Senior Project Proposal: Project WEAVER
Wireless Enabled Active Video Experimental Rover

EE/CS 3992

Tyler Lloyd
Amber Blake
Janos Opra

September 3, 2004

Table of Contents

INTRODUCTION

This paper will detail our intention to design a wirelessly controlled vehicle (rover) equipped with a camera and a collision avoidance system as a senior design project. The rover will be connected to a laptop using an 802.11g wireless transceiver. The user will control the rover by entering commands at the laptop that will be decoded and transmitted to the rover over the wireless link. The rover will check these commands against an onboard collision avoidance system that will override the user's instructions to prevent a collision. There will be a forward facing camera mounted on the rover that will provide a real-time video feed to the laptop via the wireless link. The laptop will have a software program running that will display the real-time video feed from the rover and accept user input. To demonstrate the functionality of the system, the rover will be guided along a track that can't be seen by the user except through the camera mounted on the rover.

MOTIVATION

This project was selected because of its scope and versatility. While this project offers the opportunity to learn about wireless transmission technology and video compression, it also allows us to select which portions we will develop and which will be purchased. This project has a great deal of room for creativity in added features should time allow.

DESIGN OVERVIEW

The entire system can be summarized by referring to the block diagram in Appendix A. The bill of materials associated with this high level overview is included in Appendix B.

From the block diagram, it can be seen that our project entails two major blocks one block contains the work required on the PC and the other is the work for the rover. The PC block really does not contain a hardware aspect. We will be using an off the shelf wireless card and set it up as a socket. The program can then access the information as if it were writing to or reading from a file. The main program is responsible for displaying a video stream that will be sent from the rover. The Motion Control section is responsible for receiving the User commands and sending commands to the rover. Please see below in the Project Tasks section for a more complete explanation.
The second block is the Vehicle (rover). The rover has a lot of hardware associated with it. The rover itself is a modified RC vehicle. We are using a RC truck, because it provides more space, but the actual chassis of the vehicle is unimportant. At the heart of the implementation is a DSP. The DSP will be programmed using code we will create to control all the other systems on the rover. An interface with the PC will be established using a PC Card interface (similar to IDE) and a Wireless PC Card. The PC Card interface itself will be a Linux driver similar or identical to that used on the PC. The Camera that will be used is a commercial 802.11b enabled camera. We chose to use this camera due to the difficulty we experienced in obtaining a video encoder integrated circuit. Motion control will be handled by using H-bridge motor drivers and Pulse Width Modulators (PWMs) available on the DSP. The IR LEDs and Sensors shown in the block diagram are for motion control. They are driven using a PWM and are read back by the DSP. Two battery packs will be included on the car, one for the motors and another for everything else. The voltages for all the parts excepting the motors will need to be regulated. Linear power regulators will be used. Using switching regulators is more efficient, but it also requires more board space and more components.

Connecting these two blocks is a wireless interface. The Wireless standard that we chose to use is IEEE 802.11b. We chose this protocol to accommodate the interface of the camera.

PROJECT TASKS AND RISK ASSESSMENTS

The necessary tasks to complete the project can be divided into two separate categories: hardware and software. All hardware requirements are embedded on the rover, and the software is either embedded on the rover in a Digital Signal Processor (DSP) or on the laptop (PC) as defined below. The hardware requirements are listed first, followed by the software.

At this point it is helpful to know that the DSP software development will be performed using the Codewarrior Software Development Kit (SDK). An evaluation copy of the SDK is included with the development kit that we have. Codewarrior uses C as the source language. PC Software development will be performed using Microsoft Visual Studio available in the Computer lab. We intend on using C/C++ for the PC software.

POWER REGULATION CIRCUITRY

This task is to design the circuitry necessary to regulate the power coming from the batteries. Two battery packs will be used. One battery pack will be used for the motors and the second pack will supply the rest. The input to this circuit will be the voltage coming from the two battery packs, and it must output the correct voltage level to power the custom circuitry on the rover, including the DSP, camera, and drive motors. The risk associated with this task is low. We have used switches and linear regulators for other classes and feel confident that we can meet our needs.

