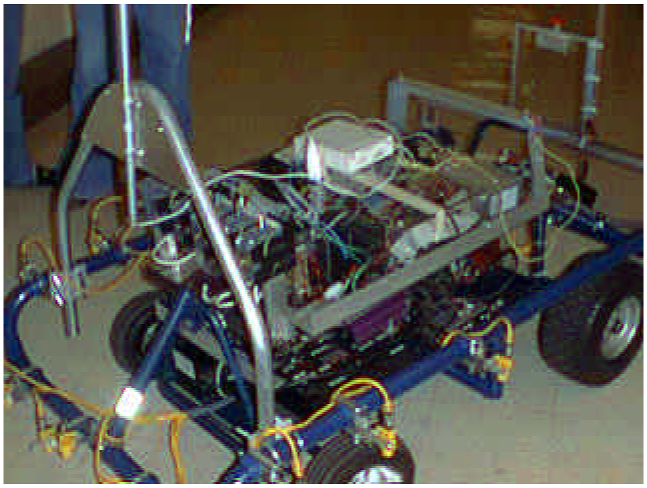
Centurion

Vehicle Design Report

Bluefield State College

Ground Robotic Vehicle Team, July 2002



Faculty Advisor
Dr. Robert Riggins
Professor of Electrical Engineering Technology
Bluefield State College
Bluefield, WV 24701
Phone: (304) 327 – 4134
E-mail:_briggins@bluefield.wvnet.edu

Table of Contents

1 - Introduction

This report describes the design of the Centurion robotic vehicle, submitted by the Autonomous Ground Robotic Vehicle (GRV) team of Bluefield State College. The Centurion GRV is a fantastic robot, and we have total confidence it can win all four competitions. Centurion is a complete rebuild of Bluefield State's intended submission to the Ground Robotic Vehicle Competition last year, named Eniac 2001 (although we applied, we could not bring ENIAC to the 9th annual IGV competition.) The Centurion design team retained the best and most innovative features of the old Eniac 2001, including the banner and sonar systems for redundancy purposes. The team was also successful in addressing several shortcomings of the previous design and added various new features and components. The Centurion GRV team has devoted over 5000 person hours in the development, design, fabrication and testing of this project. It is important for the reader to keep in mind that most of the electronic, mechanical, and computer systems were completely designed, constructed, and integrated by the student team members. The 300-lb Centurion is named after the 1st century Roman commander who led 100 Roman soldiers into battle.

The Centurion entered in this year's competition is unlike anything our school has entered so far. Although, the ideas for design, analysis and construction were derived from ENIAC, we have made significant positive changes in our total approach, including improvements in our design process, organization, and overall GRV quality.

Last year the team decided to take some time off the competition and prepare a better robot. One of the most significant major design changes made since the last time a vehicle from our school was actually entered (Blue Boy in 2000) has been the inclusion of computer vision capability. As we stated in last year's report (the competition we could not attend), analysis was emphasized with Centurion. A major change this year was a shift in focus from a "let's just make it work" attitude to an analytical attitude. Another significant change since Blue Boy is the inclusion of a comprehensive test program in our design process (2 team members dedicated to testing). Also, the structural design of the vehicle has been completely modified. This year the team started all over, using an old go-cart frame. We have also improved our steering mechanism, switching from an unreliable used automobile window motor to a stepper motor that allows precise steering. An important quality that we decided to build into this year's robot is redundancy. Our vision capability is redundant to the banner and sonar sensors. We had noticed in previous competitions *all* robots failed to navigate through the whole course because of failures in identifying road edges and obstacles. Our idea on Centurion is to keep the robot in the competition even if one system fails. We will describe all these concepts and features in this report and in our documentation.

2 - Getting a Start on Design

2.1 Design Process

As in the construction of any complicated system such as Centurion the design methodology is the most important feature leading to success. Our team initially agreed in fall of 2001 that our design process would include (in this order but with constant loop backs) the identification of the customer, formulation of goals and objectives, basic design approach, analysis of systems, construction, and testing. Based on the results of tests, Centurion is a definitely a winner, and we owe this success to the attention we gave to the design process.

On the first day of the fall semester of 2001 the faculty advisor presented to the team the GRV competition requirements. These requirements were the ones used for the 9th annual IGVC but served as a suitable starting point.

Our first step in the design process was to decide who our customer really is and then to set appropriate goals and objectives. We decided we have many customers, some more important than others. Although we have a prioritized hierarchy of customers, we decided our foremost customers are ourselves. We are proud of our design creation but realize that all this would be for naught if it were not for our school, community, and the IGVC competition folks who make this possible. We were motivated and excited about starting the design and construction of the GRV. We know beyond a doubt that we can build a successful GRV and compete because we have talent in all aspects of the design process. Our community and college are also customers in our endeavor and they will definitely be proud of Centurion's success. Our other customers include the IGVC judges and our community sponsors. Our primary goal is simple: we want to win the competition.

