Synapse
Unmanned Autonomous Vehicle

Dariel Marlow darielmarlow@hotmail.com
Michael DeLisi delisi@eng.utah.edu
Toren Monson nicmonson@gmail.com
Matt Stoker matt.stoker@gmail.com

http://www.cs.utah.edu/~delisi/cs3992/

Abstract

The Synapse UAV is a GPS guided remote controlled car that houses a Pocket PC with GPS and microcontroller. The car will follow a route, programmed by a base station laptop, and guided by a GPS. The GPS unit, connected to the Pocket PC, will be constantly receiving data from satellites and it will use this information to calculate the path to vehicle's final destination. The Pocket PC will send commands to the microcontroller via serial, which will change the car's direction to reflect the GPS data. The microcontroller will interface directly with the car's RF controller to send the updated directions. The base station will assign the route to the Pocket PC and await detailed route information received upon vehicle's return.

Introduction

The purpose of this project is to create an autonomous, self controlled vehicle through the utilization of several components. The vehicle chosen for our final project is a remote controlled car capable of carrying all of the components, which will weigh roughly two pounds. It will need to lift the following devices: a Pocket PC with GPS unit, a micro controller to interface the Pocket PC with the vehicle's guidance system, the remote control of the vehicle used, power sources for all onboard devices, and possibly other items such as a small RF camera to send back streaming video of travel and sonar range finder to avoid collisions by detecting objects in its way.

The first component, the HP H4155 Pocket PC [1], will contain the main application that will control the remote controlled vehicle. The Pocket PC was chosen because of its ability to be combined with a GPS unit and the compact programming platform that allows for high level applications. The Pocket PC, with the aid of the GPS unit, will use data received from satellites to calculate the vehicle's path to its final destination. The pre-defined route will be assigned by the base station and the Pocket PC must calculate differences between the defined route and the instantaneous GPS data to send vehicle adjustments. The main application will be written in the C# programming language [2] since the Pocket PC

has a native MS .Net platform. It will use GPS libraries from GeoFrameworks [3] since it provides an easy, but powerful, interface to the GPS unit.

The next component, the Motorola MC9S12C32 microcontroller [4], is used as an intermediate interface component from the Pocket PC to the vehicle's motors. A microcontroller is necessary to interface the serial navigational commands from the Pocket PC to the vehicle's guidance system. This specific microcontroller, the Motorola MC9S12C32, was chosen because of its advanced abilities, small size, and familiarity. The microcontroller will connect to the Pocket PC via serial interface and will have several outputs to the remote control unit. This RF modularity will allow for other RF based vehicles to be controlled with minor software changes to the main Pocket PC application. In order to send and receive information to and from the Pocket PC, we will use a standard serial communication protocol. The serial communication will have 8 data bits, 1 parity bit, and 2 bits for the start and stop signals. The microcontroller also offers the ability to add extended devices such as sonar range finder for collision avoidance. We can connect the sonar range finder to the microcontroller and pass vital information from the microcontroller to the Pocket PC about what is ahead of the vehicle; the Pocket PC can then make corrections to the vehicle's path to avoid collision.

The next baseline component is the base station. This component will have a path calculating interface that will allow the vehicle commander to select GPS coordinates that the autonomous vehicle must follow. The passing of GPS information from the base station to the Pocket PC will be done via USB. As an optional extended feature, the use of wireless communications can replace USB protocol to make the transfer of information more transparent. Once the vehicle returns from its route, detailed information about the path traveled can be uploaded to the base station for further analysis. Optionally, a RF receiver can be connected to the base station to receive live video feed from the vehicle. The RF receiver for live video view will introduce a live view display to the base station software if this is implemented.

The final baseline component is the vehicle, the Nikko H2 [5]. This vehicle was chosen because of its ability to carry all of the necessary components and still maintain a suitable speed for demonstrations. Along with all of the components, the vehicle will also need to carry the necessary power for all components during the entire route. The Pocket PC and the GPS will be powered by the Pocket PC's battery. The RF transmitter and the optional camera will be powered by a 9 volt battery. The microcontroller will need a 9 volt DC power supply and the optional sonar range finder will need a 5 volt DC power supply, which will be provided by a 9 volt battery stepped down with a transformer.

