AC 2007-2528: MICROPROCESSOR BASED, GLOBAL POSITIONING SYSTEM GUIDED ROBOT IN A PROJECT LABORATORY

Michael Parten, Texas Tech University

Michael Giesselmann, Texas Tech University

I. Introduction

Electrical and Computer Engineering (ECE) students have a need to be able to design and build systems with embedded microprocessors. They also need to be able to become familiar with different processors. There are many different ways to teach microprocessors and their applications. The objective, in this case, is to have the students design and develop a system using a microprocessor they have not seen before. In this way, students learn that their basic knowledge can be carried over to other devices and systems.

This paper describes a second semester sophomore laboratory project to design and build an autonomous robot vehicle capable of navigating an outside area the size of a small parking lot by guidance from a GPS sensor. The robotic vehicles normally use the frame of remote controlled cars. The students are divided into teams of 3 to 4 members. The teams compete at the end of the semester. The winner is the vehicle and completes the parking lot drive in the shortest time.

II. Project Laboratories

The laboratory structure in the ECE department at Texas Tech University is somewhat different than most university laboratories.[1-8] There are five, three hour credit, required laboratory classes. Although all of the laboratories have pre-requisites, they are not associated with any one class. All of the laboratories require students to work in teams on long term projects. The student teams each have a project advisor, separate from the lab instructor and teaching assistant associated with each lab. All of the teams report on their progress and answer questions on their projects in a weekly three hour lab meeting with all of the groups.

The first project laboratory, EE 3331, normally occurs in the second semester of the sophomore year. The prerequisites include the first English, chemistry and physics courses. ECE prerequisites include single courses in digital logic, circuits and microprocessors.

All of the project labs have the same basic objectives. At the completion of this course students should be able to:

1. Identify, analyze and solve electrical or computer engineering problems by applying knowledge of mathematics, science and engineering with modern engineering tools.

2. Design a system, component or process to meet desired needs within realistic constraints

3. Communicate effectively through oral presentations and group discussions.

4. Communicate effectively through written reports and other documents.

5. Design and conduct scientific and engineering experiments, and to analyze and interpret the resulting data.

6. Function and communicate effectively, both individually and within multidisciplinary teams.

7. Interact with other students, faculty and practicing professionals on professional and ethical responsibility issues.

8. Recognize the need for, and ability to engage in, perpetual learning by working on projects, both individually and within multidisciplinary teams, for which they have no prior experience and developing ways to learn.

9. Use basic statistical techniques to analyze data.

Each individual lab stresses different areas. The first lab includes basic electrical measurements procedures, equipment and the design of basic electrical circuits and digital systems. The first lab has two projects, one short (about 4 weeks) and one long (about 10 weeks). Teams, in the first lab, normally consist of three to four students. The laboratory is open and available to the students during normal office hours and for an additional 30 hours, total, at nights and on weekends. The rest of the labs have one project per semester or, for the senior labs, possibly one project over two semesters.

The complete ECE Undergraduate Laboratory Policies are given and discussed with the students every semester, including one of the important objectives described in the beginning of the policy statement. "One of the objectives of the ECE laboratory program at Texas Tech is to expose students to areas they have not seen before. It is important for students to develop confidence in their basic knowledge and to realize that they can extend that knowledge to new and exciting areas. In addition, it is important for students to begin the transition to life long learning and to not be afraid of something they haven't seen in a class. Engineers are seldom asked to solve problems that have already been solved. In industry, engineers are constantly asked to learn and develop new techniques and systems for which they may have little prior experience."[9]

III. Project Description

The specific project described below was the second project in the Fall 2006 semester.

Design and build an autonomous robot capable of navigating an outside area of the approximate size of the TTU R4 parking lot by guidance from a GPS sensor. The specific design goals are:

- *Determine the accuracy of the GPS sensor and implement methods by which this might be improved*

- *Determine the appropriateness of the sensor for the R4 parking lot navigation task and any challenges that the specific mission profile might entail*
- *Establish the stability of the navigation system over 10 minute and 24 hour time frames*
- *Smooth and straight operation of the vehicle on an EAST-WEST and NORTH-SOUTH heading.*
- *Start from one designated position in the parking lot and go to another designated position. The specific positions will not be known until the day of the competition. The final position will in a barricaded area with an opening smaller than the GPS level of accuracy. Moving to the final location will require additional sensors to determine the opening in the barricade and the path to the final position.*

Additional Criteria:

The robotic vehicles shall be evaluated on the performance of the assigned behaviors, quality of electrical design, robustness, craftsmanship and aesthetics.

