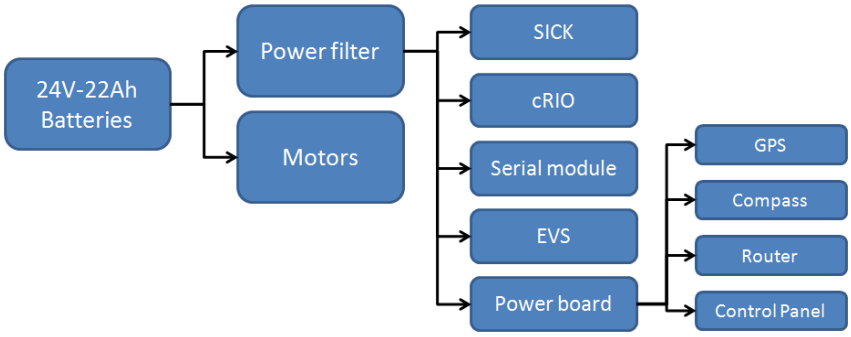


Figure 6: Diagram showing all the electrical components of the robot

Figure 6 shows the circuit diagram for the power supply. The 24 V battery pack is the only power source, and is connected directly to the motors and the filter. The output of the filter is distributed among the cRIO, cRIO Serial Module, Embedded Vision System, SICK, and the power board with fuses to protect each device.

5 Software

In light of poor software performance in last year's competition, the team has decided to update certain sections of the code running on the Q for better reliability, flexibility and execution speed. For example, the image-processing algorithm has been optimized for better frame rates and the software architecture has been radically redesigned to take advantage of the additional processing power. Additionally, new code has been written for various new system components and features. In the following sections, we will describe the design of the individual software components.

5.1 Software Architecture

The core of Q's control system is located in the cRIO and interfaces with most of the sensors and devices on the Q, including the motor controllers. All movement of the robot - autonomous or remote-controlled - is directed through this main software architecture.

Our goal in designing the main software architecture was to create a system with maximum performance and reliability, but also to make our software as intuitive and easy to work with as possible. To that end, our new design is based on a parallel architecture, where each component runs relatively independent of other components.

Our implementation of this architecture uses parallel loops, each containing code pertaining to one particular component of the robot. For instance, each of the sensors interfaced with the cRIO and the motor controllers is controlled by a separate loop. The sensor loops continuously update data buffers, while the motor controller loop waits for active commands from the main control loop. This is to ensure that there is absolutely no undesired movement of the robot.

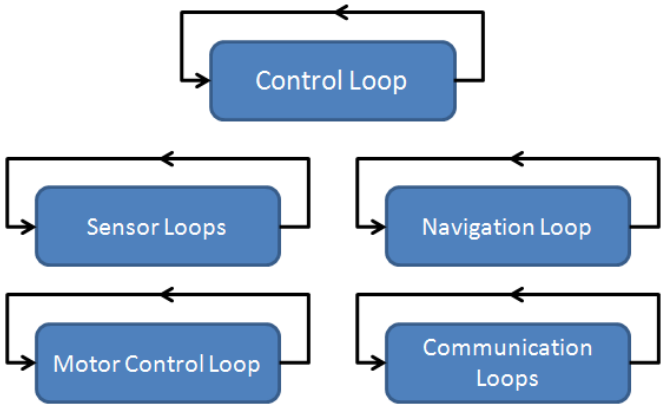


Figure 7: cRIO software architecture

The main control loop directs data flow between loops, thereby controlling the behavior of the system. For example, in remote control mode, motor commands received from the tablet PC are sent to the motor controller loop, while in autonomous mode, the motor commands are computed

by the VFH loop. Communication to the tablet PC and the Embedded Vision System is again achieved by using separate UDP loops.

The new parallel architecture has several advantages, for instance, it allows the overall throughput of the system to be much higher, allowing for smoother control of the robot. The reliability of the software seems to have also improved, as we have seen far less system crashes than last year. Furthermore, the new architecture has proved to be much more intuitive than its predecessor, since the overall structure is much more natural to understand. Having parallel loops has also allowed us to modularize the code, and work on separate parts individually, increasing efficiency and reducing the amount of time spent on piecing the various sections of code together.

5.2 Sensor Interfacing

The cameras are connected directly to the EVS's IEEE 1394 ports. LabVIEW provides library functions for capturing images from the cameras.