DSP CIRCUITRY

This task will involve the design of any hardware necessary to program and run the DSP. This circuitry will be located on a printed circuit board and will include a crystal clock, bypass capacitors, and appropriate programming connectors. It may include external memory or voltage level changing circuitry as necessary. The actual circuitry is easy and straightforward, as most DSPs that we have looked at have suggested crystals and auxiliary components. For this reason the risk is low.

WIRELESS TRANSMISSION CIRCUITRY

The purpose of this hardware component is to interface the DSP to the hardware necessary to transmit the encoded video stream, and motion commands over the 802.11g wireless link. This will include the 802.11g reference card that we plan to purchase, mounting connectors, power conditioning and any other special hardware necessary to perform this operation. The input to this circuit will be the data from the DSP to be transmitted and the outputs will be those required by the PC card to transmit that data over the wireless link. We plan to use a PC Card and an open source Linux driver to facilitate this wireless transmission link. This makes the hardware design fairly simple as only a connector and a few bypass capacitors should be required. The resulting risk assessment for the hardware development is low.

COLLISION AVOIDANCE CIRCUITRY

This circuit will include all of the hardware required to drive the IR LEDs and detect objects. This circuit will drive the LEDs using a square wave generated by a PWM, and output a signal to the DSP when an object is detected. This circuitry will include the LEDs, IR receivers, and a PWM pin from

the DSP. Parts are readily available, and we have experience working with these. For these reasons risk is considered to be low.

MOTOR DRIVER

This task is to design the hardware that will take inputs from the PWMs on the DSP and drive the motors in the desired fashion. This circuitry will include an H-bridge, and control circuitry. It may also include voltage level converters, and power conversions depending on what circuitry has already been provided on the rover. Velocity will be controlled by the duty cycle of the PWM; the maximum duty cycle is 100% and lowest is 0%. The external resistance to the wheels will also affect the speed. The rover will not compensate for inclines, or higher or lower dynamic friction environments. This compensation will be left to the user. We have found reference designs for the H-bridge but still need to investigate the motor requirements. It is possible that the motors cannot handle a 100% duty cycle. In this situation the percentages should be considered covering the maximum allowable range of the motor. Risk should be medium.

PCB DESIGN AND VERIFICATION

The purpose of this task is to combine all circuits to be placed on the printed circuit board into a master, and place and route all components. The resulting master schematic, board layout, and component footprints will then be verified prior to the construction of the printed circuit board. A member of our group has a lot of experience working with PCB design and verification. He feels that this part should be manageable with fairly low risk.

BASIC ASSEMBLY

This is the last hardware task and includes all miscellaneous assembly tasks required to create a functioning rover: populating the PCB; attaching the motors, sensors, and camera to the PCB; and mounting necessary components on the rover. We are confident in our ability to mount the components onto the rover and, while using surface mount packages, feel the risk to complete this task is low.

WIRELESS COMMUNICATION

This software will have portions residing both on the DSP and on the laptop. It will be necessary to place software on the DSP to provide the communication protocol the wireless transmission circuitry requires to transmit data through the 802.11g reference card to the Laptop. Software will also be necessary on the laptop to receive the data transmitted by the rover, and also to transmit commands through the 802.11g PC card to the rover. This task will be completed when the rover and PC can reliably transmit and receive data. Since we will be using a PC driver, the software on the laptop has a low risk. The risk for the rover software will be medium because even though we plan on using a Linux driver, we will still need to write the high-level protocol stack layers.

MOTION CONTROL

In order to provide control of the motors on the rover, software will be written both for the DSP embedded on the rover and for the PC. The software on the DSP will be immediately responsible for the movement of the motors by issuing appropriate commands to the motors via the PWMs. The software on the PC will be indirectly responsible for the movement of the motors as it will take the instructions from the keyboard convert them to command codes, and transmit the command codes to the rover.