Like every engineering problem, the GRV challenge presented to us needed to be solved effectively and cost efficiently. Therefore, next in the design process was to decide how we should start (basic design approach). Early in the design stage the team decided to take the Eniac 2001 as a base rather than starting a fresh design. Although the Eniac 2001 was an excellent base design, major design changes were needed to enhance the vehicle's performance. The design and software upgrades have produced an excellent solution in Centurion.

Throughout the design process we considered the following criteria: adherence to competition rules, safety, modularity, and reliability. In order to meet these criteria, we planned an extensive analysis program. Analysis would include calculations and verification of sensor performance (banners, sonars, camera, and GPS) as well as of battery life, climbing power, and reaction time.

With the above-mentioned criteria in mind, construction would begin. We decided at our initial meeting to reduce the number and size of components on board the vehicle, and to introduce the PC Anywhere capability with a wireless network connection. This made the testing of the robot considerably easier since we no longer needed a monitor, keyboard, or mouse. Also, it made the overall organization simpler and made components more accessible, which eases maintenance on the vehicle such as changing batteries. Additionally, we decided to spend considerable design time reducing the overall size and weight of the vehicle, as compared to the previous ENIAC 2001 and Blue Boy.

The final stage in our design process plan is to do extensive testing *before* going to the competition. Throughout the year we stuck to our initial design process plan very close, and as a result we have an outstanding robot.

2.2 Design Team Organization

The next step in the design process was to build a GRV team. All of the design work on the Centurion was done as a team of undergraduates from various disciplines at Bluefield State College, including electrical, mechanical, and computer engineering. Everyone on the team participated in the initial design process, so everyone would have knowledge of the “big picture”. Our faculty advisor and student team leader chose students with diverse talents and abilities to participate on the Centurion project. We assigned to each member tasks that matched their abilities and desires, which resulted in a team that worked very well together. The team members for Centurion and their areas of contribution are as follows:

Sauabh Srivastava	Spring leader and programmer
Joshua Fowler	Fall leader and mechanics and testing
Amy and Jarrod Snider	Structure and electrical circuitry
Tom Lambert	Electronics, networking, programmer
Shawn Nunn	Programmer and knowledge on batteries/chargers
Chandra Duty	Electronics and sponsor hunter (our PR person)
Ammar Shawli	Testing
Anthony Crews	Structure and mechanics
Jerry Pugalee	Electronics

2.3 Use of Computer Aided Design (CAD) in the Design Process

Throughout the process we used CAD to design both the mechanical systems and the electrical systems. These drawings are much too numerous for this report but will be available.

3 – Mechanical System

3.1 Basic Vehicle Frame Design

The basic structure of the vehicle comprises of three decks. The lower deck is a steel plate serving as a strong base to the rest of the robot. Welded to this lower deck are the side supports and the extended frame of the vehicle on which the diffused visible sensors are mounted. This deck serves as a mounting place for the batteries, drive motor, steering motor and charging circuit. Such a design gives us a low center of gravity and gives us an easy way to distribute weight evenly across all 4 wheels.

The middle deck is made of lexan, reducing the weight of the vehicle. It does not extend all the way across to the other side support, but it is hinged to one of the side support such that it can be lifted up to access the batteries and other components in the lower deck. This deck serves as a mounting place for the inverters and a GPS receiver.

The upper deck, like the lower deck, is lexan and extends across the full length of the robot like the lower deck. However, it is also hinged to one of the side supports so that it can be tilted up to reach the components below. This allows easy access to all the heavy components on the bottom deck such as the batteries. The second deck houses the on-board computer motherboard (733 MHz, Athlon 10GB hard drive, 256 MB of RAM), which controls all the tasks of the vehicle through our programs. In addition, this layer holds the home-built speed controller, homebuilt sonar interface, the GPS receiver, the homebuilt banner sensor electronics, the homebuilt remote stopper, and the computer-to-robot I/O card.

Attached to the front end of the vehicle is our "arc of St. Louis" pipe frame, acting as a support for the camera mount. The camera is mounted on the top of a rod, which mounts of the top of this arc-like structure. The rod slides up and down to achieve the optimal camera height for a particular setting and better testing capability.