The GPS, compass and SICK LIDAR interface with the cRIO through the FPGA. Each of them connects to one of the four ports on the cRIO RS232 module, which is directly interfaced with the FPGA. The code to read the compass bearing is written entirely in the FPGA. The FPGA examines the bytes coming through the serial port until it sees the line feed character. At this point it begins reading the heading, pitch and roll, which are sent as "headin","pitch","roll". One example would be 100,15,33<LF> for a heading of 100 degrees, a pitch of 15 degrees, and a roll of 33 degrees. Each number is extracted as a string, then converted to an integer data type, and passed along to the real time controller.

The code to interpret data from the GPS runs on the real-time controller in the Compact RIO. The FPGA is programmed to act as a serial port. The GPS outputs longitude and latitude following the NMEA format. All latitude and longitude data are preceded by the string $GPGGA, so the GPS interpreter searches for that string, and then parses the following string for the latitude and longitude, which are converted to double floating point values and transmitted to the Compact RIO.

To read data from the SICK LIDAR, some initialization commands must be sent. Once these commands are sent, the SICK sends 360 bytes preceded by a 6 byte header. Those 360 bytes are 180 16 bit distance values in cm - one value for each degree, starting at -90 degrees from the heading and going until 90 degrees from the vehicle heading. The FPGA code searches for the 6 byte header and then reads each pair of bytes into some of the dedicated memory in the FPGA.

5.3 Communication

There are several components in the system that must communicate for the purpose of debugging, monitoring, parallel processing, and remote control of the robot. Figure 8 shows how all the processing components in the system are connected. The EVS and cRIO each have alternate

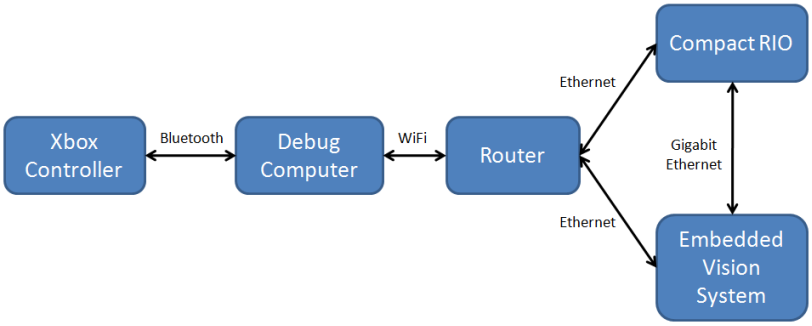


Figure 8: Communication system

Ethernet ports which are used for direct communication between the cRIO and the EVS. The primary Ethernet ports are used for the wired connection from the EVS and cRIO to the Linksys wireless router. The router is used for wireless debugging, development, and monitoring of the robot. A LabVIEW application for debugging was created that displays the information shown in Figure 9. Another application was made for remote control of the robot. The Xbox controller

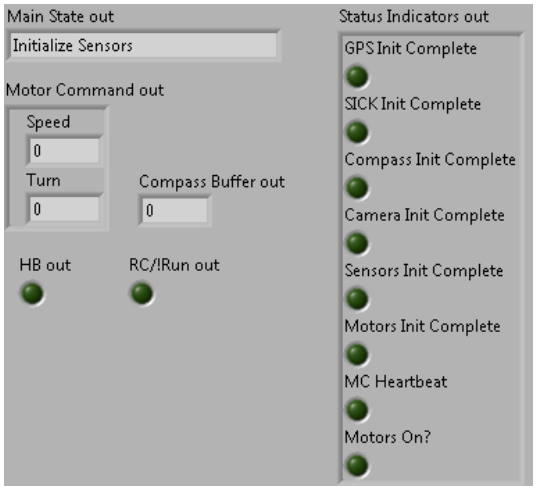


Figure 9: Debug Messages

communicates with the debug computer, which uses UDP to relay motor commands through the wireless router to the Compact RIO.

Every time the EVS processes a new pair of images, the obstacle data is sent over UDP to the Compact RIO. The RIO has a communication loop dedicated to receiving this obstacle data. When the data is received, it is en-queued on a data queue. When the Compact RIO detects a new batch

of obstacle data from the EVS, it immediately gets sensor data from the other sensors and runs its obstacle avoidance algorithm. By waiting for the image processing data (which is slowest of all the sensors) before running the algorithm, the data used for decision making is as new as possible.