The user will be allowed to enter two key requests into the GUI to control the car. The first is a direction key. The command for forward or reverse sets a new speed in that direction. The command for left or right only turns the wheel and a velocity in the forward or reverse direction is required to actually change the direction of the rover. The second key used is an intensity key. The intensity uses the number keys 0 through 9 as a fraction of the maximum. A separate intensity will be stored for left/right and forward/reverse. Alternatively, if time permits, this interface may be modified to allow the use of a joystick. The PC will translate this information into a message to the car. The message will contain an ASCII character direction and a two digit percentage 01 – 99, followed by the same for left and right. The following table summarizes the command codes.

Motion	Key	ASCII	
Forward	w	0x77	
Reverse	s	0x73	
Left	a	0x61	
Right	d	0x64	
Stop	"space"	0x20	
Keep Alive	U	0x55	
Intensity	Percentage of Maximum	Intensity	Percentage of Maximum
1	09	6	59
2	19	7	69
3	29	8	79
4	39	9	89
5	49	0	99

MOTION CONTROL COMMAND CODES

These commands will be transmitted on a regular basis. If no new user input is required a keep alive character of U may be used. The intensity is irrelevant and any value can be used when sending a keep alive. If the rover does not receive a message, emergency stop procedures will commence. This task will be completed when the PC can send commands to the rover, and the rover can comply with the commands and move accordingly. We will need to learn more about PWM and motion control before writing the code on the DSP resulting in a medium risk. The software on the laptop will need to read input from the keyboard, decode the input into a command, and then transmit the command. Risk for the laptop software is low as we have experience with this type of software.

SENSOR PROCESSING AND COLLISION AVOIDANCE

This task involves the design of the software on the DSP necessary to take readings from the IR LEDs, detect obstacles, and direct the motors in order to avoid a collision. Successful completion of this task will be indicated when the rover detects obstacles and employs evasive action. Since the sensors will indicate whether an object is detected or not, the software to perform this task should be fairly simple resulting in a low risk assessment.

VIDEO INTERFACE

This software will reside entirely on the PC. Software will be added to the main application in order to receive the video stream from the 802.11b camera and display it on the monitor. We don't have

very much experience with video; however, we have a lot of resources to draw information from. The risk for this task is medium.

FRAMEWORK SOFTWARE

This software will reside on the DSP, and will provide functions to access the serial port, debugging LEDs, and any code necessary to initialize the JTAG interface. The JTAG interface will be used to program the DSP and to provide an interface to debug the code when it is running on the DSP. Since one of the members of our group has experience with this particular DSP, the risk to complete this task is low.

GUI WRAPPER

This is the application running on the PC that integrates the video stream viewer and the command console that will accept movement commands from the user and relay them to the rover. Completion of this task will allow the user to control the rover and view the transmitted video on the monitor. Although, no one in our group has had experience working with a GUI in C/C++, we feel that the risk associated with this task is of a low level.

The following table summarizes the level of risk that we anticipate with each project as explained previously.

FOCUS AREA	TASK	RISK ASSESMENT
HARDWARE	POWER REGULATION CIRCUITRY	LOW
HARDWARE	DSP CIRCUITRY	LOW
HARDWARE	WIRELESS TRANSMISSION CIRCUITRY	LOW
HARDWARE	COLLISION AVOIDANCE CIRCUITRY	LOW
HARDWARE	MOTOR DRIVER	MEDIUM
HARDWARE	PCB DESIGN AND VERIFICATION	LOW
HARDWARE	BASIC ASSEMBLY	LOW
DSP SOFTWARE	WIRELESS COMMUNICATION	MEDIUM
PC SOFTWARE	WIRELESS COMMUNICATION	LOW
DSP SOFTWARE	MOTION CONTROL	MEDIUM
PC SOFTWARE	MOTION CONTROL	LOW
DSP SOFTWARE	SENSOR PROCESSING AND COLLISION AVOIDANCE	LOW
PC SOFTWARE	VIDEO INTERFACE SOFTWARE	MEDIUM
PC SOFTWARE	GUI WRAPPER	LOW
DSP SOFTWARE	FRAMEWORK SOFTWARE	LOW

RISK ASSESSMENT TABLE

PRELIMINARY TASK ASSIGNMENTS

All of the tasks listed above have been tentatively assigned to team members. The following table lists each of the tasks and the team member mainly responsible for the completion of each task.