The project teams were required to use a specific GPS sensor, Garmin 35PC, a specific microprocessor, an MSP430, and an H-bridge driver. This microprocessor is low power and is a different processor than the students used in their microprocessor course (68HC12). The vehicle is a remote controlled car that is stripped down and used for the chassis, motor and servo.

The students are divided into teams. The primary variables in the project are how to carry out the navigation and how to sense obstacles and enter the final area. All of the teams present to the whole class each week on their progress, including technical details. The students learn from other teams presentations. Even with this, many differences still occur in the projects.

IV. Student Work

The following are excerpts from a couple of student reports to provide a flavor of the work accomplished. A block diagram from one of the teams is shown in Figure 1. The GPS sensor communicates with the MSP430 over a serial port.

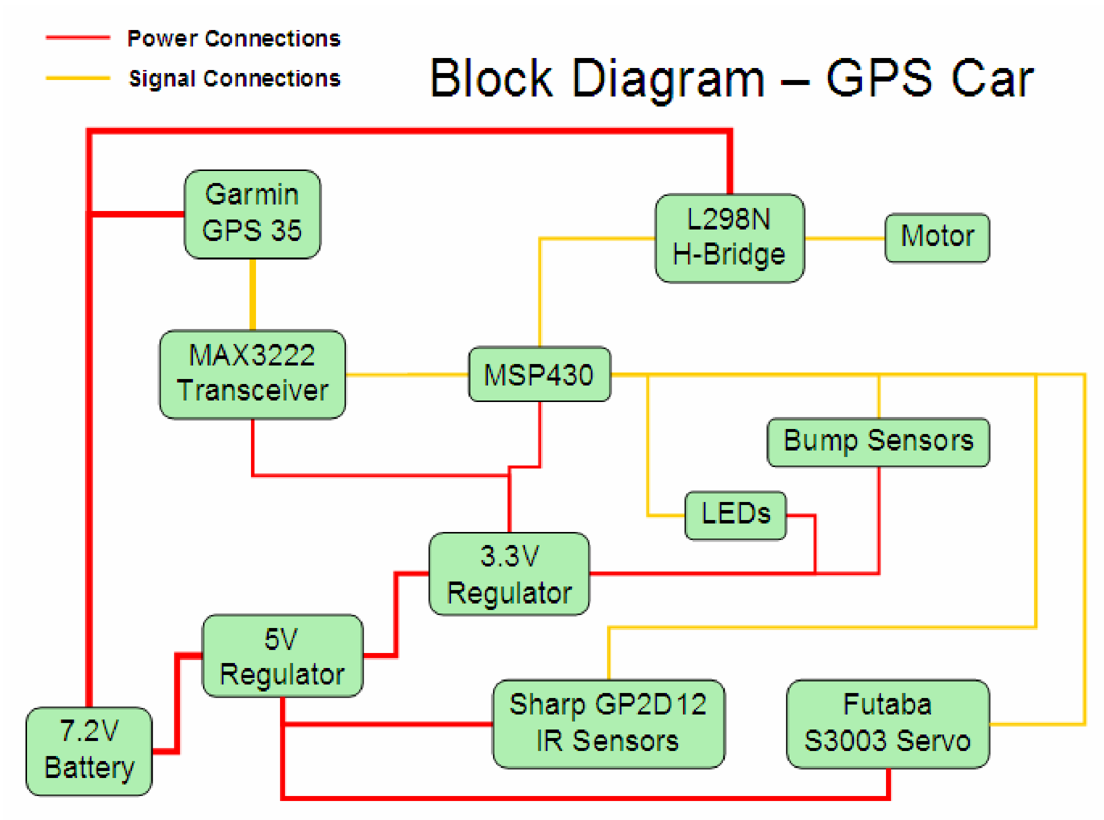
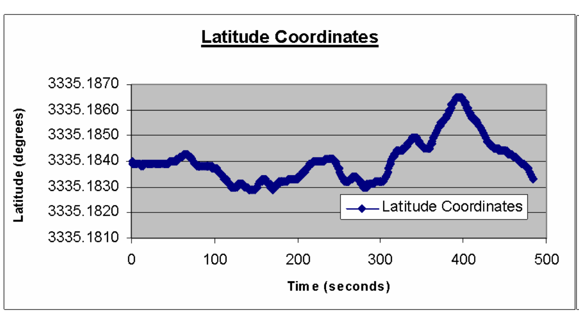


