The Vision Enhanced Car

The Pennsylvania State University at Erie, The Behrend College
Jeffery Betts
Roger Grayson
Stephen Haug
William Wykoff
Advisor: Dr. Randall Fisher
CENBD480 & 481
April 26, 2004

Table of Contents

Abstract

This paper describes the design of the Vision Enhanced Car (VEC). The description includes the problem statement, the needs assessment and related background research, the engineering requirements, the design methodology with system components and algorithms, the testing of the components and system, the realization of requirements, constraints, and standards, the project management plan, the development costs, and the results and conclusions of the project.

The VEC is a system designed to improve the safety of a vehicle by alerting drivers to any objects near the vehicle and problems detected through the On Board Diagnostics, level II (OBD II) system, using both visible and audible alerts. The VEC uses ultrasonic distance sensors, a Personal Digital Assistant (PDA), and a speech synthesis system to detect and alert drivers to objects around the vehicle. The VEC also alerts the driver to unsafe conditions in the vehicle itself whenever the vehicle is in use. To do this, the VEC accesses the OBD II system of the vehicle, allowing the user easily identify and correct error conditions in the vehicle, and thus keep the vehicle safe to drive.

The results of our testing showed that the system functions correctly. It was tested in a garage, parking lots, and highways. The sensors correctly reported any objects that were within range, and the PDA and Winbond output visual and audible warnings when necessary. The ELM322 accurately reported the vehicle's speed, the number of Diagnostic Trouble Codes present, the Vehicle Identification Number, and any actual trouble codes. The speed and trouble codes were displayed on the PDA.

If more time was allotted this project could be fabricated into a production model. The internal unit could be put onto a PCB and then encased. The sensors could be upgraded to a weatherproof model and the sensor wiring could be redone to conform to any vehicle wiring standards. Upon completion of these minor tasks the product would be marketable.

1. Problem Statement

1.1 Needs

In an effort to make vehicles safer, we have decided to build a system to help enhance the driver's awareness, and to alert the driver to problems with the vehicle.

By helping drivers avoid objects in vehicle blind spots, the VEC helps eliminate minor accidents and also helps keep insurance rates lower. Many drivers, especially shorter drivers, have difficulty seeing around all points on their vehicle, and some people have large vehicles that have large blind spots. A system is needed to help drivers see not only stationary objects such as walls and curbs, but also help them locate moving objects such as vehicles and children. The average person would find such a system useful not only when driving in reverse, but also when parking or driving (for example, when passing another vehicle on a highway).

The VEC also helps make vehicles themselves safer. By allowing the user to read the trouble codes from a vehicle's OBD II system, potential problems with many of the vehicle's systems can be identified for repair. This includes problems such as those with the transmission, fuel system, and engine cooling system.

1.2 Objectives

The main goal of the VEC is to make vehicles safer. To do this, there are two sub-objectives: object detection and vehicle monitoring and alerts. The system will also be able to be installed by either the user or a mechanic, not interfere with the normal functions of the vehicle, and be as aesthetically pleasing as possible. Finally, the system will be securely fastened to a vehicle, resistant to environmental factors, and, most importantly, accurate.

To detect objects, we will build a system that will assist in avoiding blind-spot accidents, assist in parking, and help increase driver confidence behind the wheel. The system will give accurate verbal and visual warnings about objects detected, without distracting the driver. The user interface will be simple and unobtrusive, and be able to be used while driving.

To monitor the vehicle and give alerts to problems, the VEC will access the OBD II system in the vehicle. Any trouble code detected will cause an alert to be sent to the user. The trouble codes will also be accessible manually, as well as be able to be cleared by the user.

2. Needs Assessment and Background Research

To gather information about our product, two different informal verbal surveys were conducted. One survey was conducted at Penn State Erie, The Behrend College at Dobbins Hall. This survey covered a variety of people from different backgrounds. Another survey was conducted at Sam's Club in Erie. This survey covered truck drivers and Sam's Club employees. The results from both surveys were combined, and the following is a list of features people thought the Vision Extended Car (VEC) should do:

- Audio and Visual Alerts
- Talk in English
- Keep distractions to a minimum
- Low cost
- Customizable alerts
- Compass
- Monitor warning lights
- Video output
- Increase noise as delta X gets smaller
- Weather alert
- Touch screen display
- LOUD!
- Blind spot sensor
- "Safe to Pass" alert
- Time until impact alert

We interpreted this data into customer needs, and came up with the hierarchy shown in Figure 1, on the next page.

2.1 Needs Hierarchy

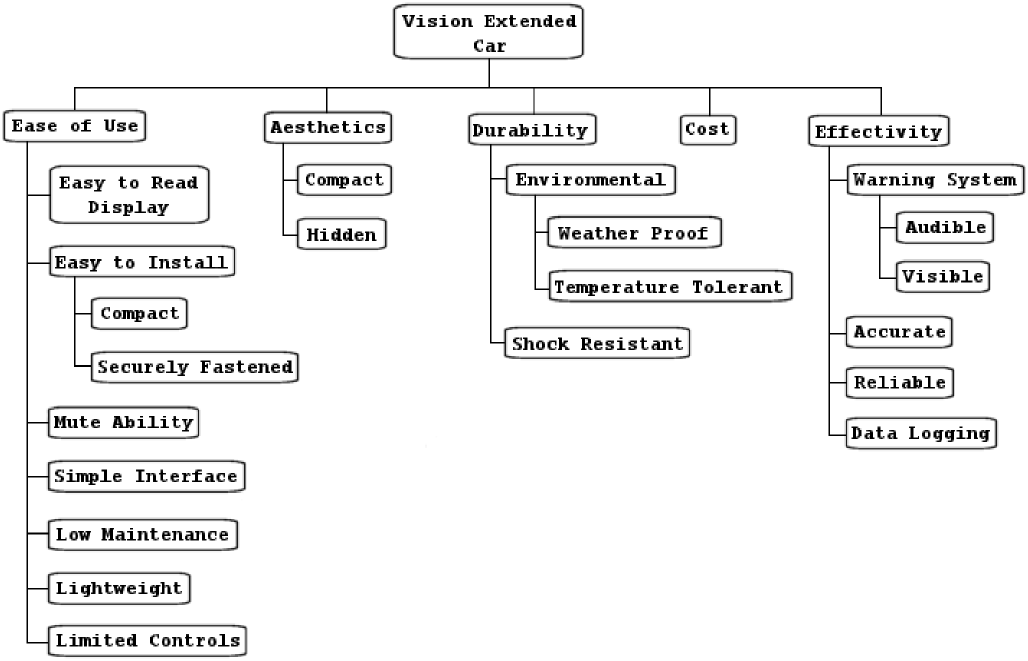


Figure 1: Needs Hierarchy

2.2 Needs Ranking

We determined that effectivity was the most important need since a system that did not work properly would be useless. Aesthetics and durability were determined to be the second most important needs. Durability is important because a system that falls apart is also pretty useless. Aesthetics are important because people tend to judge books by their covers – a system that is "ugly" will probably not get used. Ease of use was determined to be less important than the first three because a system that is slightly complex can still be effective and help prevent accidents. Cost was considered to be the least important need because one avoided accident can save thousands of dollars, and, even more importantly, may save lives.

The needs are summarized in Table 1, shown on the next page.

		Ease of Use	Aesthetics	Durability	Cost	Effectivity
		A	**B**	**C**	**D**	**E**
A	Ease of Use	▨	B	A,C	A	E
B	Aesthetics	▨	▨	C	B	E
C	Durability	▨	▨	▨	C,D	E
D	Cost	▨	▨	▨	▨	E
E	Effectivity	▨	▨	▨	▨	▨

Rankings

E	Effectivity	4
B	Aesthetics	2
C	Durability	2
A	Ease of Use	1.5
D	Cost	0.5

Table 1: Needs Ranking

2.3 Background Analysis and Research

There are five main areas in the VEC that play a major role in the project. These are the display, the microcontroller, the sensors themselves, the OBD II connection, and finally the text-to-speech operations. Selection of quality parts that work well together is essential for a successful project. The remainder of this report has been divided into discussions on the individual parts as stated above.

2.2.1 Text-to-Speech and other Voice Options:

The Vision Enhanced Car employs audio alerts and information so the user can more effectively concentrate on his or her surroundings. There are two main methods that could be used to implement such features. The first method would be to use a voice recorder chip with a fixed amount of time on it, and preprogram all possible outputs that the chip would need to speak. The second method would be to use text-to-speech synthesizer.

Using a text-to-speech chip allows for a more flexible integration of voice in the project, as this eliminates the constraint of the limited memory of the recorder chips. Furthermore, it could allow user configuration of what is or is not spoken by the system. Finally, expandability and upgradeability are greatly increased since additions would not require more memory or more recording chips.

While simple voice recorder chips cost around $10, a good quality text-to-speech synthesizer costs around $35. This slightly increased cost is justified as the added features of the chip are numerous, as stated previously.

One such chip is made by Winbond, the WTS701EF/T, and is a “one chip solution” for text-to-speech translation. The retail price of this chip is $33. It comes in a TSOP package, and this particular model features a female voice. The chip is programmable through the SPI port. The chip requires an external crystal oscillator to function, and using another microprocessor, could even be feasibly run through an EIA-232 port or I^2C connection.

Power for the chip comes from a 3.3V supply. During operation the chip typically only uses 35mA of current, and when put into standby mode consumes only 1uA.

2.2.2 Display

There are many different solutions that could be used for a display in the VEC. Some of these include a small graphic LCD, a small touch-screen display, light emitting diodes, and the possibility of interfacing with a PDA or a Personal Digital Assistant (PDA). Since the latter two combine the display with controls for the system, it is easily seen that using a small graphic LCD adds to the complexity of the project because of the need for extra buttons or controls. Also, using an LCD would require extra circuitry such as an LCD controller. Adding circuitry to the system goes against the needs of ease of use, where it should have a simple interface and limited controls, and aesthetics, where it should be compact and hidden.

A small touch screen display keeps the system small and compact, while maintaining a simple interface and minimal controls. There are three types of touch screen interfaces: resistive touch, capacitive touch, and acoustic wave. Resistive touch is the only of these types that is both temperature tolerant and shock tolerant. There are two types of resistive touch screens: glass on glass (GG) and film on glass (FG). Film on glass is the lower cost solution of the two, and recently has become more versatile than glass on glass. Figure 2, on the next page, shows a comparison of the two.

Property	Glass on Glass	Film on Glass
High-temperature operation	Yes	Yes
High-humidity operation	Yes	Yes
Vibration Resistance	Good	Good
Shock Resistance	Good	Good
Impact	Poor	Good
Resolution	Low	High
Cost	High	Low to medium

Figure 2: Comparison between GG and FG [1]

Resistive touch screens are layered with each layer being separated by small invisible spacers. When someone touches the screen, the top layer is pressed down and comes in contact with the lower layer. The touch screen then outputs a voltage in both the X and Y direction corresponding to the place on the screen that was pressed.

Touch screens are relatively costly. The cheapest touch screen found was around $249, while the cheapest touch screen add-on found was around $150. Because of the high costs, a touch-screen is not a good solution for the display. While it helps keep the system compact, it greatly increases the cost and does not provide any room in the budget to purchase other parts for the system.

Light emitting diodes provide an extremely cost efficient display device, but limit the possible display output. Using light emitting diodes restricts the display from being variable, meaning the display would have to be fixed and unchanging.

Interfacing with a PDA provides all the advantages of using a touch-screen display, since they incorporate a touch-screen as the primary input. The drawback to using either of these is that special software must be written in order to interface with them. The fact that Penn State Behrend already has a PDA available for use means that there is no cost, which is a great advantage. Since the cost is not an issue, the drawback to using a PDA is minimal. Further, many people own a PDA, so the cost will not be passed on to those people.

2.2.3 Vehicle Monitoring

To assist the microcontroller (MCU) in determining which driving mode the user is in, the VEC needs information about the vehicle's states, both current and recent past. To access the needed information, there are three options. The most difficult and expensive of these options is to place sensors inside the vehicle on various engine parts, and display the output on gauges inside the vehicle. While effective, this would not enable the MCU to access the information. A more elegant solution would be to link the sensors to the MCU and integrate the sensors to one display. Fortunately, most vehicles in service today already have many sensors embedded inside, all linked to an ECU (Engine Control Unit). The simplest and cheapest solution, then, is to connect to the ECU and request information from it.

There are two methods of interfacing to the ECU. The most difficult is to find the ECU in the vehicle, determine what communications protocol and interfaces are needed, and build it. Obviously, this would take a lot of research and time, and could be different for each automobile make or model. A simpler solution is to use the interface already built into most vehicles.

For pre-1996 vehicles, the interfaces are not standard; as a result, every manufacturer implemented a different protocol and used different connectors. Some manufacturers even used different protocols and connectors on different models. From 1996 onward, vehicles use the OBD II standard. Due to the standardization of OBD II, and the preponderance of post-1996 vehicles, the VEC will interface through OBD II.