There are several features chosen for the baseline deliverables. As described above, the GPS guidance control system is one of the baseline features that will be demonstrated; this feature will contain GPS parsing from satellite data and will allow the vehicle to know its current location at any time. The next baseline feature will be a Pocket PC that houses the GPS unit and makes logical decisions as to the path the vehicle needs to take to reach its final destination. The data path that the Pocket PC will use for comparing current GPS positioning will be given by the base station via USB interface. The course corrections will be sent from the Pocket PC to the microcontroller via serial communications. The microcontroller will have outputs to the vehicle's remote control to control the car's movement. The base station will have a path selector and it will create a data file that will be sent to the Pocket PC via USB. Upon the return of the vehicle, the actual course taken will be sent to the base station via USB so that it can be shown on a map. The demonstration will show the complete workings of all components described above. The demonstration will show a path selected through the base station and being uploaded to the Pocket PC. The Pocket PC application will then be run and signals will be sent to the microcontroller to begin the cars movement. As the vehicle starts to move, corrections will be made to ensure that all points selected from the base station are reached. Upon finishing the route, information as to actual path taken will be sent to the base station via USB for a more detailed perspective.

Several stretch goals have been omitted from baseline to ensure that deadlines are met. The addition of a sonar collision avoidance system will give an assurance that the course can be completed even though there may be obstacles in the vehicle's path. This will in turn result in the possibility of more complex routes to be completed. Another extended feature will be the ability to display a live video feed from the vehicle through an RF camera to the base station. This will be nice because we can see any obstacles in its path. In addition, adding a digital compass can stabilize the movement of the vehicle since GPS data is not very precise. One more extended feature will be the ability to control different RF based remote control vehicles using this system. With minor adjustments to the main Pocket PC application and microcontroller software, controlling other RF vehicles should be similar in practice. Finally, adding a manual override via RF to the remote control car can prove useful in situations where the vehicle may sustain damage and will need assistance from an operator.