Figure 1. GPS Car Block Diagram[11]

Aside from the serial port interface, drive for at least one motor and one servo were required for the system. The box at the end of the track required other sensors to find and enter the area. The system also required three different voltage levels.

Part of the project was to determine the GPS sensor capabilities. One team's results, as given in one team member's final report, are given below.

"Data readings were taken from a stationary GPS sensor for a period of around 8.5 minutes and were graphed to show the precision and variation of the coordinates. The variation of latitude is shown in Figure 2, and the variation in longitude is shown in Figure 3 below. The variation in latitude in this data was calculated to be 0.0037 minutes, which is ~6.85 m. The variation in longitude is 0.0055 minutes, which is ~8.49 m.



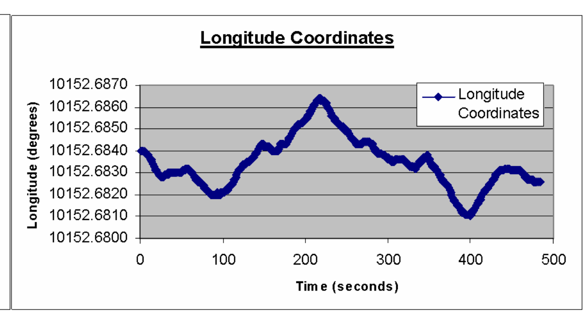


Figure 2: Latitude Data. Figure 3: Longitude Data.

To confirm a theory, more readings were taken at the same time of day and for the same length of time, but instead of a stationary GPS sensor, the sensor was constantly moved in a circular motion of a radius of about 30 cm. The readings obtained from this test are quite fascinating as they show that the overall variation of latitude and longitude has decreased dramatically. In order to see this decrease, the variation was calculated in distance, just as in the previous case. The latitude varied by 0.0026 minutes or ~4.81 m (29.8 % reduction) and the longitude varied by 0.0018 minutes or ~2.77 m (67.4 % reduction). This provides compelling evidence that the GPS sensor is much more precise while in motion, no matter how small the motion might be.

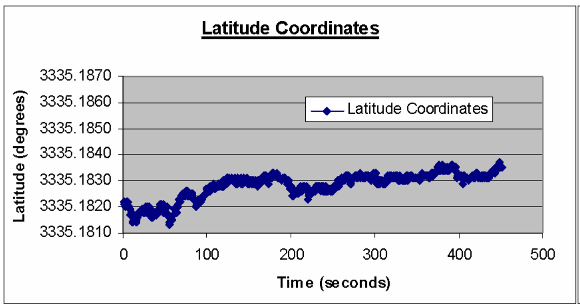


Figure 4: Latitude Data while moving.

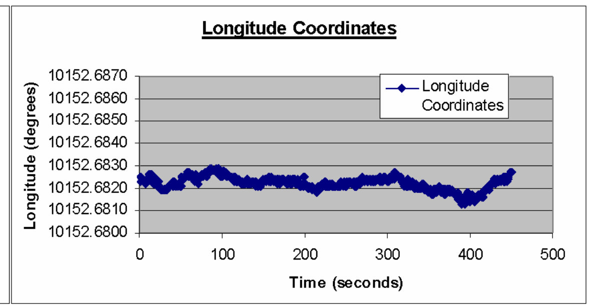


Figure 5: Longitude Data while moving.