OBD II is a federally regulated standard, mandating the connector, a minimum set of vehicle components to be monitored, and the set of diagnostic test signals between the ECU and any device attached. The connector required is a 16 pin DCL, or Diagnostic Connector Link.

OBD II allows external devices to access the ECU and request a variety of information including:

- Engine RPM
- Coolant temperature
- Calculated engine load
- Intake manifold pressure
- Airflow rate
- Intake air temperature
- Timing advance
- Fuel pressure
- Fuel system status
- O2 sensor voltage
- Battery voltage
- Vehicle speed

In addition, the ECU can communicate trouble codes though the OBD II interface. Trouble codes indicate that the ECU has detected an error, as determined by the sensors it monitors. These errors could be in several systems, including the transmission, emissions system, ignition system, power train, and even the body.

The format for messages between the ECU and external devices is set forth in the SAE J1979 standard. However, the protocol for communication varies between different automobile manufacturers. Fortunately, there are currently only three standards used: ISO 9141 (European cars, Chrysler, and most Asian cars), SAE J1850 VPW (Variable Pulse Width Modulation, on GM cars), and SAE J1850 PWM (Pulse Width Modulation, on Ford cars). Communication between the ECU and an external device (like an MCU) could be accomplished using those protocols. However, a simpler solution would be to use a converter, for example one made by ELM Electronics, who manufactures chips to convert each protocol to EIA-232, a format easily understood by most MCUs.

A further use of the OBD II interface and its output is for data logging. Since the MCU is already connected to the vehicle, the information available can be sent to another device for further processing. Three simple possibilities are a PDA, a Pocket PC handheld, or a computer. In each case, appropriate software could be developed to allow the logging device to capture data, and output it to files or a graphical display.

2.2.4 Microprocessors and Sensors

One problem that the VEC needs to overcome is communication between each device. Two main solutions come to mind. One solution is to design a multi-drop network using EIA-232 for each device. The other solution is to use a microcontroller with the features needed.

First, an analysis of the system parts is in order. Each of the five sensors used require an input capture to determine the distance from the Pulse Width Modulated (PWM) signal returned by each sensor. A touch screen or PDA will be the main input device, as well as the output. This communication can be done with a microcontroller via a SPI system or EIA-232. Next is a voice synthesis chip which communicates over an SPI or EIA-232. Finally, the VEC needs to interface with the vehicle's OBD II system, whose output can be translated to EIA-232. The need is then for a microcontroller with three EIA-232 ports or two EIA-232 and one SPI subsystem, and five input captures.

Our team has a good understanding of the Motorola HC11 so that is our microcontroller of choice. However, the HC11 used in the Embedded Systems class has only one SCI port for interfacing to EIA-232, and the VEC needs two, one for the PDA and one for the OBD II connection. One solution to that problem is to use two HC11s, and connect the two together. While this could work, a cheaper and more elegant solution is to use a different MCU.

Motorola's HC11 family of microcontrollers has a few processors that fit our needs. The 68HC711S6 and 68HC711M0 both have 2 SPI subsystems and an EIA-232 port. Unfortunately, these microcontrollers both pose the same problem: lack of availability (Motorola does not really seem to admit to making these chips).

After investigating various PIC microcontrollers and other MCUs, the best solution found is the Rabbit 3000. This chip has six available serial ports, eight input capture channels, is faster than the HC11, and inexpensive. A development board, with the Rabbit 3000 [2], all the extra circuitry needed, and RAM cost only $39.00.

For a sensor, it is apparent that a long distance sensor is needed along with a wide temperature range. Having cost be the first concern, we will use Behrend's Polaroid 6500 [3] series ultra sonic range finders. These sensors offer a range of 0.5 feet to 35 feet, which is 25 more feet than we had hoped to be able to get. Unfortunately, the Polaroid 6500 sensors can only go down to 0°C.

After looking at modern systems on Ford, Mercury, and Toyota, another sensor was discovered. They were smaller, capable of being dropped to lower temperatures, and waterproof. Unfortunately, at a cost of $482.82 per sensor, they are unattainable for the project. This makes the Polaroid 6500 sensors the sensor of choice.

3. Engineering Requirements

3.1 Engineering-Marketing Relationships

The marketing and engineering requirements are intimately related. Although each marketing requirement will not affect every engineering requirement, there is often a positive or negative relationship among the two types. To determine the relationship, the marketing and engineering requirements were analyzed in relation to each other, and the effect of maximizing (or minimizing) one was evaluated versus the other. The results are summarized in the Table 2 on the next page.

A positive correlation is represented by a '↑', and a strong positive correlation is represented by '↑↑'. A negative correlation is represented by a '↓', with a strong negative correlation represented by '↓↓'. In certain rare cases, the correlation could be in both directions. For example, minimizing the number of buttons will maximize the simplicity of the interface (a '↑') but too few buttons would complicate the interface (a '↓'). In these cases, the correlation is represented by '↑↓'. If there was no correlation, the entry in the table is blank.

The '+' or '-' entries in the table indicate whether the requirement should be minimized or maximized. A '+' indicates maximize, and a '-' indicates minimize.

	Engineering Requirements		1 Sensor Range	2 Distance Reported Accuracy	3 Object Distance Sampling Rate	4 Time From Object Detection to Output	5 Reliability	6 System Power Consumption	7 Source Voltage	8 Waterproof	9 External Component Operating Temperature Range	10 Internal Component Operating Temperature Range	11 Vibration Tolerance	12 Shock Resistance	13 Target Sales Price	14 Side Mounted Sensor Size	15 Rear Mounted Sensor Size	16 Main Unit Size	17 Volume Range	18 Number of Buttons	19 Data Logging	20 External Component Weight	21 Main Unit Weight
	Marketing Specification		+	+	+	-	+	-	-	+	+	+	+	+	-	-	-	-	+	-	+	-	-
	Ease of Use:																						
1	Easy to Read Display	+													↓			↓↓					
2	Size	-								↓					↓	↑↑	↑↑	↑↑		↑		↑	↑
3	Securely Fastened	+					↑						↑↑	↑↑	↓	↓	↓	↓				↑	↑
4	Mute Ability	+						↑							↓				↑↑				
5	Simple Interface	+													↑↓			↑		↑↓			
6	Maintenance	-					↑			↑			↑↑	↑↑	↓								
7	Weight	-	↓							↓			↓	↓	↓	↑	↑	↑		↑		↑	↑
8	Number of Controls	-													↑↓			↑		↑↑			↑
	Aesthetics:																						
9	Size	-								↓					↓	↑↑	↑↑	↑↑		↑		↑	↑
10	Concealment	+													↓	↑↑	↑↑	↑↑	↓	↑			
	Durability:																						
11	Weatherproof	+								↑↑	↑↑	↑↑			↓	↓	↓						
12	Temperature Tolerance	+									↑↑	↑↑			↓	↓	↓	↓					
13	Shock Resistance	+											↑↑	↑↑	↓	↓	↓	↓				↑	
	Cost:																						
14	System Cost	-	↓	↓	↓	↓	↓	↓	↑	↓	↓	↓	↓	↓	↑↑	↓	↓	↓	↓	↑	↓		
	Effectiveness:																						
15	Audible Warning System	+				↑		↓							↓			↓	↑↑				
16	Visual Warning System	+				↑		↓							↓			↓					
17	Accuracy	+	↑↑	↑↑	↑	↑							↑	↑	↓						↑↑		
18	Reliability	+					↑↑																
19	Data Logging	+				↓									↓			↓			↑↑		

Table 2: Engineering-Marketing Relationships

3.2 Engineering Tradeoffs

As with the engineering and marketing requirements, there is a correlation between each of the engineering requirements and the rest of the engineering requirements. Often maximizing one requirement will affect another requirement – sometimes positively, sometimes negatively. To understand these tradeoffs, the requirements were set against each other in a matrix, and the effects of meeting one assessed against the others. This is shown in Table 3, on the next page.

Again, a ‘+’ indicates the requirement should be maximized, and a ‘-’ indicates that the requirement should be minimized. A ‘↑’ represents a positive correlation, and a ‘↓’ represents a negative correlation. No entry in the table indicates that there is no correlation.

			1 Sensor Range	2 Distance Reported Accuracy	3 Object Distance Sampling Rate	4 Time From Object Detection to Output	5 Reliability	6 System Current Requirement	7 Source Voltage	8 Waterproof	9 External Component Operating Temperature Range	10 Internal Component Operating Temperature Range	11 Vibration Tolerance	12 Shock Resistance	13 Target Sales Price	14 Side Mounted Sensor Size	15 Rear Mounted Sensor Size	16 Main Unit Size	17 Volume Range	18 Number of Buttons	19 Data Logging	20 External Component Weight	21 Main Unit Weight
			+	+	+	-	+	-	-	+	+	+	+	+	-	-	-	-	+	-	+	-	-
1	Sensor Range	+		↑											↓	↓	↓				↓	↓	
2	Distance Reported Accuracy	+													↓							↓	
3	Object Distance Sampling Rate	+						↓							↓							↓	↓
4	Time From Object Detection to Output	-													↓							↓	
5	Reliability	+													↓								
6	System Current Requirement	-													↓						↓		
7	Source Voltage	-													↑								
8	Waterproof	+													↓	↓	↓					↓	
9	External Component Operating Temperature Range	+													↓	↓	↓					↓	
10	Internal Component Operating Temperature Range	+													↓			↓					↓
11	Vibration Tolerance	+													↓							↓	↓
12	Shock Resistance	+													↓							↓	↓
13	Target Sales Price	-														↓	↓	↓	↓	↑	↓		
14	Side Mounted Sensor Size	-																				↑	
15	Rear Mounted Sensor Size	-																				↑	
16	Main Unit Size	-																		↑	↓		↑
17	Volume Range	+																					↓
18	Number of Buttons	-																					↓
19	Data Logging	+																					↓
20	External Component Weight	-																					
21	Main Unit Weight	-																					

Table 3: Engineering Tradeoffs

3.3 Marketing and Engineering Requirements

In order to begin the design process, the marketing needs (as determined by the Needs Assessment) need to be transformed into quantifiable engineering requirements. To do this, each marketing requirement was evaluated as to which engineering requirement would meet the need. Then, each engineering requirement was evaluated, and a specific value or range of values was determined as a target for each requirement. The resulting requirements, targets, and rationale are summarized in Table 4, below and on the next several pages. The marketing specification numbers correlate with the numbers in the Engineering-Marketing matrix, or Table 3.

Requirement Specification	Target	Marketing Specification(s)	Rationale
Sensor Accuracy	0.5ft to 10ft	17	• Closing objects need to be detected far enough away to stop in time • Objects need to be detected close to the vehicle • Sensors in this range are common [3]
Distance Reported Accuracy	< 1% error	17	• Needs to be small enough to prevent accidents from inaccurate measurements • Needs to be large enough to allow for MCU rounding errors • Sensors in this range are common
Object Distance Displayed Refresh Rate	2Hz to 10Hz	17	• Must update distance information displayed frequently enough to be accurate • Must be slow enough for calculations to be performed on sensor output

Table 4.1: Marketing and Engineering Requirement Relationships

Marketing and Engineering Requirements continued

Requirement Specification	Target	Marketing Specification(s)	Rationale
Time From Object Detection to Output	< 0.5 sec	15, 16, 17	• Must be fast enough for output to be useful • Must be slow enough for calculations to be performed on sensor output
Reliability	xx% operational after 5 years	6, 18	• System must be able to operate with a low failure rate to be useable by consumers, and to minimize warranty replacements • Actual percentage to be determined later
System Power Consumption	≤ 1A	18	• This is the most power that one voltage regulator can handle
Source Voltage	12Vdc	14	• Vehicle batteries are 12Vdc. • Using a different voltage would require additional voltage source(s) or additional circuitry, raising the system cost
Waterproof (Eternal Systems)	Submersible	6, 11	• System must be able to withstand rain or snow
External Component Operating Temperature Range	-20°F to 110°F	12	• The external components need to be able to withstand the average temperature ranges throughout the U.S.