Project Tasks

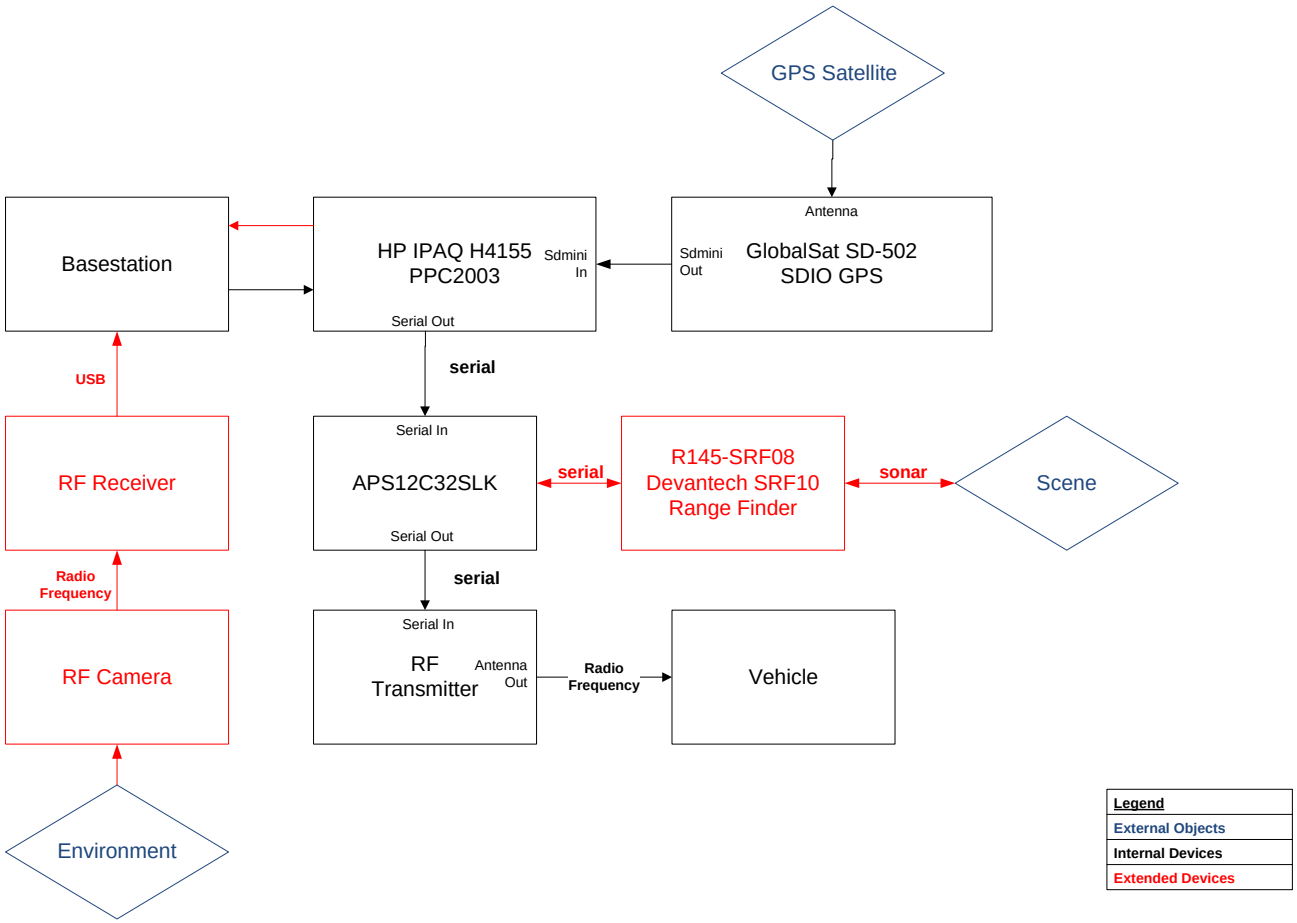
Task	People	Begin Date	End Date
Vehicle			
Select a Vehicle	Matt, Dariel, Toren, Michael	3/6/2007	4/3/2007
Combine Components onto Vehicle for Testing	Michael, Toren, Dariel	7/10/2007	8/7/2007
Optional: Sonar Integration	Toren, Michael	7/31/2007	8/21/2007
Acquisition of Parts	Dariel, Matt, Toren, Michael	4/2/2007	5/1/2007
Main Pocket PC Application			
GPS Data Parser	Dariel, Michael	5/14/2007	6/12/2007
Path Calculator Based on GPS	Dariel, Michael	5/28/2007	6/26/2007
Software for Serial Communication Interface with Microcontroller	Dariel, Michael	6/11/2007	7/10/2007
Application to Send/Receive Path From Base Station	Matt, Dariel	6/21/2007	7/21/2007
Base Station Application			
Map Interface to Select Location Points	Matt, Dariel	5/7/2007	6/7/2007
Planned vs. Actual Path Comparison	Matt, Toren	5/21/2007	6/21/2007
Optional: Real Time RF Video Feed	Matt, Toren	6/19/2007	7/10/2007
Microcontroller			
Circuit to Power Microcontroller	Dariel, Toren	5/7/2007	5/12/2007
Serial Interface Pocket PC	Michael, Toren	5/7/2007	7/7/2007
Interface to Vehicle RF Controller	Toren, Matt, Michael	5/7/2007	7/7/2007
Optional: Sonar Interface	Toren, Michael	5/7/2007	6/20/2007
Combined Pocket PC, Microcontroller, and Base Station Integration	Dariel, Matt, Toren, Michael	7/2/2007	8/7/2007

*Names shown in order of task responsibility.

Task Interfaces

The Pocket PC will be composed of several components: the GPS parser, path calculator, serial communications interface, and base station data interface. These subcomponents will be integrated on a modular interface and will be able to communicate with each other via programming platform standards. In order for the Pocket PC to communicate with the microcontroller we have decided to use the serial protocol standard. This will facilitate the transfer of information between the Pocket PC and microcontroller and allow for both components to make smart decisions on how to complete the mission. The microcontroller will interface with the vehicle's motor system via RF. This will be done by connect-

ing output pins of the microcontroller to the vehicle's remote control. This will allow us to send RF signals using the native control system of the vehicle. The base station will also need to interface with the Pocket PC. This has been greatly facilitated by the software provided by Microsoft, namely Activesync. We can calculate the desired route from the base station and transfer it via USB to the Pocket PC. The figure shown below shows a basic interface schematic for the Synapse UAV.



Component Integration and Testing

The issues encountered during integration are difficult, if not impossible to foresee. By thoroughly designing our protocols and physicals interfaces, we hope to avoid most of these issues. The team will have two people dedicated towards software integration and two people dedicated hardware

integration as to solve the problems that may arise. The possible areas of integration problems in our project are the base station to Pocket PC interface, the Pocket PC to microcontroller interface, the microcontroller to RF controller interface, and the Pocket PC software to microcontroller interface.