More tests were done to see how close to the desired point the autonomous vehicle would end up. It was observed that the vehicle almost always stopped past the desired point, and 60 % of the time, it stopped within 1.0 m of the desired point. This can be seen in Figure 6 below."[10]

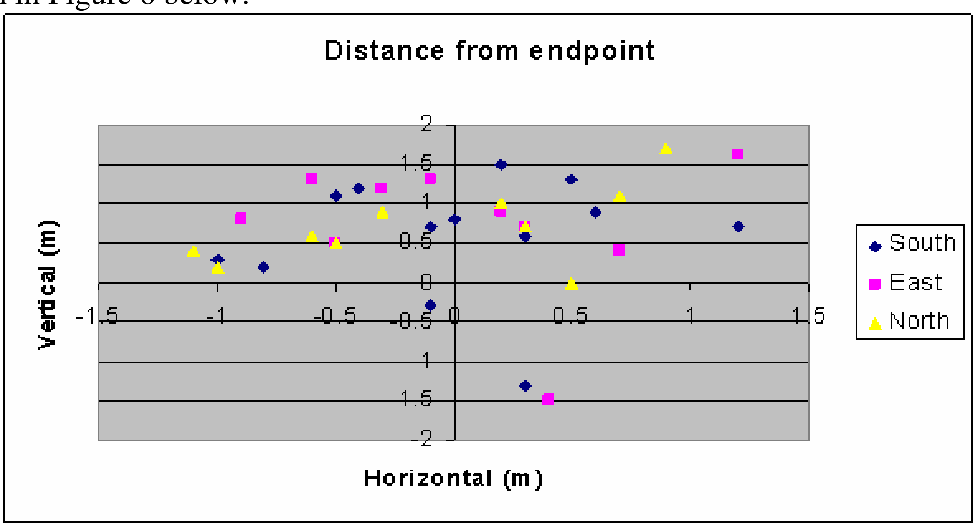


Figure 6: Endpoint Frequency."

A team member from another group describes the navigation algorithm the team used.

"GPS accuracy issues caused problems that needed to be accounted for by software navigational control. As a result, the car moved around the chosen path by traveling in three strait line legs, each following a longitude (for north/south travel) or latitude (for east/west travel) line. The car adjusted itself to this straight line by way of adjusting the servo PWM signal through a proportional/differential navigation method that relied on comparing current GPS position data to previously or manually stored GPS position data. The proportional/differential formula used was:

$$\theta = kd(\text{current point} - \text{wayline}) + kp(\text{current point} - \text{previous point}) \qquad \text{Equation 1}$$

where *kd* is the differential constant
kp is the proportional constant

This formula provides a number, θ, by which we can adjust the PWM sent to the servo. Each point is a measure of longitude (for east/west travel) or latitude (for north/south travel). The value θ was in relation to a travel way-line that was created from the leg starting position of the car, which was a single point taken at the beginning of each leg. Our testing phase showed that unlike the corner waypoints that were averaged over time, the start point for each leg needed to be simple because the averaged point could still be at least 3 m in *any* direction from the car itself. This caused the car to go into an infinite circle as it could never correct itself to be parallel to the way-line, being too far away from the line originally.

The value of *kd* and *kp* was experimentally determined to be of a 1:5 ratio respectively. This meant that the car needed to compensate more for adjustments in forward travel compared to left/right adjustments. This allowed the car to oscillate wide at first if the start point was slightly erroneous, but as it traveled forward, the oscillations became gradually less until the car is traveling nearly on the way-line (within ±1 m). Our testing showed that in most cases, the car would travel strait with little oscillation provided that the initial start point data was given time to settle. This method was accomplished by utilizing a "go" button that allowed the user of the car to start the course navigation at will. For each leg, the car was stopped momentarily to obtain a better position before calculating its straight line path.

The way-line was offset from the car's initial position by half of an arc degree (approximately three meters). This offset was an attempt to compensate for the ever changing GPS position data such that the car would have a line to be proportional to at all times, allowing for continual adjustment to its straight line navigation. Figure 7 shows this concept.