Table 4.2: Marketing and Engineering Requirement Relationships

Marketing and Engineering Requirements continued

Requirement Specification	Target	Marketing Specification(s)	Rationale
Shock Resistance	3g	6, 13	• According to an expert, Mr. Englund at Penn State, Behrend, this is the normal operating range for a vehicle
Internal Component Operating Temperature Range	-20°F to 150°F	12	• The internal components need to be able to withstand the average temperature ranges throughout the U.S., plus the increased temperature in a vehicle due to the greenhouse effect
Vibration Tolerance	up to 30g at 100Hz	6, 13	• According to an expert, Mr. Englund at Penn State, Behrend, this is the normal operating range for a vehicle
Target Sales Price	< $500	14	• Comparable distance sensing systems with considerably less features cost up to $230 [4] • OBD II readers cost $150 to $1000 • One accident will cost more than $500
Side Mounted Sensor Size	< 6in x 9in x 3in (L x W x H)	2, 3, 7, 9, 10	• Small enough to be unobtrusive • Large enough to hold components, and allow for the fastening system
Rear Mounted Sensor Size	< 6in x 9in x 3in (L x W x H)	2, 3, 7, 9, 10	• Small enough to be unobtrusive • Large enough to hold components
Main Unit Size	Between 2in x 3in x 2in and 6in x 9in x 3in (L x W x H)	1, 2, 3, 7, 9, 10	• Small enough to be unobtrusive • Large enough to hold components • Large enough for an LCD screen that is big enough to read

Table 4.3: Marketing and Engineering Requirement Relationships

Marketing and Engineering Requirements continued

Requirement Specification	**Target**	**Marketing Specification(s)**	**Rationale**
Volume Range	silent to 60dB	4, 15	• Must be mutable • Allows for different volume ranges for different levels of warning • Not loud enough to damage ears
Number of Buttons	< 5 buttons	2, 5, 8, 9	• More than 5 would be too complicated for a driver to use while driving • Allows the main unit size to remain small
Data Logging	≥ 10 previous measurements	17, 19	• Enough data for the unit to operate properly • Small enough to allow a small RAM to hold data, thus minimizing space
External Component Weight	< 1lb	3, 7	• Light enough to be securely fastened
Main Unit Weight	< 10lb	3, 7	• Light enough to be securely fastened

Table 4.4: Marketing and Engineering Requirement Relationships

3.4 Constraints

There are several standard constraint categories that all projects must address during development. These include, but are not limited to, economic, environmental, ethical, and health constraints. In this section we will examine how several of these constraints relate to the VEC.

3.4.1 Economic

The VEC system should operate in an economically profitable environment. The device must be able to be manufactured and produced at a minimal price, enabling the average driver to purchase the product. The income of the most elderly citizens is such that they are not necessarily able to spend several hundred dollars on a car safety system. Since a somewhat large share of our target audience is elderly, this is a determining factor.

At the same time, the installation of a VEC system could save the user hundreds, if not thousands of dollars in accident repair costs. Marketing from this angle would allow the base cost to raise higher than if it the marketing of the system was merely safety oriented.

3.4.2 Environmental

Various components of the VEC will operate in extreme operating temperatures. A locked car during a hot sunny day can reach temperatures over 120 degrees Fahrenheit quite quickly. For this reason, the display and controller among other interior parts of the system must withstand high heat. Furthermore, the interior parts must be able to operate in the cold winter months of the coldest regions, so the internal parts of the VEC must also be safely operational at a temperature of -40 degrees Fahrenheit.

The external parts of the system will see extreme heat temperatures in the warmer months of the year. For this reason, the VEC external sensors must operate at temperatures of up to 120 degrees Fahrenheit. Furthermore, the external sensors will also be subject to extreme cold temperatures during the winter months. The target temperature for safe operations is -40 degrees Fahrenheit.

Another aspect of the operational environment of the VEC is vibrations. Vehicles are subject to shocks and vibrations based upon pavement types, road conditions, and vehicle conditions. The main concern involving shock and vibration for the VEC lies in the anchoring system used for both the external sensors and the inside display/controls. The VEC must be able to withstand the day to day vibrations, as well as the crater-like potholes common to Pennsylvania.

Finally, electromagnetic interference from other car systems must be considered when routing data and power lines for the VEC system. The VEC's data and power lines will be separated as much as possible, much like what is done in the car stereo industry.

3.4.3 Ethical

Several aspects come into play for this category. First, several systems for blind spot detection and reverse assistance already exist in the market. For this reason, we must ensure that we do not infringe on any patents or copyrights. Second, there are many standards that have been developed regarding the wiring of devices in and on automotives. In order to obtain UL approval and have the product considered a safe addition to the customer's auto, we must adhere to all such specifications. Finally, we must be sure that in no way do we impair a vehicle's current operation. The installation of the VEC system must not decrease the functionality in any system already present in a vehicle.

3.4.4 Health and Safety

The VEC system installed on a vehicle should in no way pose shock hazards to individuals in or around the car. Once again, UL approval for the system is necessary. Furthermore, the system must be accurate enough that it is possible to use safely; a bad sensor or calculation could prove fatal if the user believes it true.

3.4.5 Manufacturability

In order to make the VEC system a feasible product, several physical characteristics must be defined. First, the VEC must interface with the standard OBD II interface of all vehicles 1996-present. Second, the size of the unit should not interfere with user operation of controls found already in the vehicle. The VEC's installation system must include several solutions to mounting the display device in the interior of the car. Third, the software for the VEC should be coded in accordance to the chosen processors language.

3.4.6 Political

Since the VEC is going to be designed for use on vehicles in the U.S., the U.S. Department of Transportation should be consulted for applicable rules, standards, and permission needed.

3.4.7 Social

The successful implementation of VEC systems in many automobiles across the country could lead to much safer highways. Safer highways would translate into many cost reductions in other areas, such as healthcare, auto insurance, and even auto repair. Necessary to the success of the VEC system is a simple interface that will allow targeted audiences to easily use the product. It's important to remember that not all users will be computer literate.

3.4.8 Sustainability

Ideally, the VEC system will be a one time purchase per car, possibly even transferable to other vehicles. The lifetimes of the electronic components in the system are all longer than the average lifetime of a car. If we program in a standard language such as C, and meet all the standards, upgrading the system should be possible.

3.5 Standards

When developing a device for use in a vehicle in the U.S., there are many standards to be considered. The standards come from the Electronic Industries Alliance (EIA), the International Organization for Standardization (ISO), the National Fire Protection Association (NFSA), the Society of Automotive Engineers (SAE), Underwriters Laboratories (UL), and the American National Standards Institute. The relevant standards are listed on the next page.

EIA:

- EIA-232 – Standard for Serial Data Communication

ISO:

- ISO 7722 – Vehicle Wiring
- ISO 9141 – OBD II Communication Protocols

NFPA:

- NFPA70 – The National Electric Code

SAE:

- J163 – Low Tension Wiring and Cable Terminals and Splice Clips
- J561 – Eyelet and Spade Wire Terminals
- J858 – Blade Type Wire Terminals
- J928 – Pin and Receptacle Wire Terminals
- J1850 – OBD II Communication Protocols
- J1879 – General Qualifications and Production Acceptance Criteria for Integrated Circuits in Automotive Applications
- J1939 – Standards Collection for Serial Communication in a Vehicle
- J1962 – Standard OBD II Connector
- J1979 – Standard OBD II Commands and Timing

UL:

- UL50 – Enclosures for Electrical Equipment
- UL 776 – Standards for Printed Wiring Boards
- UL 2086 – Vehicle Battery Adapters

ANSI:

- C/C++ - The standard for the C/C++ programming language

These standards can be obtained from the SAE, ISO, UL, NFPA, EIA, and ANSI for a fee. The web site for each is listed in the references.

4. Design

4.1 High Level Design

The VEC can be broken down into the following high level concept in simplest form. The Rabbit 3000 MCU will control and act as the coordinator between all other components of the system. To better understand the relationship between these components we must examine Figure 3, below, which shows a physical layout of the system and the orientation of its devices.

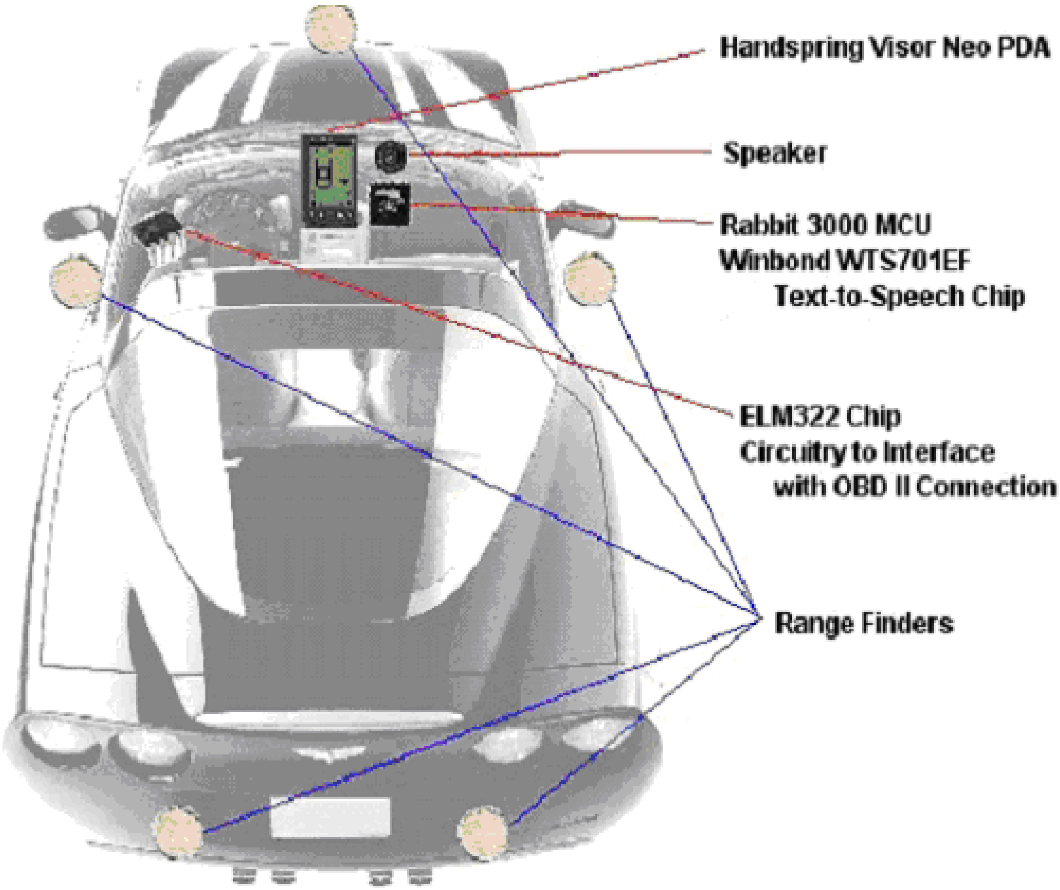


Figure 3: High Level Design Hierarchy

4.2 System Organization

4.2.1 Hardware Organization

A Rabbit 3000 microprocessor serves as the center of the VEC hardware system. This processor receives input from a set of ultrasonic sensors, the OBD II system of the vehicle, and user input that is received via the PDA. It also directs the appropriate output to a Winbond WTS701EF [5] text to speech chip and the PDA's display. Figure 4 shows a diagram of the system organization.

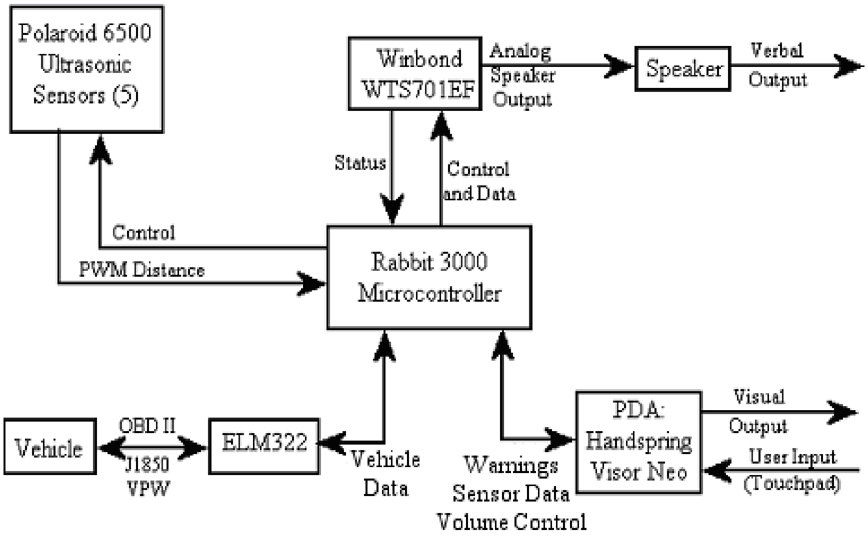


Figure 4: System Organization

The ultrasonic sensor signals are the pulse width modulated representation of the distance to any detected object, which is computed using circuitry included in the sensor's package. There are five ultra-sonic sensors surrounding the vehicle; two on the back, one on each side, and one in the front. The ultrasonic sensors send distance information to the Rabbit 3000, and constantly monitor object distances as the vehicle is moving down the road.

Communication with the Winbond WS701EF/T is accomplished by sending ASCII characters to it, along with a series of commands to set up its operations. Status in the form of digital signals is also sent back to the Rabbit 3000 from the Winbond WS701EF/T. The Winbond WS701EF/T outputs the voice signal output directly to the speaker, without need for further speaker driving circuitry. Digital signals to and from the PDA are sent via a serial communications interface (SCI), using the EIA-232 standard [6] and a Max3323 [7] chip for translation.

Finally, the OBD II system sends the Rabbit 3000 information from the vehicle's computer, again via EIA-232 and a Max3323. The OBD II communication protocol is translated to EIA-232 via an ELM322 [8]. The power to the system comes from the vehicle battery through the OBD II system, which is described in detail later.

Power to the entire system, aside from the PDA, is provided via the OBD II's 12Vdc output. This will be stepped down to 5Vdc for the ELM322 and sensors, and 3Vdc for the remaining components. The PDA is powered by two 1.5 volt AAA batteries.