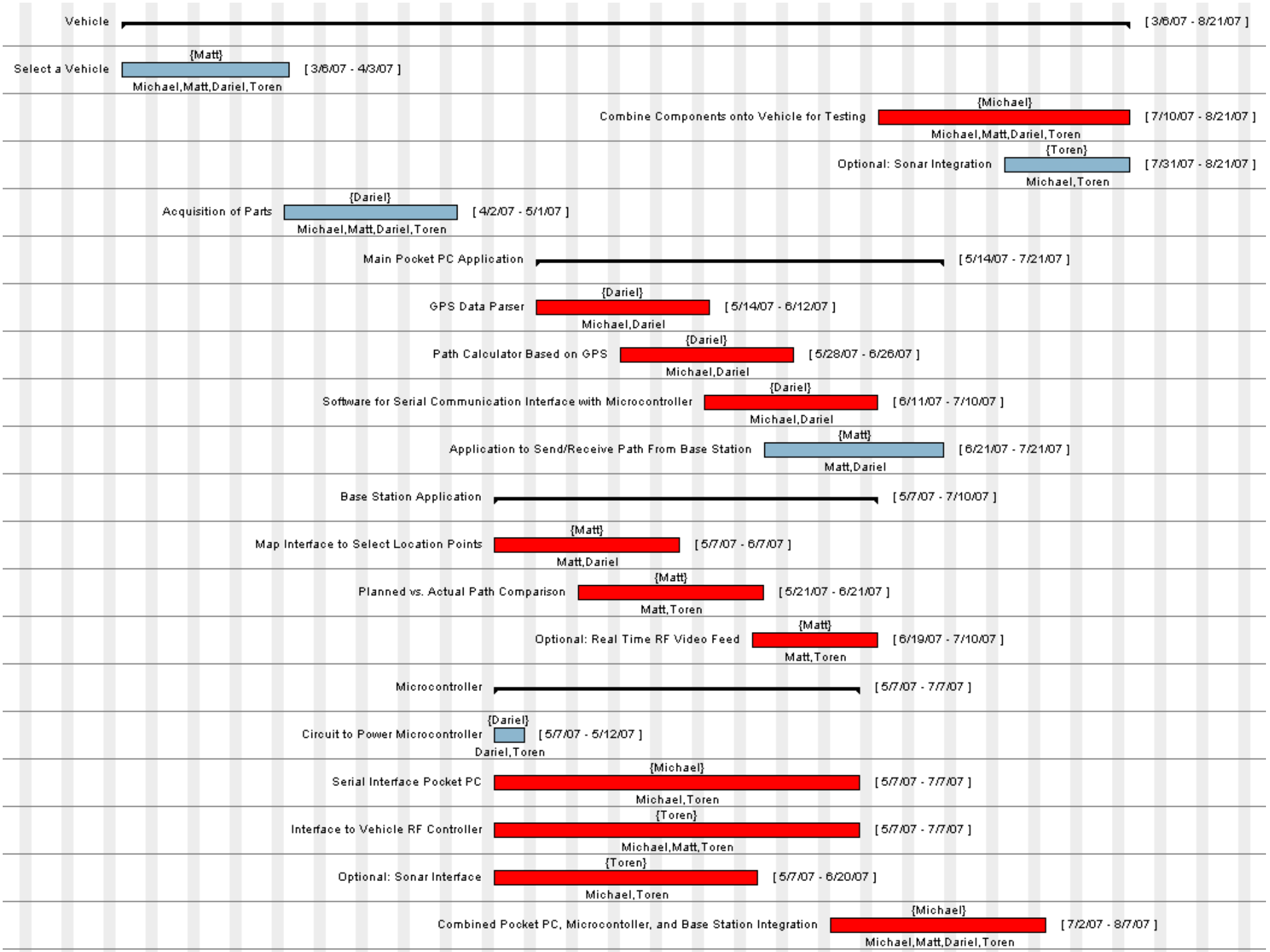
We have planned the project such that one large integration step does not occur. The people responsible for each task will be working on integration throughout, and not just at the end. For example, the task of getting the microcontroller to communicate with the Pocket PC is an integration step, because that task involves getting the Pocket PC's software and the microcontroller's software talking together. This same integration approach is carried out throughout the project. Also, we have planned our project so there is a long period between when we plan to integrate and when the project is due if we do run into any unforeseen problems.