Angling out from the rear side supports are two bars, which provide extra support and strength to the vehicle design. Also, attached to the rear extended frame is a tower-like structure that provides additional room for mounting all antennas used in this project: the GPS antenna, the wireless network antenna, and the wireless remote-stop antenna. Also mounted here is the homebuilt battery monitoring system, which is helpful in monitoring the charge on the batteries.

Lastly, to provide a strong support for the solar panel, two pillar like steel bars where mounted on the upper deck. The solar panels are attached to these supports with the help of nuts and bolts. The purpose of the solar panel is to provide charging to the batteries via our homebuilt on-board charger (we know the sun gets bright and hot in Orlando and we may be outside a lot).

All banner sensors are mounted on the extended frame of the robot so that none of the other components of the robot interfere with them.

3.2 Drive Motor

Centurion is a one-wheel drive robot so that wheel slippage is minimum while turning. From our analysis (see the analysis section), a one-wheel drive system provides plenty of power but it is simple to build. Also, our extensive testing has proven the analysis to be correct.

Also, from our analysis (again, refer to the analysis section for details) we chose to purchase a 1 horsepower 24-volt DC permanent magnet Scott motor as our main propulsion drive motor. Also our gear analysis led us to use two sprockets in order to reduce the 3000-rpm motor to a 5 mph maximum robot ground speed (we tested the actual top speed of Centurion and found it to be 4.9 mph).

3.3 Steering Motor

The steering motor is a linear stepper motor with 300-lbs of force designed to rotate a rod in or out a specified number of revolutions. This linear motion is translated to turning by the use of a worm gear mechanism that we built ourselves and mounted it to the go-cart frame. If a sensor senses an object or line, the program will command the steering stepper to turn the robot away from the object or edge. Essentially, the stepper allows precise turning by the program (in other words, precisely to a specified angle).

3.4 Auto CAD drawings.

We have Auto CAD drawings for many systems on centurion, but decided not to include them here in the report because of the length restriction. One of our drawings is a 3D view that we can rotate within Auto CAD and see the location of the major components. We will have this, as well as the other drawings, in our documentation package at the competition.

4 – Electrical Devices and Sensors

4.1 Overview

In this section we briefly describe the electrical devices and sensors. For details we refer the reader to our extensive documentation package.

4.2 Power System

Four Genesis Hawker Energy batteries power everything except the sonars. Batteries #1 and #2 power the 24-volt main propulsion motor, battery #3 powers the turn motor and GPS, and the 4th battery powers the computer and sensor electronics. We built an on-board charger so that we can charge the batteries without opening the robot (although if needed it is not that difficult to open it). We connected one 300-watt inverter to battery #3 for the turn motor and another 350-watt inverter to battery #4 for the computer power supply. We also built a battery monitoring system using integrated circuit comparators. With a glance at this box we know exactly how much battery charge remains. The sonars have their own 6-volt polapulse batteries. These polapulse batteries are a little larger than credit cards and are designed to handle a large pulsating current that the sonars need.

4.3 Control System

The control system uses a Curtis pulse-width-modulation MOSFET controller connected to 100-amp contactors to control speed of the main propulsion motor. We connected a main contactor between the 24 volts on the series-connected motor batteries and a 300-amp fuse. A forward/reverse contactor pair allows us to switch polarity to the motor, thereby changing directions between forward and reverse. To control the contactors we have a homebuilt darlington pair interface between the contactors and the computer relays. We control the computer relays with our program. We are currently using 8 relays: one for main power, one for forward motion, one for reverse motion, and five for speed control. We have extensive drawings on all these systems, which are all available in our documentation bundle.

4.4 Video Camera

Our video camera is an inexpensive web-cam (a more sophisticated camera such as a camcorder is not needed!) connected directly to the computer usb connector. We found a capture control for visual basic that allows us to capture pictures from video and import the picture directly into the visual basic form. There we can easily examine and set pixels. This vision approach that we are using is simple, inexpensive, and straightforward, but yet yields superior results in all our testing, both indoors and outdoors. We are currently using the web-cam for lane following as well as obstacle detection and location.

4.5 Banner Sensors

Our banner sensors are actually industrial grade diffused visible sensors that detect changes in a pre-programmed color. We have them distributed around the bumper of the robot to detect road edges and potholes. This edge-detection function is redundant to the camera lane following function. We have purposely designed in this redundancy so that in the event the camera lane following function fails, the banners will pick up the edge. Of course, all banners are connected to the computer interface so that our program can constantly monitor them. Another reason to use the banners other than redundancy is the ability to detect sand. We programmed two of the banners to detect sand and built a “sand tester”, which has given us great results so far (we’re going to get through your sand this year!)