TASK	RESPONSIBLE TEAM MEMBER
POWER REGULATION CIRCUITRY	JANOS
DSP CIRCUITRY	TYLER
WIRELESS TRANSMISSION CIRCUITRY	AMBER
COLLISION AVOIDANCE CIRCUITRY	AMBER
MOTOR DRIVER	TYLER
PCB DESIGN AND VERIFICATION	ALL
BASIC ASSEMBLY	ALL
DSP WIRELESS COMMUNICATION	JANOS
PC WIRELESS COMMUNICATION	AMBER
DSP MOTION CONTROL	TYLER
PC MOTION CONTROL	JANOS
SENSOR PROCESSING AND COLLISION AVOIDANCE	AMBER

PRELIMINARY TASK ASSIGNMENT TABLE

INITIAL SCHEDULE

Our initial planned schedule flow contains the projected completion dates of major milestones and is detailed in the Gantt chart included in Appendix C. Please note that all of the hardware design will be completed in the summer so that the printed circuit board will be ready by the beginning of the fall semester.

TASK INTERFACES

To provide the final functionality of the rover, the preceding tasks must interface together. The interfaces necessary to provide streaming and viewing of video, ability to control the rover, and collision avoidance will be discussed.

In order to display streaming video from the rover on the monitor, the wireless transmission circuitry, camera interface circuitry, wireless communication software, video interface software, and GUI wrapper must all interface together. The camera will provide the video feed to the laptop via the 802.11b connection. Once the signal has been received by the laptop the video interface software will display the image on the monitor through the GUI wrapper. We will be using the UDP transmission Protocol for the wireless link.

The user will be able to control the rover as a result of the cooperation of several components: the wireless communication software, the wireless transmission circuitry, the motor driver hardware, and the motion control software. Commands will be entered at the laptop where they will be encoded by the PC side of the motion control software and transmitted by the wireless communication software to the rover. After transmittal the commands will be received by the wireless transmission

circuitry on the rover, and transferred to the DSP where the motion control software will provide the appropriate signals to the motor driver circuitry that will turn motors in the desired fashion.

The collision avoidance circuitry, the sensor processing and collision avoidance software, and motion control software will work together in order to avoid collisions. The collision avoidance circuitry will detect an obstacle and transmit a signal to the sensor processing and collision avoidance software on the DSP. The software will determine what course of action to take and communicate with the motion control software appropriately.

INITIALIZATION

Initialization of each component is necessary in order for the functionality of the system to be realized. The exact specifications for each hardware device are not yet known, but the DSP will need to provide any necessary initialization signals to the hardware devices on the rover. These could include initial stop signals to the motors, and any initialization necessary for the wireless PC card to transmit. All of the code for the DSP will be written on a PC, and then downloaded into memory on the PCB on the rover where it will run. Thus, we will not be able to alter the behavior of the rover due to this code while it is running. On the PC side, a socket will be opened to facilitate wireless communication with the rover. It will then wait to receive the heartbeat signal from the rover before allowing the user to enter commands to send to the rover. This will prevent the PC from attempting to transmit to the rover before it is ready to receive commands. Once the heartbeat signal is received, the GUI will be displayed, video data will be displayed as it is received, and commands will be sent to the rover as they are issued by the user.

DEBUG STRATEGY

Our project can be broken down into three main groups. They are wireless communication, motion control, and video interface. Each of these is going to be worked on in two parts. One will be on the rover and the other on the PC laptop. While one team member is working on the rover another will be working on the laptop interface.

Wireless communication will be the first main part to be started in September. Then the motion control will be worked on starting the last week in September, lasting for three weeks. Finally the video interface will be started and should be done by the last part of October.