Acquisition of Others Parts

- The FreeScale MC9S12C32 microcontroller to control the vehicle's motor system
- An IR transmitter to connect the Pocket PC to the microcontroller controlling the motor
- The C# GeoFrameworks library that will allow us to program the GPS using C#
- The RF camera to send images back to the base station
- Circuitry to make RF transmitter, power converters, etc.

Milestones

- Individual subsections completed
    - Pocket PC modules
        - GPS Parser – June 12th
        - Path Calculator based on GPS – June 26th
        - Serial Communications Interface – July 10th
    - Microcontroller
        - Sonar (extended) – June 19th
        - Pocket PC serial connection – July 10th
        - RF remote control interface – July 10th
    - Base station
        - Map interface to select location points – June 19th
        - Planned vs. Actual path comparison – June 21st
        - Real time RF video feed(extended) – July 10th


Vehicle
[3/6/07 - 8/21/07]
Select a Vehicle
{Matt}
[3/6/07 - 4/3/07]
Michael,Matt,Dariel,Toren
Combine Components onto Vehicle for Testing
{Michael}
[7/10/07 - 8/21/07]
Michael,Matt,Dariel,Toren
Optional: Sonar Integration
{Toren}
[7/31/07 - 8/21/07]
Michael,Toren
Acquisition of Parts
{Dariel}
[4/2/07 - 5/1/07]
Michael,Matt,Dariel,Toren
Main Pocket PC Application
[5/14/07 - 7/21/07]
GPS Data Parser
{Dariel}
[5/14/07 - 6/12/07]
Michael,Dariel
Path Calculator Based on GPS
{Dariel}
[5/28/07 - 6/26/07]
Michael,Dariel
Software for Serial Communication Interface with Microcontroller
{Dariel}
[6/11/07 - 7/10/07]
Michael,Dariel
Application to Send/Receive Path From Base Station
{Matt}
[6/21/07 - 7/21/07]
Matt,Dariel
Base Station Application
[5/7/07 - 7/10/07]
Map Interface to Select Location Points
{Matt}
[5/7/07 - 6/7/07]
Matt,Dariel
Planned vs. Actual Path Comparison
{Matt}
[5/21/07 - 6/21/07]
Matt,Toren
Optional: Real Time RF Video Feed
{Matt}
[6/19/07 - 7/10/07]
Matt,Toren
Microcontroller
[5/7/07 - 7/7/07]
Circuit to Power Microcontroller
{Dariel}
[5/7/07 - 5/12/07]
Dariel,Toren
Serial Interface Pocket PC
{Michael}
[5/7/07 - 7/7/07]
Michael,Toren
Interface to Vehicle RF Controller
{Toren}
[5/7/07 - 7/7/07]
Michael,Matt,Toren
Optional: Sonar Interface
{Toren}
[5/7/07 - 6/20/07]
Michael,Toren
Combined Pocket PC, Microcontoller, and Base Station Integration
{Michael}
[7/2/07 - 8/7/07]
Michael,Matt,Dariel,Toren

- Sections completed
    - Pocket PC main application – July 17th
    - Microcontroller – July 17th
    - Base station – July 17th
- Component integration
    - Combine Pocket PC main application, microcontroller and base station – August 7th
- Final build and integration
    - Put combined components onto vehicle for testing – August 21st

Risks

One of the main risks for this project is the possibility of not properly interfacing the Pocket PC to the microcontroller. The output of the Pocket PC is USB and the input to the microcontroller is RS-232. Interfacing these two ports is a high risk since we do not know how to perform this conversion. We believe that there are adapters that will perform this conversion, but until we learn more about these adapters, this is still a high risk. Once we have researched and tested these adapters to a greater degree, we are confident that this can be reduced from high risk. A medium risk is interfacing the microcontroller program to the remote control car's RF receiver. We know that certain pins are bridged in the remote control unit in order to control the car, but we have yet to fully understand how to connect the microcontroller to the remote control unit. We think we may be able to use MOSFET transistors as an electrically controlled switch. Another risk to the project is finding time as a team to design and produce all of the required components. We will try to meet during summer to complete the project before fall semester, but every member of the team will be taking summer classes and working. This could delay the timeline and push tasks forward, which could in turn delay the entire project and cause us not to finish on time. To prevent this from happening, we will meet at least once every week, to keep everyone on track. Another risk is the car will carry all of the components, so a crash could cause damage to these components. To mitigate this risk, we will run the vehicle in entrapped areas until we feel more confident about its abilities.