5.4 Intelligence Algorithms: VFH and Navigation

A modified form of the Vector Field Histogram (VFH) algorithm was used for obstacle avoidance. Using a SICK LIDAR sensor, a compass, and camera data, a polar histogram representation of the obstacles is created and paths are planned accordingly. The algorithm utilizes a cost function which considers the target direction, wheel orientation and previous direction. When there is more than one opening between obstacles, this cost function is used to select the best candidate direction. The opening with the lowest cost is chosen. The result is a heading which represents the best path. Several improvements from previous years were instantiated. The scan, which covers 180 degrees, was split into 180 separate sectors, rather than 30, for path consideration for better resolution. Detected obstacles were widened to account for the robot radius and turning ability. Finally, the SICK data has been split into three distance thresholds, with the nearest threshold (at one meter) receiving priority in motor functions. The further two thresholds, at two and three meters, are considered in path planning. They influence the cost calculation in the same manner as the first threshold, with robot direction and target heading taken into account, but have less weight than the threshold at one meter.

5.5 Waypoint Navigation

There have been several improvements in the navigation algorithm of Q. After the GPS waypoint coordinates are obtained, Q creates an X-Y coordinate system with its current location as the origin and plots each waypoint with coordinates based on the distance of the waypoint from Q's original location. Q then moves toward the closest waypoint. When an obstacle is encountered, Q plots the obstacle on the X-Y graph and routes a path around it. When the waypoint is reached, Q approaches the next closest waypoint with obstacles that have been plotted considered in shortest distance calculations. If the closest waypoint changes when Q is navigating to a waypoint (i.e. going around a wall and increasing distance from the original waypoint) then Q will pursue the closer waypoint having plotted obstacles along the way. The GPS readings are accurate to within one meter.

5.6 Image Processing

The cameras have been positioned for maximal viewing range. The obtained images go through blue plane extraction since blue is the primary color in the RGB spectrum that is prominent in the

painted grass but not in the surrounding green grass or the barrels. Next normalization is performed to ensure that all images are of the same brightness before they are processed, regardless of actual lighting conditions. Texture masking then filters out shiny white objects to ensure barrels or other white objects are not mistaken as lanes. The algorithm goes through stages of morphology, particle filtering to remove noise from the picture. The resulting picture is a binary image containing only the line. Finally a lookup table is use to convert these images into distance values. Figure 10 illustrates the step by step process:

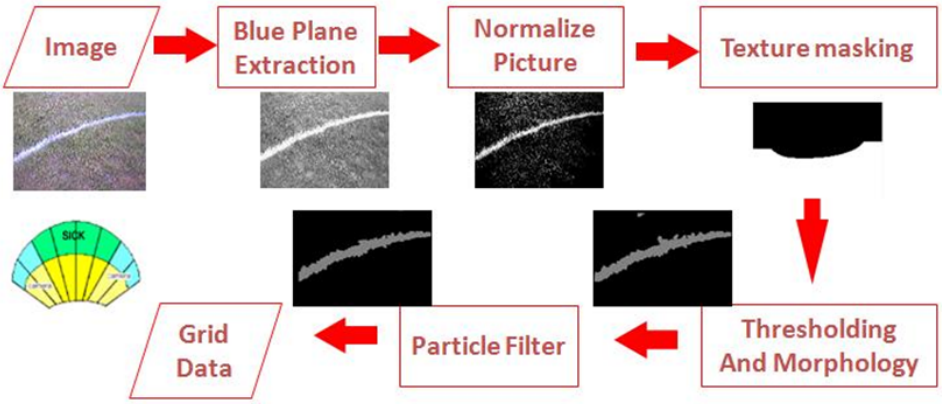


Figure 10: Communication system

The throughput of the vision system has limited the reaction time to the robot. The primary improvement to the system was the addition of the Embedded Vision System (EVS). The EVS replaces the laptop thus reducing overhead from commercial operating systems. Additionally the algorithm has undergone pipelining. The various subtasks in the algorithm now run parallel to each other thereby reducing the execution time and increasing throughput. Efficient management of memory has further reduce the processing time to approximately 70ms per frame.

5.7 Motor Control

The motor control uses shaft encoders and feedback loop to maintain desired motor speed. A simplified kinematic model has been used to convert the speed and heading generated by the VFH algorithm to motor controller commands.