4.2.2 Software Organization

Software for the Rabbit 3000 consists of several main sections including the Rabbit 3000 to PDA interface code, sensor code, Winbond interface functions, and a data section that handles information obtained from the OBD II system. All of the components are controlled with a multitasking main function. The sensors are polled at a set interval by use of delay functions. The communication between the ELM322 and the Rabbit 3000 and between the PDA and the Rabbit 3000 are handled by calls to functions that allow serial communication to occur while the Rabbit 3000 handles other tasks. The calculations on the OBD II data and the update of the display are performed in the main structure of the program.

The PDA software recognizes input from a touch screen interface on the PDA. This input is used to give commands to the Rabbit 3000, configure the system, and access any available OBD II data or logged data. The software also draws the vehicle on the PDA screen, and shows the distance and direction to any object detected. Finally, the software handles the communication through the serial port of the PDA.

Whenever the Rabbit 3000 requests it, the OBD II system sends the speed of the vehicle through the ELM chip to the Rabbit 3000. This is used to determine the severity of any warning.

For instance, if the vehicle is traveling 5 mph, an object 20 feet in front would be less of a concern than if the vehicle is traveling 60 mph. This is due to the fact that the time the driver has to react is inversely proportional to the vehicle's speed.

The Rabbit 3000 uses the speed of the vehicle in conjunction with the distance information to determine whether an object is too close to the vehicle or approaching too quickly. If either of these is true, the Rabbit 3000 sends the appropriate information to the Winbond chip, which outputs a verbal warning to the user through the speaker. The Rabbit 3000 also sends information to the PDA to display a visual warning to the user.

The Rabbit 3000 requests trouble codes from the OBD II system through the ELM, so the VEC system can also alert the user if something goes wrong with the vehicle. This happens approximately once per second. The VEC interprets the trouble codes via a look-up table, and displays a message on the PDA to alert the user of the error. The error codes may also be accessed manually, and may be cleared from the OBD II system by the user.

The user can interact with the system at any time via the PDA. The user has the ability to change the volume level of the audio alerts, browse logged data such as past speeds and previous distances, and see the OBD II's error codes and performance data.

4.3 Component Level Design

The following is a functional description of each component from Figure 4.

4.3.1 Rabbit 3000

The microprocessor used is the Rabbit 3000 from Rabbit Semiconductor. The inputs and the outputs of the Rabbit 3000 are shown on the next page.

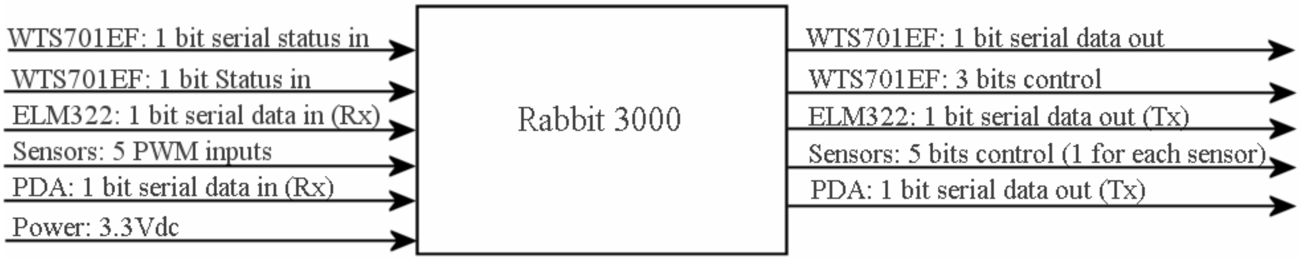


Figure 5: Rabbit 3000 I/O

Inputs:

- WTS701EF: serial status in: This is a serial data line that communicates status from the WTS701EF to the MCU. The data transmitted is the status of the WTS701EF, and is the serial data input of the SCI interface.
- WTS701EF: status: This one bit input is a busy signal from the WTS701EF. If 3V, or high, the WTS701EF chip is not busy. If 0V, or low, the WTS701EF is busy.
- ELM322: serial data in: The serial data input is, in the EIA-232 protocol terminology, the Rx signal. It is the data being sent from the OBD II system, through the ELM322.
- Sensors: PWM inputs: This is a digital signal, and represents the distance to any object in the range of the sensors.
- PDA: serial data in: This is the serial data input from the PDA. This is the second Rx input to the Rabbit 3000, and is the control and status data from the PDA to the rest of the system.
- Power: 3.3Vdc

Outputs:

- WTS701EF: Serial Data Output: This is control and data sent to the WTS701EF, to tell it what to output and when. This is the serial data output of a Serial Peripheral Interface (SPI).
- WTS701EF: Control: This is the three main control bits for the SCI interface between the WTS701EF and the Rabbit 3000. The three bits are: Slave Select, sent low to indicate that the Rabbit 3000 is transmitting; INT, sent low to indicate that the Rabbit 3000 wants to interrupt the WTS701EF's operation; and SCLK, the clock for the SPI connection.
- ELM322: Serial Data Output: This is the serial data output (Tx) from the Rabbit 3000 to the ELM322. This transmits both control and data.
- PDA/Computer: Serial Data Output: This is the second Tx line – the serial data output from the Rabbit 3000 to the PDA. This transmits data to the PDA for the user display.

Functionality:

The Rabbit 3000 in the VEC is responsible for:

- Receiving the sensors' input, and calculating from this the distance to any object.
- Asking for and receiving from the OBD II system the speed of the vehicle, through the ELM322.
- Calculating, from the sensor data and vehicle speed, what, if any, alert should be issued to the user.
- Determining, from the PDA input and the speed of the vehicle, which mode to operate in.
- Outputting to the PDA the data needed to indicate to the user any objects that may be present.
- Outputting to the WTS701EF chip the control and data information needed to output any pertinent alerts to the user.
- Output to the PDA any trouble codes present in the vehicle.
- Output to the PDA any stored trouble codes or diagnostic information from the OBD II system upon request from the PDA.
- Clear trouble codes in response to PDA input.

4.3.2 ELM322

The inputs and the outputs of the ELM322 are shown in Figure 6, below.

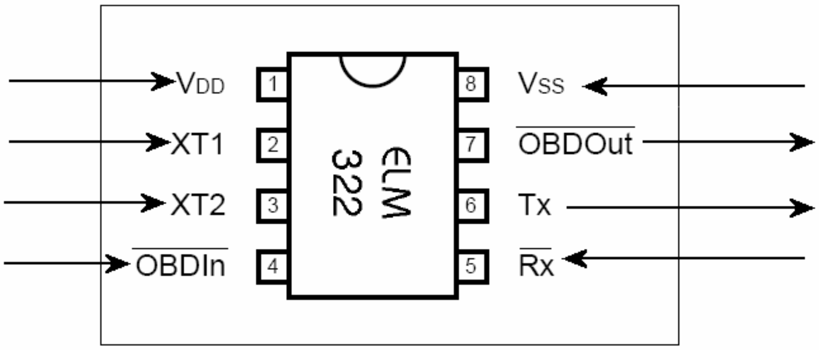


Figure 6: ELM322 I/O [8]

Inputs:

- VDD: The positive voltage input, 3Vdc.
- VSS: The ground reference input.
- X1 and X2: These are the inputs for the crystal oscillator, acting as a clock. This must be a 3.58 MHz oscillator.
- OBDIn: This data is in the format of the J1850 VPW protocol, and is the data output from the OBD II system, to be sent to the Rabbit 3000.
- Rx: The serial data input from the Rabbit 3000. This is in the EIA-232 protocol, and is the configuration and status request data from the Rabbit 3000 for the ELM322 and the OBD II system, and then request for data from the Rabbit 3000 to the OBD II system.

Outputs:

- Tx: The serial data output to the Rabbit 3000. This is the data that is sent from the OBD II system, as translated to the EIA-232 format. Also, this can be status information from the ELM322, in response to the Rabbit 3000.
- OBDOut: This data is in the format of the J1850 VPW protocol, and is the data input to the OBD II system, sent from the Rabbit 3000, and translated by the ELM322.

Functionality:

The ELM322's sole purpose is to translate commands from the Rabbit 3000 into a format that the OBD II system can understand, and translate data from the OBD II system into a format that the Rabbit 3000 can understand. The two formats are J1850 VPW and EIA-232. The ELM322 receives commands from the Rabbit 3000 to configure it, then receives data for the OBD II system, which it then translates. The data from the OBD II system is in response to the Rabbit 3000's commands, and is translated again by the ELM322.

There are some additional components required to make the ELM322 function properly. This includes resistors, the crystal oscillator, capacitors, diodes, and transistors. A detailed schematic is included in the data sheet provided by ELM [8]. A slightly modified version is shown in Figure 7, on the next page.

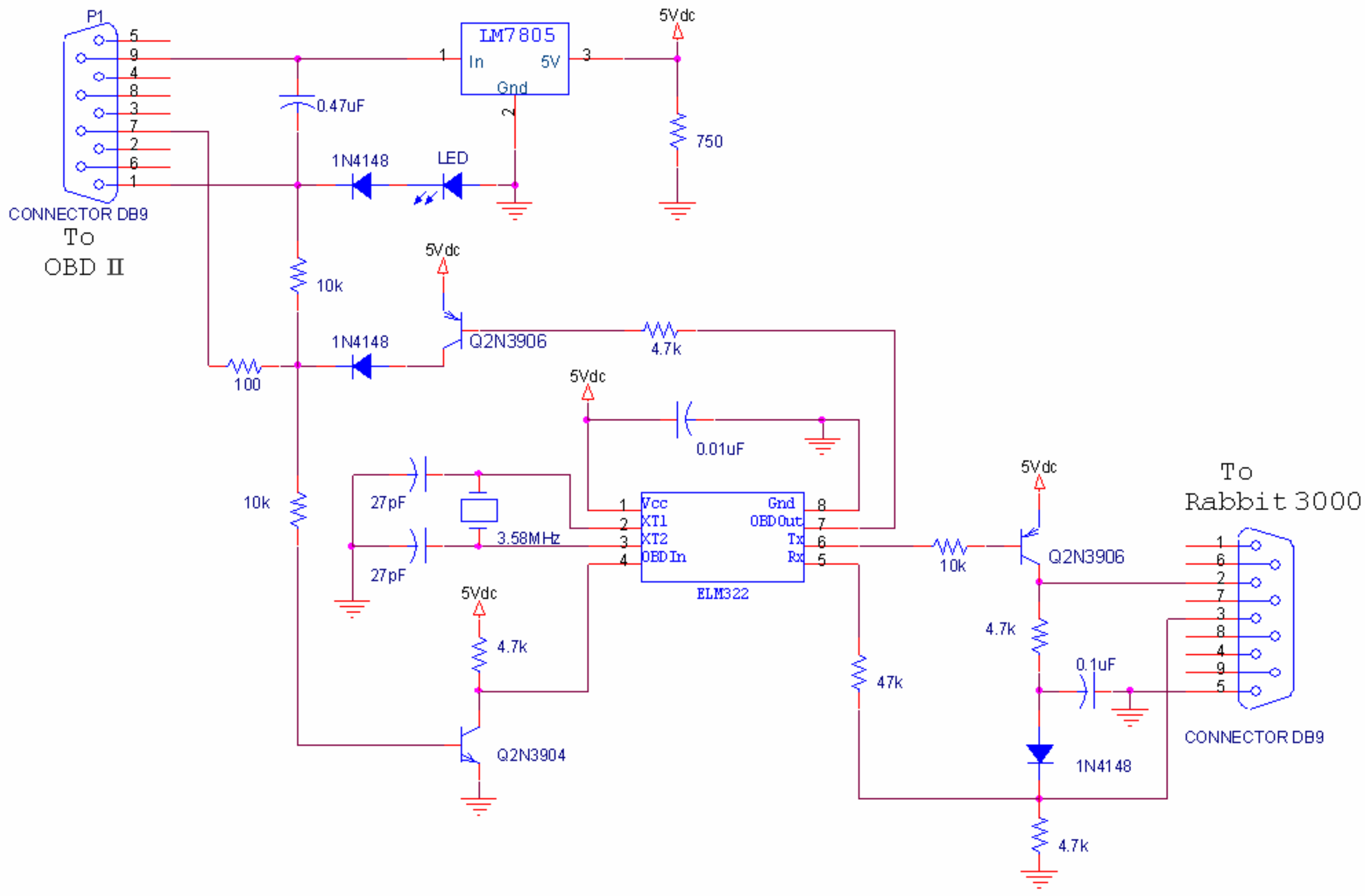


Figure 7: OBD II to Rabbit 3000 interface, with ELM322 [8]

In the Figure 7, the DB9 connector to the OBD II system connects the ELM322 to the vehicle ground on pin 1, the vehicle battery positive on pin 9, and the bus+ connection on pin 7. The bus+ connection is a bi-directional data line, whose format is the J1850 VPW protocol. The DB9 connection is converted to a J1962 connection via a cable available from B and B Electronics [9].