Current status of the project

Overview

The overview of the project has been completed at a high level. The hardware-level communications have been decided upon and most of the basic protocols between system parts are understood. Now, our main concern is deciding upon the software implementation and testing the protocols we understand at a high level.

GPS

The GPS system has been empirically tested for accuracy. It needs WAAS assistance to be within the 5m tolerance of our application, but this should not be problematic in the areas where the system is to be used (i.e. the United States). It interfaces with the Pocket PC by way of the SD card slot, and utilizes COM ports for software interaction. Now, it just needs to be tested with the software we write around it.

Sonar

The protocol for communication with the Devantech SRF10 Range Finder unit has been explored and has been found compatible with our micro controller's capabilities. It utilizes an I^2C serial communication protocol, in which the microcontroller serves as a master unit, sending a clock pulse to a variable number of slave sonar units. This pulse synchronizes with a data line signal shared by the master and slave units. In this way, the slave sonar units send data to the master, and receive data from the master quickly on the same data line. The next step is to obtain the part and tests its true abilities.

Car Controller

The Nikko H2 is a typical, commercially-available, radio-controlled car. Its controller uses a 45MHz transmit frequency to relay four distinct commands to the car, to control two independent motors. The controller could be made to interface with our microcontroller by way of a voltage and current amplifier attached to the control pins by way of mechanical switches. Each could be controlled independently using software, and sent through the microcontroller to control the car. The microcontroller will

provide a 4-bit output, using a proprietary protocol to control the car's controller. Now, we just need to start experimenting by applying different voltages to the controller from external sources to simulate what the microcontroller will do.

List of mentors and other resources

The intellectual resources that we have drawn on have been those of the companies that provided us with physical resources. This includes the web pages of the GPS subsystem providers, Devantech sonar, FreeScale, and Nikko Corporation. General web searches on various tidbits have been used as well. The ideas on how to bring all of these parts together have been entirely conceived within the project group.

Proposal summary

The most important lesson that the group has learned up to this point is the necessity for good communication among the members of the group. Each member must be flexible in their time and desires in order for the project to make progress. Organization of the tasks that need to be finished during the interval between meetings by each member is important to keep each group member utilized. Lack of such organization results in some members doing most of the work, and others not knowing how they can contribute. The team has found that by keeping the web page current with individual tasks listed along with regular meetings, the organization has vastly improved.

Bill of Materials

- Pocket PC (HP IPAQ H4155 PPC2003) – Donated by Okland Construction
- SD GPS Module (GlobalSat SD-502 SDIO GPS) – $129 (semsons.com)
- GeoFrameworks SDK – Free for educational purposes (geoframeworks.com)
- Remote Control Car (Nikko H2) – Donated by Matt Stoker
- Microcontroller (APS12C32SLK) – Donated (freescale.com)
- Microcontroller programming platform (PBMCUSLK) - Donated (freescale.com)

- Various circuitry (RF Transmitter, Power supply – resistors, capacitors, transistors, etc) - $30 (University of Utah parts room)

Vendors

Dariel Marlow
Okland Construction
1978 S West Temple
Salt Lake City, UT 84115
801.486.0144

Semsons – semsons.com
Address: 555 E. Live Oak Ave. Arcadia, CA 91006
Phone: (626)574-5557
Email: info@semsons.com

FreeScale – freescale.com - 800.521.627

Bibliography

[1] HP H4155. [Online] HP. [Cited: April 14, 2007.] http://h10025.www1.hp.com/ewfrf/wc/product?lang=en&lc=en&cc=us&dlc=en&dest_page=product&product=425743.

[2] Microsoft C#. *Visual C#.* [Online] Microsoft. [Cited: April 14, 2007.] http://msdn2.microsoft.com/en-us/vcsharp/default.aspx.

[3] GeoFrameworks. [Online] GeoFrameworks. [Cited: April 14, 2007.] http://www.geoframeworks.com/.

[4] Freescale MC9S12C32. [Online] Freescale. [Cited: April 14, 2007.] http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=MC9S12C32.

[5] Nikko RC. [Online] Nikko. [Cited: April 14, 2007.] http://www.nikko-rc.com/.