4.6 Sonar Sensors

We have 6 sonar sensors mounted to the front of Centurion. The purpose of the sonars is to detect and locate objects in front of the robot. Again, this object detection and location function is redundant to the camera object detection and location function. Again, we incorporated this redundancy initially in our design. If the camera function fails to locate an object, then the sonars will see the object. (For more details, see the software design below.)

4.7 GPS

We are currently using 3 different GPS receivers in the navigation design stage, but we will choose one for the competition. The one we will probably use is our WAAS-enabled Garmin etrex. WAAS is the Wide Area Augmentation System that gives these specially equipped receivers a one-sigma error bound of 3 meters (almost as good as differential!) We will not be equipped for differential GPS, so this, together with a "relative" GPS algorithm we wrote, is an adequate substitute. In our relative GPS algorithm we subtract our receiver's position at the start from the competition's position for the start and use this "error" throughout the navigation challenge run (see our analysis below on why we can do this.) This receiver also has an electronic compass so we are able to get azimuth without having to move the robot. The receiver connects directly to the computer comport, and latitude/longitude/velocity/azimuth data is imported into the visual basic form for computations. Our GPS program has performed adequately in tests but more testing and tweaking the program is needed at this time.

4.8 Wireless Network and PC Anywhere

One of design goals in fall 2001 given to us by our faculty advisor was to incorporate a wireless connection between the robot and an off-board laptop (our advisor was at last year's competition and liked what VT had done with their "Maximus" robot and a wireless connection.) We satisfied this goal by purchasing a wireless network access point for the robot and a wireless network modem pc-card for the laptop. We also downloaded software from the web called "PC Anywhere" that allows us to control the robot computer with the laptop. This feature has been wonderful, allowing us to test and program the robot without actually touching the robot! Also, no monitor, keyboard, or mouse is needed on Centurion.

4.9 E-Stop

As the requirements stated, we have both a manual E-Stop on the robot and a remote E-Stop. One of our teammates designed and built the remote E-Stop using code modulation so as not to have interference from other robots or emanations from other sources such as computers and motors.

5 – Software Design

5-1 Overall Approach

In this section we will very briefly discuss the overall software design approach. The entire program, including the testing software, is over 100 pages long, so we will only highlight here and refer the reader to our documentation package for details.

At the beginning of the fall semester we decided on what our overall design structure would be so that programmers could develop programs that would easily fit into this overall design (without such an organized approach, a chaotic programmer's nightmare could happen because of the many programmers contributing to Centurion!). To do this we built a visual basic main form with certain command button controls that point to the various major tasks. These tasks are the following: autonomous run, sonar tests, banner tests, camera tests, remote control, GPS test, GPS run, I/O port tests, and motor control tests. Each programmer has to go to one of these functions to enter his/her program into the robot. For example, under the camera tests, we have many programs written by many students over the past year. Team members can write their software at home and later install it on the robot within this special structure. With several clicks we can easily access anybody's program and go with it.

5-2 Lane Following and Obstacle Avoidance Software

We have a built-in redundancy for both lane following and obstacle location, as stated in the hardware sections. The autonomous run program has the following structure:

```
Do to loop below every 0.5 seconds:
Check Banners, if Banners indicate no edge then skip this section
    If left banners indicate road edge then do:
        Turn right, go straight
    If right banners indicate road edge then do:
        Turn left, go straight
    If front banners indicate road edge or a pothole then do:
        Stop, turn left, backup, straighten, go forward, turn left
```

```
        If sand banners indicate the presence of sand then do:
            Ignore other sensors except camera, proceed straight through sand
    Check sonars, if sonars indicate an object then do:
        If the object is on the front left within 8 feet then do:
            Turn right,  go straight, turn left, go straight
        If the object is on the  front right within 8 feet then do:
            Turn left, go straight, turn right, go straight
        If the object is right in front within 8 feet then do:
            Stop, turn, backup, stop, go forward, turn, go straight
    Use camera to navigate around barrels, potholes, and the road edges.
    Loop back to beginning until a stop command is received
```

5-3 GPS algorithm:

For our GPS algorithm we use the following general approach:

```
Input all coordinates
Find error between given and measured start coordinates
Manually list the targets in order at the start
Monitor obstacles with sonars
Move to target 1 by matching target coordinates with measured
If the sonars see an object, abandon GPS and move around as above
    then return to GPS
If reach a threshold, repeat to target 2
Do the same for all targets including home base.
```

If you would like to know details, please refer to our section on GPS and to our documentation.

6 -- Analysis

6-1 Overview

Over the past two years our team and the previous year's team have done analyses on all systems of Centurion. In last year's report we described in detail our analysis of the banner sensors, the sonar sensors, the motor analysis (ramp climbing ability), and speed. We will defer to that report for those analyses and include here only the analyses not in last year's report (we'll have a copy of that report, along with the judge's comments, at the competition.)