Each of the three main parts will have to work independently before they will be implemented together. The Wireless communication will be complete when we can establish a link between the rover and the laptop and data can be exchanged. At this point the content of the data is not important. For the motion control group there will have to be correct movement of the rover when the inputs are given from the laptop though not necessarily through the wireless link. Finally the video system will need to send a stream to the laptop where it needs to be displayed. Once all three parts are working by themselves we will integrate them and fix any problems that arise.

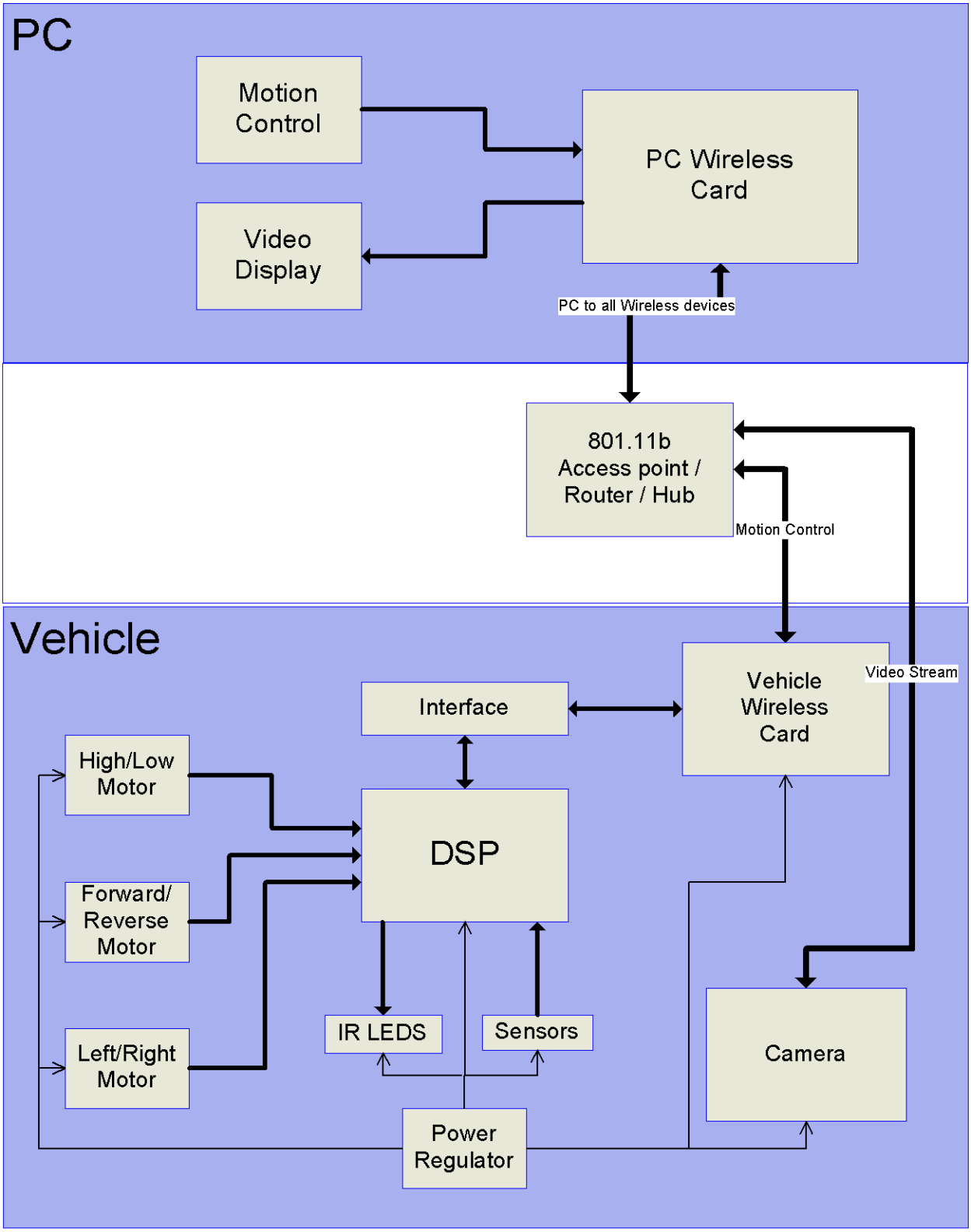
This modularization will aid us in the testing and debugging of the entire system. Another tool that we will have available is a standard RS-232 serial communication port. The port will be on the rover and will provide communication to the rover before the wireless link is implemented.