The DB9 connector to the Rabbit 3000 is a standard 9 pin EIA-232 connector. The Tx output is pin 2 and the Rx input is pin 3. These are the same as defined in the Inputs and Outputs sections in the functional description on the previous page. The common ground, provided by pin 5, is needed between the two sides of an EIA-232 connection to ensure that the positive and negative voltages are interpreted correctly.

The standard J1962 connector is shown in Figure 8 below. The pins used with the J1850 protocol are vehicle ground on pin 5, vehicle battery positive on pin 16, and bus+ on pin 2.

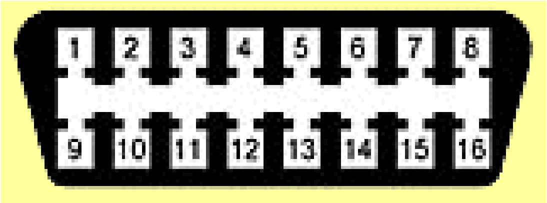


Figure 8: Standard J1962 OBD II connector – female end (on the vehicle) [10]

4.3.3 OBD II

The inputs and the outputs of the OBD II system are shown in Figure 9, below.

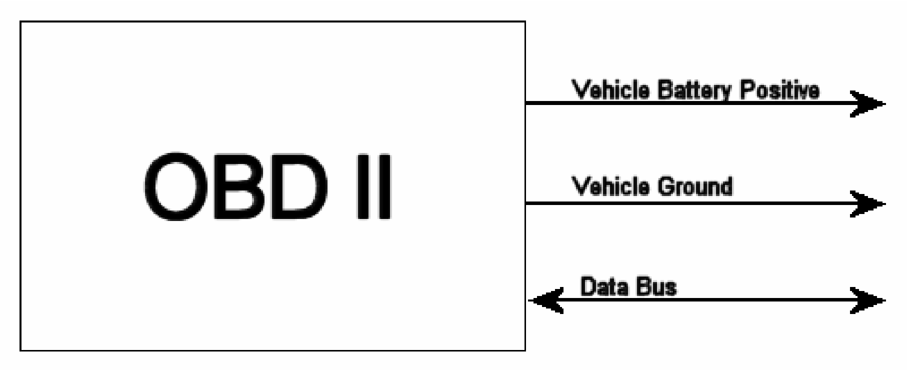


Figure 9: OBD II I/O

Inputs:

- Vehicle Battery Positive: This is a direct connection from the positive battery terminal of the vehicle. This is 12Vdc, and will be used to provide the power to the VEC.
- Vehicle Ground: This is a direct connection from the negative battery terminal of the vehicle, and provides a ground reference for the rest of the system.

Bidirectional:

- Data Bus: This is a bi-directional line, and is in the format of the J1850 VPW protocol. The input is the commands and configuration for the OBD II system, sent by the Rabbit 3000 and translated by the ELM322. The output is the status and data information requested by the Rabbit 3000.

Functionality:

The OBD II system is a connection to a vehicle's computer. All cars and light trucks built and sold in the US after January 1, 1996 have an OBD II system installed, as mandated by the EPA. This system monitors the many and varied sensors in place in the vehicle, and can, when requested, output useful data. For the VEC, the primary concerns are the vehicle speed and any current trouble codes. When the OBD II system receives the command "68 6A F1 01 0D", it will respond with a 6 hexadecimal digit value, of which the last two digits represent the speed in kph. When the OBD II system receives the command "68 6A F1 01 01", it will respond with the number of trouble codes currently stored.

In order to determine the meaning behind some of the DTCs, the VIN of the vehicle must also be known. This requires three commands, as each only gets one third of the DTC. The commands are "6C 10 F1 03 01", "6C 10 F1 03 02", and "6C 10 F1 03 03", for the first 5 digits, the middle 6 digits, and the last 6 digits, respectively.

4.3.4 Power Supply

The inputs and outputs of the power supply are shown in Figure 10, below.

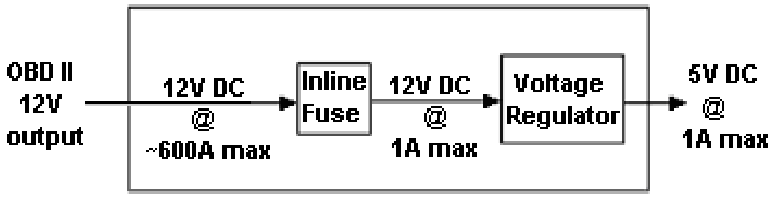


Figure 10: Power Supply Design

Sub-Module:

Inline Fuse (1A)

Inputs:

12V DC @ approximately 600A max

Outputs:

12V DC @ 1A max

Functionality:

Limits the maximum current draw to 1A, acting as a safety device for the system.

Sub-Module:

Voltage Regulator

Inputs:

12V DC @ 1A max

Outputs:

5V DC @ 1A max

Functionality:

Regulates the voltage from the car battery to 5 volts to provide 5V DC to the system.

Module:

Power Supply

Inputs:

12V output from OBD II

Outputs:

5V DC signal @ 1A max

Functionality:

Converts the 12V DC @ ~600A maximum signal from the OBD II output to a 1 amp 5 volt DC supply that can power the rest of the system.

4.3.5 PDA

The inputs and outputs of the PDA are shown in Figure 11, below.

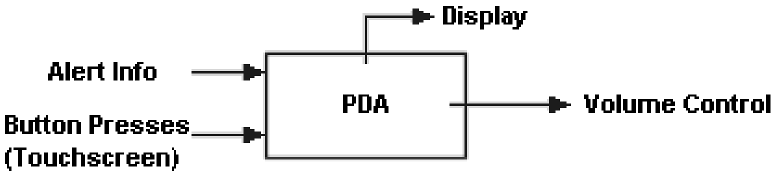


Figure 11: PDA I/O

Module:

PDA

Inputs:

User button presses via touch screen, alert info from the Rabbit 3000.

Outputs:

Visual display and volume control information.

Functionality:

Allows the user to interact with the VEC system. Displays visual alert if object(s) are too close to the vehicle. Allows the user to change the volume of audio alerts. Allows user to browse logged data (past object interferences, OBD II error codes).

4.3.6 Winbond WTS701EF

The inputs and outputs of the Winbond WS701EF/T are shown in Figure 12, below.

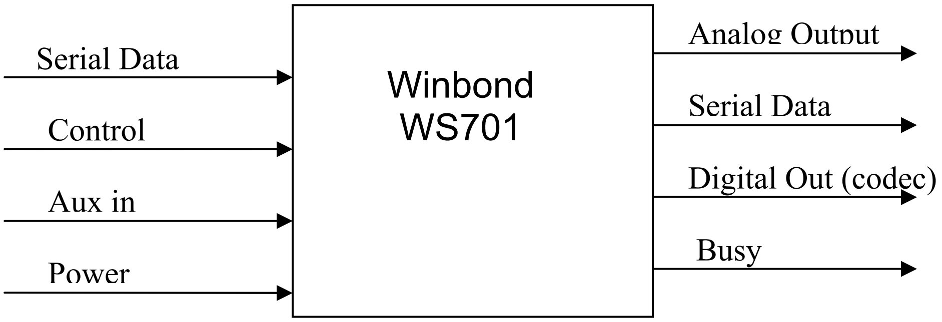


Figure 12: Winbond I/O

Input:

- Serial data: ASCII input through SPI port from microcontroller
- Control: slave select, chip select, SPI clock, and interrupt request, all from the MCU
- 3.3Vdc power
- Aux in: an analog or digital input that can be passed directly to the analog or digital output

Output:

- Analog signal to drive a speaker
- Digital output using a codec
- Serial data: status information to MCU
- Busy: a one bit busy indicator – active low

.

Functionality:

The Winbond WS701EF/T will perform the following tasks for the VEC:

- Translate ASCII messages to audio alerts
- Drive the speaker with its built-in speaker driver.

4.3.7 Speaker

The inputs and outputs of the speaker are shown in Figure 13, below.

Figure 13: Speaker I/O

Inputs:

Analog signal from the Winbond WS701EF/T which features a speaker driver built in

Outputs:

Audio noise in the form of alerts and other text messages the user should hear

Functionality:

The speaker will output all forms of audio from the VEC system. This is possible because the Winbond WS701EF/T has an auxiliary in pin which is passed directly through the speaker driver.

4.3.8 Polaroid 6500 Range Finders

The inputs and outputs of the Polaroid 6500 Range Finders are shown in Figure 14, below.

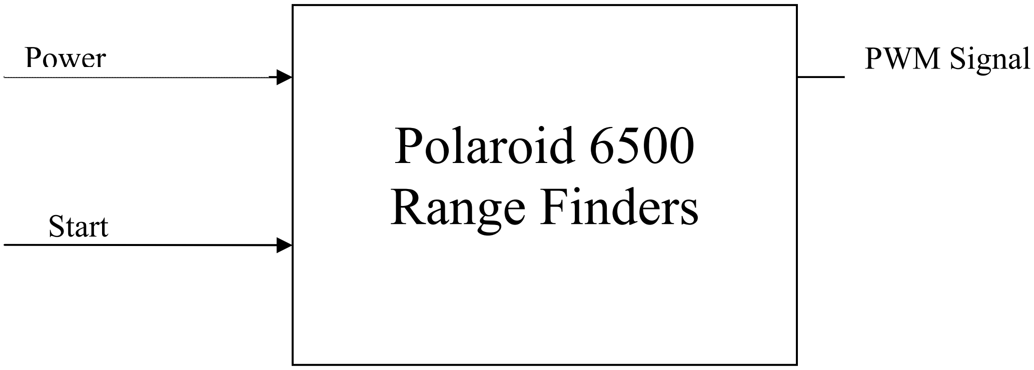


Figure 14: Polaroid 6500 Range Finder I/O

Inputs:

5Vdc from power supply

Start command (tells sensor to send out one chirp)

Outputs:

Pulse width modulated (PWM) signal representing the time for an ultrasonic pulse to be returned.

Functionality:

Giving the sensor power and the appropriate time to "boot up", the microcontroller will then initiate a chirp with the start line. The Polaroid 6500 will then send out about a 50 kHz sound that will be reflected off of objects and then received by the sensor. A timer will watch for the sensor to transmit a transition from logic low to logic high. The time represents the time from the sensor to the object and back again.

4.3.9 Rabbit 3000 Algorithm:

The main algorithm for the Rabbit 3000 is essentially a large loop, running several sections inside the loop simultaneously. The Rabbit 3000 is programmed using Dynamic C [6], which is provided by Rabbit Semiconductor. One of the great features of the Dynamic C is multitasking through costates. A costate is a section of code that acts like an endless (while(1)) loop in that when the end is reached, the program returns to the beginning of the costate. The most important difference is that the costate can be left, another costate run, and eventually return to the original costate.

If several costates are placed inside an endless loop, the program will run the first costate until told to exit that costate, then run the next until told to exit, and so on until the bottom of the loop is reached. The program will then return to the top of the loop, and run the first costate it reaches that is ready to run again.

In the VEC code, there are two ways that a costate is left. The first is to call a delay function. For example, by calling "waitfor(DelayMs(250))", the program is told to leave the costate and not return for 250 milliseconds. This allows time for the other costates to run, and also easily allows the creation of a 4Hz cycle with the ultrasonic sensors.

The second method used to leave a costate is to call a serial communication function. Dynamic C has functions to send and receive characters from the serial ports of the Rabbit 3000 that instruct the program to try to send or receive, and not return to the costate until the communication is done. Since serial communication is done with buffers, the program can even send and receive while a second costate is actively running.

The different costates for each component connected to the Rabbit 3000 are described below.

4.3.9.1 Rabbit 3000/ELM322 Costate:

All communication between the ELM322 and the Rabbit 3000 is done through one of the serial ports on the Rabbit 3000, via the EIA-232 protocol. Since the Rabbit 3000 serial ports are set up to communicate this way by default, the only command needed to start is to open the port, and tell the Rabbit 3000 to operate that port at 9600 baud. This is all done with one command, prior to entering the ELM322 costate.

The ELM322 costate begins by initializing the ELM322. By default, the ELM322 is set up for communication to a software terminal emulator, like HyperTerminal, and as such has extra formatting to translate the OBD II data to ASCII text and extra line feeds. The initialization disables these, allowing for faster communication. The costate then retrieves the VIN from the vehicle, and stores this in an array to be output to the PDA.

After this, the costate enters a while(1) loop, retrieving the speed of the vehicle and then number of DTCs every time. The costate also instructs the ELM322 to get any stored DTCs and/or clear the DTCs, but only if the PDA has requested one or both.

Throughout the communication process, the costate repeatedly sets a sentinel variable to 0. This variable is set to 1 by the PDA costate on every run through its loop. This enables the

PDA to know if the ELM322 quits responding – if the variable stays 1, the ELM322 is malfunctioning.