6-2 Reaction Times

We have identified hundred's of possible situations to calculate and test reaction times between the computer and actual motor action, between seeing an obstacle and the robot's response, between taking a picture and the robot's response, between seeing a road edge or pothole on the left, right, front, or back and the robot's reaction. The main propulsion motor has its reaction time and so does the steering motor. For example, the reaction time for the robot to capture a still to reacting to what it "sees" depends on the speed of the robot, the steering motor delay, the processing speed, which algorithm we use (we have many) and other things. By far our biggest contributor to the overall reaction time is the steering motor, which can take up to four seconds to turn a full 45 degrees (even our camera processing only takes 2 seconds per frame.) However, from our tests, the steering motor starts turning in less than a second after the camera program sees the object and finishes the processing.

6-3 Battery Life

With everything turned on, the computer battery (#4) lasts for 3 hours, the turn motor battery (#3) lasts for 4.5 hours, and the main motor batteries (#1 and #2) lasts for 5 hours. The polupulse batteries are good for 30 minutes in continuous operation. The laptop last for about 2 hours when on battery.

6-4 Distance at Which Objects are Detected

As stated in last year's report the maximum distance the sonars can detect is 35 feet and the minimum distance is 6 inches. The camera's reach depends on the camera angle. In testing we have been able to see objects as far as 50 feet away (although we will probably not use such a shallow camera angle.)

6-5 *GPS Accuracy*

We realize that differential GPS is allowed this year, so we analyzed the accuracy of non-differential GPS, differential GPS, and WAAS-enabled GPS. For non-differential GPS we used a CMC receiver, for differential GPS we used a Trimble receiver, and for the WAAS-enabled receiver we used a Garmin etrex receiver. To compare these we connected each receiver to the PC and ran our own visual basic software that we wrote to graphically display the points on a grid over a period of time. The antenna was not moving, so all apparent movement related directly to dynamic error. To get our static error we made sure our antenna was on a surveyed marker that our civil engineering department set up for us. This is approximate but it does give us a good relative idea between these 3 approaches. Our results are as follows:

Non-differential	10-20 meters
Differential	2-3 meters
WAAS	5 –10 meters

As the reader can see, our WAAS result was a little disappointing but we feel this may improve by the time of the competition (WAAS is supposed to be down to 3 meters.) Because the Trimble unit we have is very large and bulky, we decided to go with the WAAS receiver.

7 – Other Design Considerations

7-1 Safety, Reliability, and Durability

Throughout this project our advisor put safety first. For example, one reason for changing from metal supports to lexan is that lexan is non-conductive. Also, Centurion always has one of us walking beside the e-stop and one of us manning the remote stop during testing. The on-board charger connection is far from the batteries to prevent a spark-caused explosion. We placed the contactors on the opposite side of the robot from the batteries because contactors inherently spark when energized. These precautions are just some of our safety features.

Reliability has always been a concern for us because most of our components are designed and built by us and not bought “off the shelf”. Therefore we had to ensure quality control by helping each other. Some of the team members had expertise in areas that would enhance reliability. For example, one of our team members was an expert solderer. By helping with the soldering she was able to ensure reliable circuits. Also our built-in redundancy will help in overall performance reliability.

Durability was actually enhanced when we changed over from metal supports to lexan. Lexan is just flexible enough that bouncing the components is reduced (in other words it acts as a shock absorber.) We had a lot of talent in the area of mechanics, so the entire robot is very durable.

7-2 Cost

By far the most expensive item on Centurion is the stepper motor controller ($800). Next in line are the banner sensors ($100 each), followed by the propulsion motor ($250) and Curtis controller ($250). We have thousands of smaller parts, but if we had to duplicate this robot without enjoying the donations we've had, then the total robot would cost about $5000. This number assumes that we would build all of our circuitry ourselves as we have done and it does not count the laptop that we remotely connect the robot to (the laptop is not unique and is used for a multitude of other purposes other than for Centurion.)

8 – Conclusion

We are confident Centurion will perform well in all four competitions this year. We have discussed in this report several innovative approaches that give us a definite advantage. These innovations include the addition of banners to provide pothole, sand, as well as road edge detection. Also we have included redundancy as a part of the overall design process. A solar panel provides a means to charge the batteries while out in the bright Orlando sun. And finally, we are believers in Visual Basic. Centurion is controlled and tested by our user-friendly visual basic program, that was easy and a delight to write. With these innovations we have a fantastic product this year.

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