CONCLUSION

The schedule to complete these tasks would be fairly tight if they were all to be completed during the fall semester. Since most of the hardware will be completed over the summer, the actual schedule during the fall semester should be fairly open. The riskiest components of this project are the wireless transmission circuitry and software, and the video compression. The risk associated with the wireless transmission hardware and software is due to a lack of experience within the group. If we find an open source Linux driver and determine the exact interfaces, this task will pose less of a risk. We intend on using a pre-purchased 802.11b enabled camera (webcam style) that outputs a digital signal, sending it over the wireless link, and then inputting it into the video driver on the PC. We hope to be able to integrate all of these components together into to a project that not only demonstrates our engineering skills, but that is also fun.

Appendix A

BLOCK DIAGRAM



APPENDIX B

BILL OF MATERIALS

Part	Description	Manufacturer	Cost	Status / lead time
Laptop	User interface	Non Specific	-	Acquired
Laptop wireless interface	802.11abg PC Card	3COM/atheros based	-	Acquired
Visual Studio	PC development environment	Microsoft	-	Acquired
RC vehicle	RC vehicle with motors	Radio Shack	$37.28	Acquired
Power regulators	Circuitry required for power	Non Specific	Sample	Acquired
DSP	PWM, wireless control, processor, motion control	Motorola	Donation	Acquired
Rover wireless interface	Transmit compressed video and receive control	Dlink, Netgear	$60	Acquired
Camera	Generate raw video stream	WIS	$179.00	Acquired
IR LEDs	IR transmitter for obstacle avoidance	Digi-Key	$0.40/each	Acquired
IR detector	IR receiver for obstacle avoidance	Digi-key	$1.02/each	Acquired
Main Motor driver	Integrated H-bridge for main drive motor	Allegro	Sample	Acquired
Auxiliary Motor driver	FETS and supporting logic for auxiliary motors	Non Specific	Donation	Acquired
MPEG encoder	Convert raw video to MPEG2	WIS, NEC, Cirrus	$300-500	Need research
PCB	Interface for all hardware on the rover	Circuit Graphics	Free for first run	1 week
Miscellaneous	Various passives, logic gates, and connectors	Non Specific	Donation	Acquired

APPENDIX C

TASK SCHEDULE GANT CHART

Project WEAVER Schedule

Group Members: Tyler Lloyd, Amber Blake, Janos Opra

D indicates the date that the task will be completed by. Done indicates tasks that are completed.

Hardware		August			September			
Tasks	point-person	13	20	27	3	10	17	24
Power regulation	Janos			Done				
DSP circuitry	Tyler			Done				
Wireless transmission circuitry	Amber			Done				
Collision avoidance circuitry	Amber			Done				
Motor driver	Tyler			Done				
PCB design	All			Done				
PCB build								
Part aquision	All			Done				

Software & Last Hardware Task		September				October					November				December
Tasks	point-person	3	10	17	24	1	8	15	22	29	5	12	19	26	3
Basic Assembly & Hardware Documentation	All				D										
PC Wireless Communication	Amber							D							
PC Motor Control	Janos							D							
DSP Framework Software	Tyler							D							
PC Video Interface	Amber										D				
DSP Wireless Communication	Janos										D				
DSP Motor control	Tyler										D				
Video IV/D	Amber											D			
Wireless IV/D	Janos											D			
Motor Control IV/D	Tyler											D			
Collision Avoidance	Amber														D
Documentation & Test	Janos														D
Documentation & Test	Tyler														D

IV/D Indicates an extra week to integrate and document the task. We have added this week to the tasks that we believe will be the most difficult.