Another check is done during each communication. If the Rabbit 3000 ever gets the wrong input from the ELM322, again indicating an error, the costate instructs the ELM322 to reset by sending it the command "ATZ", then returns to the top of the costate to reinitialize the ELM322. The Rabbit 3000 also tracks the number of times this error occurs, and if it happens more than 10 times, the Rabbit informs the PDA that the ELM322 is malfunctioning by setting the previously mentioned sentinel variable to 1. The Rabbit 3000 then quits trying to communicate with the ELM322 by waiting forever for a character from the ELM322 that will never come since the Rabbit 3000 did not ask for anything.

4.3.9.2 Rabbit/Sensor Costate:

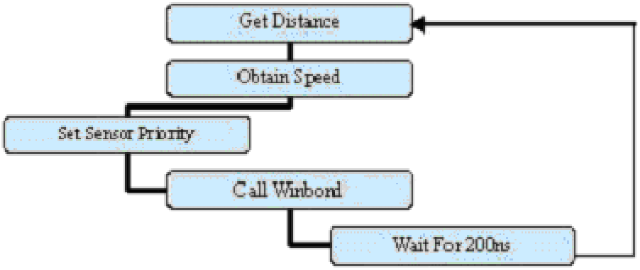


Figure 15: Rabbit/Sensor Algorithm Basic Layout

The sensor costate starts off by sending INIT line to logic high (5V). When this occurs the sensors output a 50 kHz sound wave. The rabbit "watches" for all five sensors to send their ECHO line to logic level high. The rabbit then takes this difference in time and multiplies it by 343.2m/s, or the speed of sound. This distance is then divided in half so only the distance to the object is used in other computations. The sensor costate then obtains the speed from a global variable supplied from the ELM322. The speed is then broken down into three modes: parking mode (0 – 5 mph), urban mode (5 - 45mph), autobahn mode (above 45mph). In all three driving modes the sensors costate tells the Winbond to output a verbal warning. While in parking mode, all Winbond calls are followed by distance conversions from meters to feet and saved for the Winbond costate to output.

Within each mode, three warning levels determine the urgency of the detection. These three modes are; zero = no warning, one = minor warning, and two = major warning. No warning indicates that no object is detected. A minor warning indicates that an object is detected, but not very close. A major warning indicates that an object is very close to the vehicle. For instance, a vehicle traveling 3 mph is in "parking mode," and an object less than three feet away is "very close." However, a vehicle traveling over 45 mph is in "Autobahn mode." In that mode, an object less than 18 feet from the front or rear is very close, while on the sides, very close is less than 3 feet.

4.3.9.3 PDA/Sensor Costate:

The only device the PDA communicates with in the system is the Rabbit 3000. To communicate, the PDA sends the Rabbit 3000 one byte of data to inform the Rabbit that it is ready to receive data. This byte is for synchronization, and it also contains information on volume control. When the Rabbit 3000 receives the sync byte, it sends 64 bytes to the PDA. The byte ordering and information is shown in Table 5 below.

Byte #	Information
1	Vehicle Speed
2	Number of DTCs from OBD II
3	Left Rear Sensor Information
4	Right Rear Sensor Information
5	Left Side Sensor Information
6	Right Side Sensor Information
7	Front Sensor Information
8	ELM Error Information
9	Winbond Error Information
10	Sensor Error Information
11-46	DTC codes, if reported
47-63	Vehicle Identification Number
64	NULL

Table 5: Data Sent from Rabbit 3000 to PDA

4.3.10 PDA Algorithm

After receiving this data, the PDA first updates the display of the vehicle's speed and the number of DTCs. If the number of DTCs is one or more, the user may click on the 'Show DTCs' button, in which case the PDA uses bytes 11 through 46 to determine the specific DTC and displays the error(s) that the OBD II connection is reporting. The Vehicle Identification Number (VIN), bytes 47 through 63, is also necessary to determine specific errors as some of the errors are manufacturer specific. These errors are all stored in a look up table (LUT) in the PDA so the entire error(s) would not have to be sent each time. We opted to use a LUT rather than transmit the entire error when it occurred because the use of a LUT reduces the time the serial port on the PDA must remain open, saving battery power, and reduces the time the Rabbit 3000 takes to send data, allowing the Rabbit to process other tasks.

The PDA then checks the sensor information from each of the five sensors. This data will contain a zero, one, or two. If it is zero, it means there is no warning. If it is one, there is a warning and the section of the car which something is too close begins to blink on screen. If it is two, there is an urgent warning, and the corresponding section of the car blinks faster than for a normal warning. We chose to perform the calculations as to whether or not an object is too close and hence a warning should be given to the user on the Rabbit 3000 rather than on the PDA. The main reasoning for this is because the Rabbit 3000 also has to tell the Winbond chip that a warning needs to be given. If the PDA were to perform these calculations, it would have to pass the data to the Rabbit 3000, and then to the Winbond chip. The extra time this would take would delay the warning.

Next, the PDA checks bytes 8, 9, and 10. If any of these bytes are not zero, that means an error has occurred in the respective subsystem of the VEC system. A warning is shown to inform the user what part of the system is not working correctly.

The basic structure of the PDA software is shown in Figure 16, on the next page.

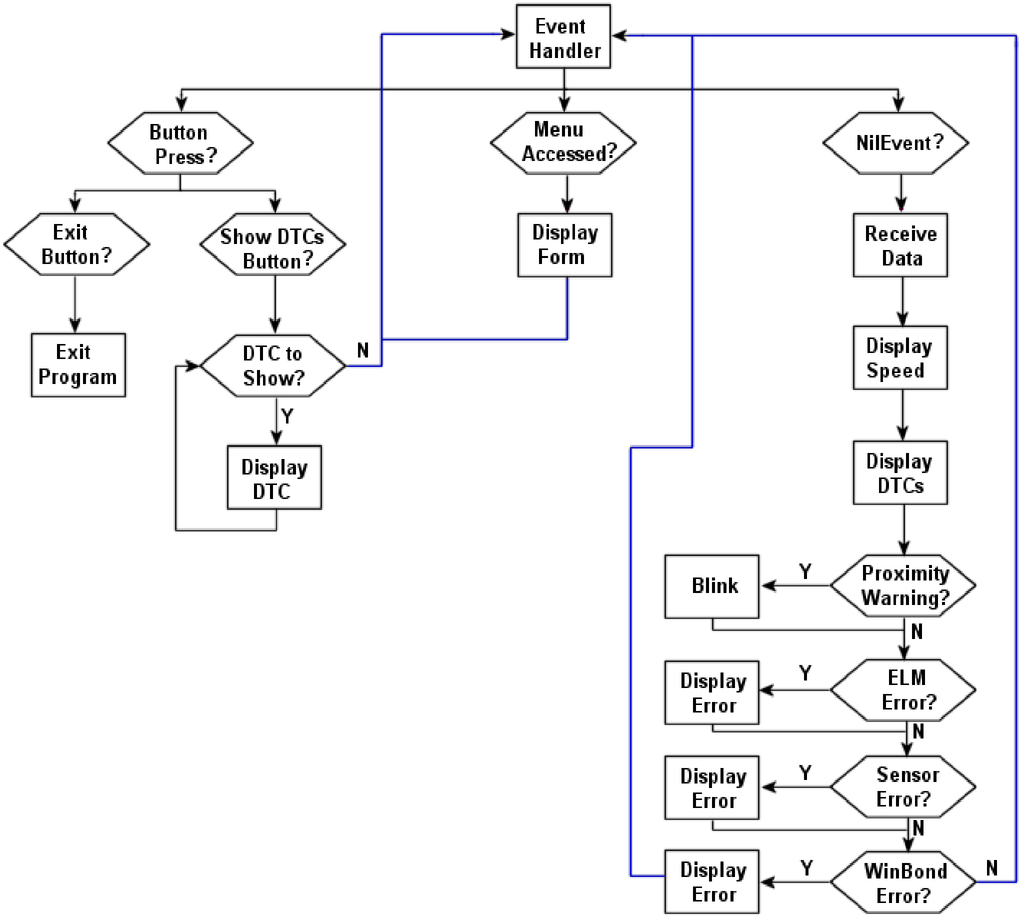


Figure 16: PDA Software Structure

4.4 Design Decisions

Table 6, on the next age, shows the different choices available for each component of the design. The explanations on the following pages describe how we arrived at the decisions of what components will make up the VEC system.

User Interface	Display	Power	Internal Size	External Size
Buttons	CRT	Car Battery	Shoe Box	Quarter
Touch Pad	LCD	Solar	Dash Mount	Shoebox
Voice Recognition	LED	Fuel Cell	Notebook	Inside Bumper
Keyboard	Touch Screen	9V Batteries	Glove Compartment	Side view Mirror
10 digit pad	None	Thermal Transfer	Stereo Mount	
Mouse	HUD	Cigarette Lighter		
		Fuse Panel		

Microcontroller	Alerts	Sound Output	Connectivity	Sensors
MCU	Car Speaker	Own speaker	OBD II	Laser
PDA	Own Speakers	Car Speakers	RS232	Infrared
Notebook computer	Flashing Lights	Buzzer	USB	Ultrasonic
Digital System	LEDs	Car Horn	Wireless	Physical
	Shock	Bull Horn	Wired	Camera
	LCD output	Cell Phone	Modem	Car Computer
	Vibration		Telephone	
			OneWire	

Table 6: Components Brainstorming

4.4.1 Connectivity

There are several connection issues in the VEC design. The first deals with communication between the VEC and the vehicle. The VEC needs to know the speed of the vehicle in order calculate how fast an object is closing, and to determine which mode to operate in. One mode, Highway Mode (HM), will show the user less detailed information about objects behind and in front of the vehicle, in order to show any objects in blind spots on the sides of the vehicle. The other mode is Reverse/Parking Mode (RM). In this mode, the VEC will show more detail behind the vehicle, and more detail to the front and rear.

While the user can always use one of the buttons on the PDA to set the mode, the VEC will attempt to determine the mode on its own by using the vehicle speed. If the vehicle is traveling less than 10 mph, the VEC will assume RM, since few people reverse or park at speeds greater than this. Otherwise, the VEC will assume HM.

The simplest way to get the vehicle speed is through the OBD II system. Other options considered were mounting sensors and doing calculations, or hacking into the vehicle's speedometer. Since the OBD II system already does all the necessary work, it would be inefficient, costly, and time consuming to pursue the other options.

To access the OBD II system, there are several options, shown below. The advantages and disadvantages of each are shown in Table 7 below.

1. Use the input and output ports of a microcontroller (MCU), and develop code to interpret the OBD II input and output signals that the OBD II can understand.
2. Devise a system of various analog components to translate OBD II and some form of communication that an MCU can understand.
3. Use a single chip from ELM Electronics (the ELM322) to convert OBD II to EIA-232, and use an MCU that can understand EIA-232.

Method	Strengths	Weaknesses
Write Code	• Least expensive option of the three • Adds no weight or extra size to the VEC • Slow	• Code would be very difficult to develop • Large development time needed • Largest potential for error
Design Circuitry	• Most flexible of the three • Fast	• The design would be very difficult for even an experienced electrical engineer, let alone for a student computer engineer • Large development time needed • Adds weight and extra size to VEC
ELM322	• Simplest Option • Least development time needed • Most MCUs can understand EIA-232 • Fast and accurate	• Most restrictive option – limited to the choices that ELM made • Adds weight and extra size to VEC

Table 7: Pros and Cons to OBD II interface methods

Of the three options, the third is the best for this project. The advantages of the ELM322 far outweigh the disadvantages, and the ELM322's $12 price tag is not large enough to offset the advantage of simplicity.

The ELM322 easily meets the applicable engineering requirements generated, shown in Table 8 below.

Requirement	ELM322	Required Value
Temperature Range	-65ºC to 150ºC	-40ºC to 150ºC
Supply Voltage	5V	< 12V
Power Consumption	2.4mA	< 10A for system
Cost	$17.65 with shipping	< $500 for system
Size	8 Pin DIP	Between 2in x 3in x 2in and 6in x 9in x 3in (L x W x H)
Weight	Approx 0.1 lb	< 10 lbs, main unit
Speed	10.4 kHz communication to OBD II, 9600 baud to MCU	< 0.5 sec for object detection to data output

Table 8: ELM322 vs. Engineering Requirements

In addition, the ELM322 will allow the VEC to access and clear any trouble codes stored in the OBD II system, and enter any of the diagnostic modes that the OBD II provides. A trouble code indicates an error condition in the vehicle, with the code indicating the exact condition. In normal operation, the VEC will check for and alert the driver to any trouble codes automatically. The user will also be able to manually check for and clear the trouble codes through the PDA interface.

As a direct result of the ELM322 choice, the VEC project will need to use the EIA-232 communication protocol. This leads directly into the next connection issue. The VEC must also interface to a PDA in order to interface with the user. The options for that communication are:

1. USB
2. EIA-232
3. Firewire
4. Dallas One-Wire
5. External modem (external to the MCU)
6. Wireless communication

The advantages and disadvantages of these options are shown in Table 9, below.