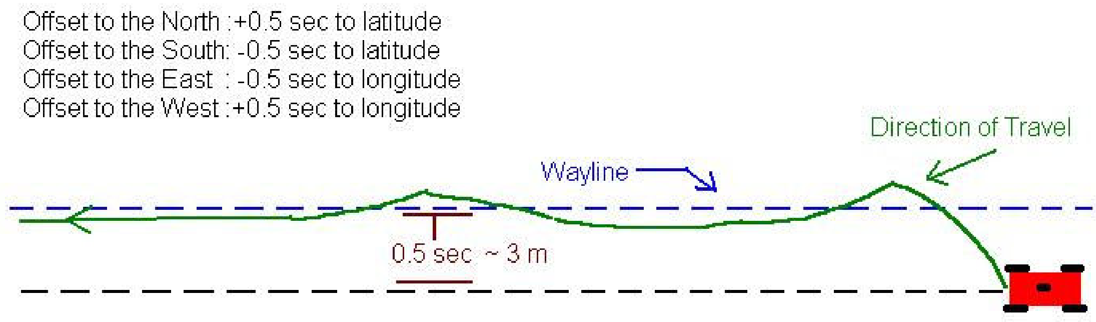


Figure 7: Vehicle Movement by PD Navigation

Therefore in theory, the car would proceed to a point in parallel with this way-line and travel straight. However, this situation only occurred if the GPS data were 100% accurate all the time. As it was not, outliers caused the car to oscillate slightly across the way-line. This was compensated for by keeping the motor speed at 50% of its full capacity, therefore allowing more time for correct adjustments to be made, and less travel error to occur.

The car would travel along the way-line until it crossed a predefined "waypoint" threshold. Three waypoints were taken by use of a loop at the beginning of the code by way of a manual button, which was pressed at the point where a waypoint was desired, and stored into the microcontroller RAM. Each waypoint was made up of an average of ten latitude and longitude points. Therefore, if the car is traveling north, then the car would compare its current latitude position with the waypoint latitude, and if it crossed this threshold, the car would turn into the next leg until it ran out of the pre-assigned waypoints. These turns were hard coded left turns due to the course that was ultimately ran. The servo would pull hard left for approximately three seconds before straightening out the servo, resulting in a well defined 90˚ turn."[12]

V. Specific Results

There were 9 groups in EE 3331 in Fall 2006 with 3 team members per group. All vehicles were mobile by the end of the project. Most could navigate to some degree. Two groups were able to completely navigate the course multiple times, but only one was able to enter the box. The winning car is shown in Figure 8.

Figure 8: Successful Entry of Vehicle [12]

VI. Conclusions

As is always the case, some projects faired better than others. However, in all the students enjoyed the projects and felt they got a lot out of using a different processor. They also learned they could tackle complex, long term projects that involved new and exciting areas. The competition also adds a lot of excitement to the project. Many of the teams worked long hours to complete their projects, principally because of the competition.

References

1. M.E. Parten, “Microprocessor Based Quasi-Autonomous Robotic Projects,” Proceedings of ASEE 2006 Annual Conference, Chicago, Illinois, June, 2006
2. M. E. Parten, “Embedded Microprocessors in a Project Laboratory," Proceedings of ASEE 2004 Annual Conference, Salt Lake City, Utah, June, 2004
3. M. E. Parten, “Using Virtual Instruments in a Measurements Laboratory,” Proceedings of ASEE 2003 Annual Conference, Nashville, Tennessee, June 2003
4. M. E. Parten, "Project Management in the Laboratory," Proceedings of ASEE 1995 Annual Conference, Anaheim, Calif., June 1995
5. M. E. Parten, "A Different Approach to Engineering Laboratory Instruction," Proceedings Frontiers

in Education, November 1994, San Jose, Calif.
6. M. E. Parten, "Progressive Design for Instrumentation Development in Project Laboratories," 1993 ASEE Gulf-Southwest Annual Meeting, Austin, TX, April 1-2, 1993.
7. M. E. Parten, "Design and Research in Project Laboratories," Proceedings of Engineering Education: Curriculum Innovation and Integration, Engineering Foundation Conference, Santa Barbara, CA, January 1992.
8. M. E. Parten, "Design in the Electrical Engineering Laboratory," 1988 ASEE Gulf-Southwest Annual Meeting, San Antonio, TX, March 17-18, 1988
9. Project Laboratory Objectives, http://www.ee.ttu.edu/lab/labpolicy/policy.html
10. Charles Kleuser, "GPS Car Final Report", EE 3331, Texas Tech University, Team Members: Brandon Brice, Tasha Franklin, December, 2006
11. Davis Chanjaplammootil, "GPS-Guided Autonomous Robot," EE 3331, Texas Tech University, Team Members : Dustin Graham, Steven Loveless, December, 2006
12. Gabriel G. Ramirez, "Autonomous GPS Guided Robot," EE 3331, Texas Tech University, Team Members: Arthur Graves and Kevin Stebbins, December, 2006