6 JAUS

The Joint Architecture for Unmanned Systems (JAUS) implementation on Q is intended to provide an interface for the robot to communicate with its subcomponents, convey information about its state to the user and for the IGVC challenge to the COP. Although the main JAUS specification document has several messages and protocols that need to be implemented for complete JAUS

compliance, Q must support a core set of messages for the IGVC 2010 Interoperability Challenge. Q does not have many processors and sensors to interface with the Compact RIO and the sensors used are fairly easily accessible. Building JAUS components on the Compact RIO as well as the Vision System was deemed unnecessary as it would have had substantial overhead in communication. Q contains a single JAUS component that handles all of the services required for each of the six tasks in the Interoperability Challenge. Q's internal communication structure is used to pass data between sensors and navigation algorithms. A JAUS thread communicates with each of these internal components whenever specific data is required by an outside source. The wireless router is used as the gateway between Q and other JAUS subsystems.

7 Safety

Safety has been given high priority in the design of Q. Gauge 10 wires were used to connect power sources to the motors and filter, and gauge 16 wires were used for other power wires. A circuit breaker was used to protect the motors from current spikes. A single phase dual stage power line filter was used to prevent transient current generated by the motors from affecting the other electrical components. Slow-blow fuses were installed for each component to ensure that individual electronic devices did not receive too much power.

Three main motor safety measures have been implemented. They include the motor controller FET's, the physical emergency-stop, and the wireless emergency-stop. The motor controller FET's are connected in series with each motor and will control the current to the motors. These FET's are controlled by the cRIO; they will disconnect power from the motors unless the cRIO constantly applies voltage to them. The physical and wireless e-stop features also work by immediately cutting power from the motor controllers as soon as either of them is triggered.

8 Vehicle Cost

Throughout the design process, industry donations were actively pursued to reduce the cost of most components. Since the robot's mechanical base was judge to be more than satisfactory, this base along with many of the other components were reused for the 2010 version of Q. The bulk of the cost for this robot is the NI Compact RIO, NI Embedded Vision System, and SICK laser rangefinder. The cRIO and EVS allow true embedded operation of the robot, making these components worthwhile investments. The SICK is also an essential component for the obstacle avoidance algorithm, easily justifying its cost.

Component	List Price ($)	Cost Incurred($)
ADS Tech Pyro Cameras(2)	180	180
Crescent A100 DGPS	2000	0
Caster Assembly	150	150
Chassis	650	0
Encoders	100	100
Honeywell Compass (HMR-3300)	750	0
Motors	1000	0
NI cRIO-9022 Real-Time Controller	3199	3199
NI cRIO-9133 Reconfigurable Chassis	1999	1999
NI EVS-1464RT Embedded Vision System	4499	4049
SICK LMS291 Laser Range Finder	4000	4000
Power Supply Board	150	0
Encoded Receiver/Transmission Set	77	77
Remote Control	180	0
Roboteq AX3500 Motor Controller	400	300
Wiring	50	50
Total	19384	14104

9 Concluding Remarks

Overall, the numerous changes made to Q have led to a significant performance boost and have improved its usability. In Q we see a great potential to be a robust autonomous vehicle and we look forward to a more successful IGVC 2010.

10 Sponsors

- Enterprise Rent-A-Car
- Hemisphere GPS
- Honeywell International Inc.
- National Instruments
- PerMobil Corporation
- Travelers Insurance
- Trinity College

References

[1] "USA - Products - Support - Product Support - Permobil." *Power Wheelchairs - Permobil.* Web. 14 May 2010. <http://www.permobil.com/USA/Products/Support/Manuals–drivers/>.

[2] Wright, Adam, Orko Momin, Young Ho Shin, Rahul Shakya, Kumud Nepal, and David Ahlgren. "Application of a Distributed Systems Architecture for Increased Speed in Image Processing on an Autonomous Ground Vehicle (Proceedings Paper)." Intelligent Robots and Computer Vision XXVII: Algorithms and Techniques. Proc. of IS&T/SPIE Electronic Imaging, San Jose. Vol. 7539. San Jose: SPIE, 2010. Print.

[3] Ulrich, Iwan, and Johann Borenstein. "VFH+: Reliable Obstacle Avoidance for Fast Mobile Robots." Proceedings: 1998 IEEE International Conference on Robotics and Automation, May 16-20, 1998, Katholieke Universiteit Leuven, Leuven, Belgium. IEEE International Conference on Robotics and Automation, Leuven, Belgium. Piscataway, NJ: Robotics and Automation Society, 1998. 1572-577. Print.