Method	Strengths	Weaknesses
USB	• Fast data transfer rate • Most PDAs have a USB connection	• Few MCUs have a USB connection and those that do are more expensive • The USB interface for a PDA is more difficult to program for than a serial port
EIA-232	• Many inexpensive MCUs have two EIA-232 ports • The VEC will already be using this protocol • Serial port access is easy to program on both computers and PDAs	• A MCU with two EIA-232 ports is more expensive than a MCU with one • Data transfer rate is slower than USB or Firewire
Firewire	• Fastest data transfer rate	• Can't find even one MCU that supports Firewire natively
Dallas One-wire	• Only two wires needed	• Would need to buy or build a device to convert the protocol to a format that a computer or PDA could understand
External modem	• Fast • Standard interface • Most computers already have modems	• External modems are expensive • Would need to write code for the MCU to communicate with the modem • Most PDAs don't have a modem
Wireless communication	• No wires needed • Fast	• Extra devices would be needed to transmit and receive on both ends • Longer development time needed

Table 9: Pros and Cons of connectivity between Rabbit 3000 and PDA

From examination of Table 9, the best choice for the VEC is the EIA-232 protocol, due to its near universal use in PDAs, and the fact that the protocol is supported by most MCUs. The slower data rate (9600 baud) is not slow enough to miss the engineering requirements, and the negligible development time and additional cost easily meet the requirements. The Rabbit 3000 has six serial ports, each of which can be configured for EIA-232 with the use of a MAX3323, and only cost $39.00 for the entire development board.

The last connectivity issue is the communication between the Rabbit 3000 and the sensors. There are two possible protocols for communication – EIA-232, and a ‘homemade’ code using the input and output ports of the Rabbit 3000. These two choices could be done wirelessly or wired. The options then become:

1. EIA-232, wired
2. EIA-232, wireless
3. Input and output ports of the Rabbit 3000, wired
4. Input and output ports of the Rabbit 3000, wireless

The advantages and disadvantages of these options are shown in Table 10, on the next page.

Method	Strengths	Weaknesses
EIA-232, wired	• Most MCUs have SCI ports • Wires are less likely to have interference errors • The VEC will already be using EIA-232	• Since two SCI ports are already needed, this would require an MCU with 4, which is more expensive and much harder to find • Wires will need to be run through the entire vehicle to connect the various devices • Will make the system harder to install and transport between vehicles • Most sensors and LCDs don't have EIA-232 built in, thus requiring additional circuitry to encode data • Sensors and LCDs that due support EIA-232 are more expensive
EIA-232, wireless	• Most MCUs have SCI ports • No need for wires • Allows the system to be more portable, and easier to install • The VEC will already be using EIA-232	• Since two SCI ports are already needed, this would require an MCU with 4, which is more expensive and much harder to find • Wireless transmission is more prone to errors from interference • Would require transmitters and receivers • Most sensors and LCDs don't have EIA-232 built in, thus requiring additional circuitry to encode data • Sensors and LCDs that due support EIA-232 are more expensive
MCU, wired	• Doesn't require a third SCI port • Wires are less likely to have interference errors • Cheaper than building circuitry for supporting EIA-232, or using devices that have support built in • Easier than designing circuitry to support EIA-232	• Requires an input capture for each sensor • Wires will need to be run through the entire vehicle to connect the various devices • Will make the system harder to install and transport between vehicles
MCU, wireless	• Doesn't require a third SCI port • No need for wires • Allows the system to be more portable, and easier to install • Cheaper than building circuitry for supporting EIA-232, or using devices that have support built in • Easier than designing circuitry to support EIA-232	• Requires an input capture for each sensor • Wireless transmission is more prone to errors from interference • Would require transmitters and receivers

Table 10: Pros and Cons of Rabbit 3000 connectivity to sensors

The additional cost in both development time and equipment rules out using the wireless approaches, even though wireless connections would most easily meets the portability requirement. As such, the portability requirement will likely not be able to be met. Fortunately, this was not a critical requirement, only a desired option. Instead, installation of the VEC will likely require drilling small holes in the vehicle, and running wires between components. This should not be any more difficult than installing a stereo or speakers, and, as such, will not likely cause any undue problems for the user.

Furthermore, the wired option is more reliable than the wireless option. Wireless communication is subject to interference and jamming, much more so than wired communication. Since reliability and accuracy are critical requirements, wireless communication should be avoided.

The last choice was to use the MCU's I/O ports to interface to the sensors. Although that requires input capture for the sensors, this is a common feature in MCUs, and should not add additional cost to the system. Using EIA-232, however, would add additional cost. Since most sensors do not have EIA-232 support built in, we would be required to buy more expensive devices, buy converters, or attempt to design and build our own converters. In addition, MCUs that have four EIA-232 ports would be more expensive, and, in fact, we have found none yet that have four. Since each case involves additional cost and/or time, the EIA-232 protocol was rejected for this choice. Therefore, the final decision was to use the MCU's I/O ports in a wired fashion.

4.4.2 Microcontroller

The performance requirements for this portion are that the VEC must be able to accept input from the distance sensors, from the OBD II system, and from the PDA. It must also be able to send the controlling pulse to the sensors, request data from the OBD II system, and send data to the PDA. The MCU must compute the distance of any object in range, how fast the object is closing, and decide whether or not to output a warning. Finally, the VEC must accept user input, and determine the appropriate action to take based on the input. All of this must be done quickly enough to be useful, but the components must be as inexpensive as possible.

The methods that could be used to perform all the required actions are:

1. MCU
2. PDA
3. Notebook computer
4. A digital system designed to perform all the tasks

The different options are compared in Table 11, below.

Method	Strengths	Weaknesses
MCU	• MCUs can easily perform all the tasks • The programming for each task is straightforward • Development time would be the least of the 5 options	• Must find an MCU with enough input capture and output ports • MCUs are slower than notebook computers and PDAs • MCUs are less capable than notebook computers and PDAs
PDA	• More flexible than a MCU • More instructions than a MCU could mean a more powerful program • Audio warning could be done through internal speakers	• Much more expensive than MCUs • Only one output port (the serial port) so input would need to be processed prior to reaching the PDA • Similarly, output would need processed after leaving the PDA
Notebook Computer	• More flexible than a MCU or PDA • More instructions than a MCU or PDA could mean a more powerful program • Fastest processing of all the options • Audio warning could be done through internal speakers	• The most expensive realistic option • Without input capture, the sensor output will need to be processed before getting to the notebook, adding development time and cost • The LCD driver would need to be hacked from some available port
Digital Logic System	• Most flexible option	• Most difficult to design • Probably more expensive than a CPU • More expensive to maintain

Table 11: MCU Options

After evaluation of the data in the table above, we decided to use a MCU. Of the other options, the MCU will most easily meet the requirements, while keeping the cost of the system and the development time minimal. The notebook option, although fast, accurate, and more flexible, is too expensive to include with the system, while keeping the system cost under $500. Since most of the general public do not have notebooks, not including but requiring one for using the system would also be unacceptable, as it would severely limit the number of potential buyers. The PDA would give us visual alerts but lacks input capture capabilities; hence it would require more hardware for the input capture to be accomplished.

Finally, the digital logic system, while also effective, would be difficult to design, and likely more expensive than an MCU. Further, the failure of a single gate in the digital logic would require a great deal of testing to determine which component failed. With an MCU, only one chip would need to be tested.

4.4.3 User Interface

As the VEC system is not being designed to be completely autonomous, we need some way for the user to interact with the system. Whether it is to adjust the volume of the system, retrieve past data from the system, or anything else that requires human interaction, the possible methods for a user interface are:

1. Use pushbuttons that would be built in to the system and hardwired to do certain functions
2. Use a touch-screen display so the user can touch a part of the screen that would be programmed to do a certain function
3. Use voice recognition technology so the user can verbally state commands, rather than having to press something
4. Connect a 104-key keyboard to the system
5. Connect a 10-digit keypad, or a number pad
6. Use a USB or PS/2 mouse, which would in turn control the system by the mouse movement and button presses
7. Use a PDA's touch-screen, which allows for many virtual buttons

The strengths and weakness of each are shown in Table 12, on the next page.

Method	Strengths	Weaknesses
Buttons	•Easy to use •Easy to implement •Lowest Cost •Most configurable	•Takes up space •Requires extra hardware
Touchpad	•Simplest Design •Least amount of space •Interactive	•Extremely expensive
Voice Recognition	•Don't have to look away from the road	•Extra circuitry required •Requires voice specific programming •Increases space
Keyboard	•Already designed	•Extra circuitry required to decode •Too many controls •Increases space •Would have to look away from road
10 digit keypad	•Already designed •Compact •Low Cost	•Extra circuitry required to decode •Too many controls •Would have to look away from road
Mouse	•Compact •Few Controls •Low Cost	•Extra circuitry required to decode •No surface to use for mouse pad •Would slide all over as car was moving
PDA Touch-screen	•Simple design •Compact •Interactive •Easy to mount •Relatively configurable	•Requires special software to be written

Table 12: Pros and Cons of User Interface Options

Using buttons as input would provide a very configurable system that would be easy to use, and at a low cost. However, it would take up a lot of space since it would require extra hardware. Voice recognition would provide an interface where the user would not have to look away from the road, but it requires voice specific programming, and extra hardware, which means more space. Both a keyboard and 10-digit keypad would provide too many controls, making the system complicated, and increasing the space greatly. A mouse is compact and has few controls, but requires extra circuitry and a smooth 'mouse-pad like' surface. A touch-pad is a simple design, and has the advantage of combining the user interface with the display, but the cost is out of our budget. A PDA, which is the option we chose, has all the advantages of a touch-screen and is already available to us because Behrend owns one. Also, it is required for the competition that we entered.

4.4.4 Display

In order for the user to interact with the VEC system and receive feedback from the system, some type of display should be used to pass information on to the user quickly and efficiently. The various possibilities are:

1. Use a standard CRT (computer monitor)
2. Use an LCD
3. Use many LED's configured in a pattern
4. Use a touch-screen display
5. Make a Heads Up Display (HUD) which would be projected onto the windshield of the car in front of the driver
6. Use a PDA
7. Do not use any form of display

The strengths and weaknesses of each are shown in Table 13, shown on the next page.

Method	Strengths	Weaknesses
CRT	•Variable display	•Large •Expensive •Requires 120V AC power supply
LCD	•Custom sizes •Variable display	•Extra circuitry required to decode
LED	•Extremely inexpensive •Smallest size •Least amount of circuitry required	•Non-customizable display
Touch-screen	•Compact •Custom sizes •Integrates controls	•Extra circuitry required to decode •Extremely expensive
HUD	•Small amount of space	•Extra circuitry required to project •Complicated •Cost
PDA	•Small amount of space •Integrates Controls •Variable display	•Requires special software to be written
None	•Least amount of space •Lowest Cost •No circuitry required	•Not able to give visual warning

Table 13: Pros and Cons to Different Displays

While the touch-screen display would be our primary choice, and a space efficient solution, the high price tag is a major weakness as we have to stay under a certain budget and keep the cost of our system down. The PDA, which is the option we chose, provides the benefits of a small touch-screen with the only drawback being that software needs to be written for it. A CRT is simply too large to fit in the front of a car comfortably. An LCD would provide a nice display, but doesn't integrate the controls like a touch-screen or PDA. Using LEDs would save space, but does not provide a display that can be changed. While a HUD would be space efficient, it is complicated to design and could cost too much for our budget. Not using a display does not allow for a visual alert for the user.

4.4.5 Power

The VEC system is not made to create its own power; therefore it needs some sort of power supply. The possible options for this are:

1. Connect the system to the car battery directly
2. Create a solar powered system and use that to power the VEC
3. Use fuel cells to power the VEC
4. Connect another battery source, such as a small 9 volt battery
5. Create a thermal transfer system, which would use the heat from the engine and convert it to electrical current
6. Create an adapter and plug the system into the cigarette lighter
7. Run the power through the fuse box, using an empty slot
8. Use the 12V output from the OBD II system

The strengths and weaknesses of each option are shown in Table 14 on the next page.

Method	Strengths	Weaknesses
Car Battery	•Readily available •Already at 12v DC •Sufficient power •No extra space •Long lifetime	•Requires extra wires and at least one fuse •Need to investigate and conform to additional standards
Solar	•Efficient •Doesn't require voltage source •Long lifetime	•Requires storing energy to be able to drive at night •We aren't knowledgeable in the subject •Extra space for panels/collectors •Extra circuitry required to convert •Extremely expensive
Fuel Cell	•Efficient •Doesn't require voltage source •Long lifetime	•We aren't knowledgeable in the subject •Extra circuitry required to convert •Extra space •Extremely expensive
9v Battery (or other battery)	•Low cost	•Extra space •Short lifetime
Thermal Transfer	•Efficient •Can use heat from engine •Long lifetime	•Extra circuitry required to convert •Extra space for thermal converters •We aren't extremely knowledgeable in this area
OBD II Output	•Readily available •Direct Connection to battery •Already at 12V DC •Sufficient power •No extra space •Long lifetime	

Table 14: Pros and Cons to Power Supplies

Using thermal transfer, solar, or fuel cells for power is currently beyond our knowledge and would require extensive research. A 9V battery is low in cost and provides sufficient power, but has a very short lifetime compared to the car battery. The car battery is already in the car, so we would not have to add another source and increase the size of our system, it has sufficient power to run the VEC system, and it has a long lifetime. The OBD II output, which is the option we choose, provides a direct connection to the battery with easier access.

4.4.6 Sound Output

Along with a visual output for information regarding the system status, the VEC will feature sound outputs as well in case the user may not be able to reference a visual object at any given time. While these sound events may be just system status such as ready messages, a subcategory of alerts is discussed further in this document, as they are higher priority than a general sound message.

Many electronic devices may emit beeps and tones in a varying series to distribute information about their state. However, due to the nature of the VEC and circumstances while using the system (driving), we feel that a system that is more straightforward is necessary. After review of Table 15, shown on the next page, the best sound output option is using a system speaker that emits the sound from the voice synthesis chip. Voice synthesis is the best option as far as meaningful output, as this form allows for variable messages and a far less limited amount of messages possible than voice recorder chips.

Method	Strengths	Weakness
Existing Speaker	• Space saving • Cuts cost • Less wiring	• Involves dealing with car wiring more • Requires removal of stereo to install
System Speaker	• No interaction with car speaker • Speaker concealable in main unit	• Raises cost • Increases size
Buzzer	• No visual required • Small size • Inexpensive	• Unexplained meaning • Stereo could overpower
Horn	• Louder than buzzer • On every car	• Involves dealing with car wiring more • Lack of meaning
Bullhorn	• Can send messages • Loud	• Bulky
Cell Phone	• Could log instances on phone	•Illegal in some states
Voice Recorder	•Inexpensive •Meaningful alerts	•Limited storage space •Limited number of responses
Voice Synthesizer	•Large quantity of responses possible •Expandability without reprogramming voice commands	•Cost •Some overhead with operation

Table 15: Pros and Cons to Audio Alert Types.

4.4.7 Alerts

When an event occurs that the operator must know about, the VEC system will gain the user's attention by having an alert. There exist many options for alerts that could be implemented. Important to the selection of the alert is its cost to implement and its overall effectiveness during an event. While a flashing LED is fairly inexpensive, it may not be easily noticeable to all VEC users. Furthermore, with its simple on/off behavior, an LED has no way to convey meaning behind the alert without resorting to some sort of coded scheme.

After reviewing the methods in Table 16, on the next page, we have decided to use a voice synthesizer in the VEC in order to allow many meaningful voice outputs and allow for the expansion of such features in future versions of the system.

Method	Strengths	Weaknesses
Existing Speakers	• Space saving • Cuts cost • Less wiring	• Involves dealing with car wiring more • Requires removal of stereo to install
System Speakers	•No interaction with car speakers • Speaker concealable in main unit	• Raises cost • Increases size of main unit
Flashing Lights	• Small size • Low cost • Low power	• Little meaning behind a light • May adversely affect vision in a car
LEDs	• Inexpensive • Bright • Low power	• Non user replaceable • Little meaning behind a pulsing light
Shock	• Very noticeable	• Health hazards • May affect steering wheel usability
LCD output	• Detailed info available • Can flash like LCD • Centralized location of messages	• Cost • Space issues
Vibration	• Silent • Very noticeable	• May scare driver • May affect steering wheel usability
Voice Recorder	•Inexpensive •Meaningful alerts	•Limited storage space •Limited number of responses
Voice Synthesizer	•Large quantity of responses possible •Expandability without reprogramming voice commands	•Cost •Some overhead with operation

Table 16: Pros and Cons to Alert Methods.

4.4.8 Internal Size and Internal Mounting

There are several size and mounting issues concerning the VEC system. The first of these issues that will be discussed is internal size and then mounting. Sizing and mounting generally go hand in hand, so they have been placed together. Since the display size is not known it makes internal size very difficult. By staying as broad as possible for now, and keeping within some reasonable constraints, options are open when PCB fabrication time comes.

Weighing the strengths and weaknesses, the decision was made for the VEC system's internal size to be less than a shoe box lid. Since the system needs to alter the current appearance of the car as little as possible, while still leaving enough room for the components and a display that is big enough to read, this makes less than a shoe box lid the best solution. The expectation is that the VEC will use a 3" x 6" PDA, and thus require slightly larger than that, plus 2" to 3" of depth to allow for the MCU, ELM322, and other internal components.

Now, how to mount this system to the interior of the car? Preferably, the system could be integrated into a DVD navigation system if the VEC was to be a stock option. Since this is a retro fit system, that option is not viable. Making a dash mount is the best option, as the dash mount allows the system to be universal, easy to see, and easy to interact with. These are two very important needs for the VEC system.

4.4.9 External Size and External Mounting

There are several size and mounting issues concerning the VEC system on the exterior of the car. This is a critical issue with the needs of the VEC system. If system detracts from the appearance of the vehicle, the VEC system will be a flop.

Looking at the options, the VEC will probably have half-dollar sized external components, with extra circuitry not visible to other people. Current cost is $37 for the Polaroid 6500 ultra sonic range finder.

Mounting of these sensors will need to be secure and in strategic places. Current options include under side view mirrors, rear bumper, and the car frame. The ultimate location will be determined by the exact needs of the sensors.

The options, and their strengths and weaknesses are shown Table 17, on the next page.

Method	Strengths	Weaknesses
Quarter	● Aesthetically more pleasing ● Increases mounting points	● Increases difficulty of design ● Increases difficulty of weatherproofing
Shoe Box	● Able to fit an "All In One" range finder unit	● Aesthetically not pleasing ● Hard to mount
Inside Bumper	● Aesthetically more pleasing ● Could be integrated with bumper design	● Increases retro mount difficultly ● Would need to be key colored
Side View Mirror	● Good range for blind spots ● When integrated with mirror would increase aesthetics	● Possible sensor loss ● Wires could be pinched in mirror hinge ● Exposed to more wind and elements
Frame	● Helps hide sensors from elements ● Solid structure to mount to	● Wiring ● Ground clearance

Table 17: Pros and Cons to Size Options

4.4.10 Sensors

There are several possible sensors that the VEC system could use to detect objects. The system needs a sensor that can be fast and accurate along with one that can be used for approximately 10 feet. Since the system is going to require 4-5 sensors, cost is also a major issue.

All these factors were taken into consideration bringing the ultra sonic to the forefront. The ultrasonic can spread its signal out, unlike the light sources that are discussed in the table below. It is also a cheaper solution with less interference. Finally, the Polaroid 6500 is excellent for size, cost, accuracy, and keeping the exterior of the car as stock as possible.

To enable faster calculations of an object's closing rate (how quickly it is getting closer), the car computer option will be used, since the OBD II system provides this. Without the OBD II, the VEC would need several distance measurements and the exact time between each measurement. This does not allow for interrupts, and doesn't leave much time for MCU calculations.

The options, and their strengths and weaknesses are shown in Table 18, below.

Method	**Strengths**	**Weaknesses**
Ultrasonic	• Small size with no visual display (laser light) • Large range - 6 inches to 28 feet • Accurate • Inexpensive	• Range finder cost • Increases difficulty of weatherproofing
Infrared	• Non-visible light • Small range - usually under 6 inches	• Interference • Hard to mount
Laser	Longer distances Highly accurate	• Expensive • Pin pointing would have to be done on object
Camera	State of the art visual imaging	• State of the art visual imaging • Less accuracy • Expensive
Car Computer	Easy capture speed Warnings can be saved or displayed	• Does not give distance output, only the vehicle speed

Table 18: Pros and Cons to Sensor Options

5. Testing, Results, and Discussion

5.1 Rabbit 3000

The Rabbit 3000 was purchased on a RCM 3610 Core Module, which is a PCB with the Rabbit 3000 pre-mounted, as well as Flash RAM, SRAM, a crystal oscillator, and all other circuitry needed to use the Rabbit 3000. It also includes a 2x20 pin header for I/O pin access, and a 2x5 pin header for programming. A programming cable was also purchased for the 2x5 pin header, that handled the EIA-232 conversion for interfacing to a computer for programming and debugging. As such, the testing of the Rabbit 3000 was very simple.

The Dynamic C [11] development software that was used included several test programs. Several of these were compiled, downloaded to the Rabbit 3000, and run, to test the general functioning of the Rabbit 3000. The Rabbit 3000 passed these tests.

A test program was then written to implement costates (the multitasking that the Rabbit 3000 is capable of) that caused four LEDs attached to four I/O pins to be turned on and off at different rates. This Rabbit passed this test as well.

The last test program written was designed to test the serial communication of the Rabbit 3000. This involved two costates. One costate did nothing but try to receive characters, and then echoed these back on the same line that it received them on. This was implemented using cofunctions, which, when used in a costate, allows other costates to run until successfully completed. The second costate performed 100 arccosine computations on 100 random numbers, then waited for 0.5 second. The wait was also done as a cofunction, allowing the serial communication costate to run. The serial port was then connected to a computer, and a software terminal emulator called HyperTerminal was run to communicate with the Rabbit 3000. Even with the long time taken to do the arccosines and random number generations, the Rabbit 3000 still managed to “catch” all of the characters sent to it. This was important to ensure that the communication between the PDA, ELM322, and Rabbit 3000 could take place while allowing time for the Winbond WS701EF/T and ultrasonic sensor operations.

The last test performed on the Rabbit 3000 tested the Flash RAM. A downloaded program is stored in the Flash RAM, and can be run from the Dynamic C environment, or by removing the programming cable. The Flash was tested by removing the programming cable while the serial communication test was loaded. The Rabbit 3000 immediately ran the installed program. This test ensured that the Rabbit 3000 would run our final program.

5.2 ELM322

The ELM322 circuit described in section 4.3.2 and shown in Figure 7 on page 31 was transferred to a PCB. This design was done with Orcad Layout software, and the PCB was milled with the assistance of Glenn Craig. The components were then soldered to the board, resulting in the PCB shown in Figure 17 below.

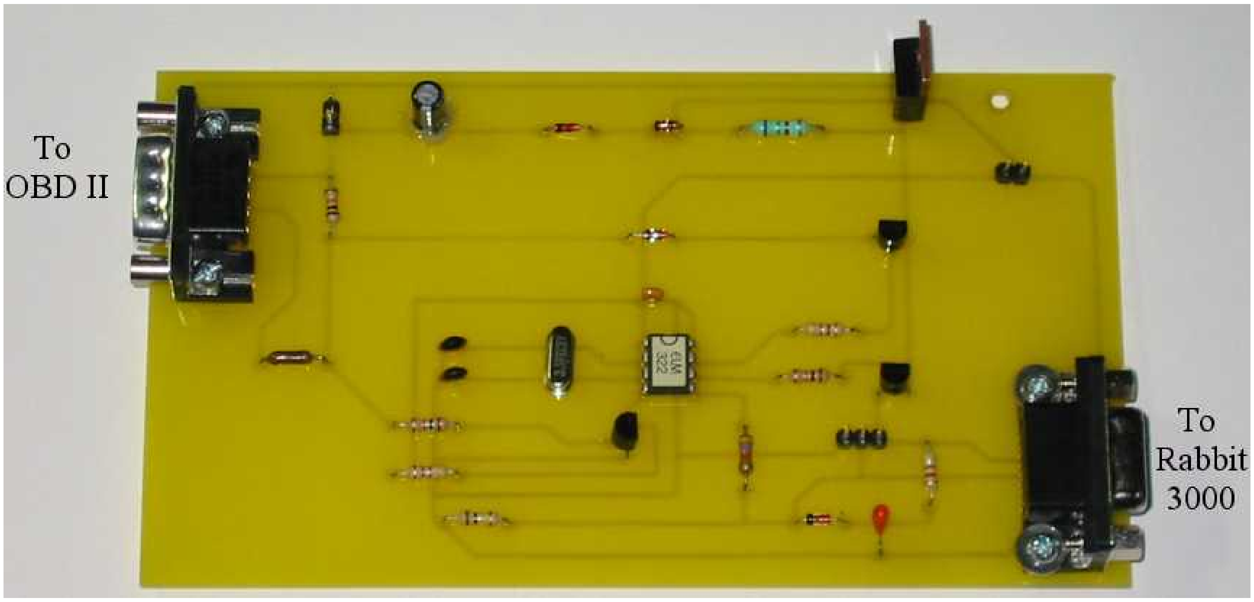


Figure 17: ELM322 circuit PCB

From previous design decisions, the VEC must be able to get the speed of the vehicle, the number of DTCs, any stored DTCs, and also be able to clear the stored DTCs. From research into DTCs, it was discovered that some DTCs mean different things depending on various digits of the VIN, therefore the VION must also be retrieved from the vehicle. All of this is done through the OBD II interface, and thus through the ELM322. These, then, were the requirements used to test the ELM322 PCB.

The first tests performed on the ELM322 were done with HyperTerminal. By opening a HyperTerminal connection through a computer's serial port, the ELM322 and the computer can communicate. The other DB9 connection on the ELM322 PCB is used to connect the ELM322 to the OBD II system.

The HyperTerminal tests ran flawlessly. The ELM322 documentation was thorough enough to allow a demonstration that the ELM322 PCB was functioning correctly. Tests performed, results, and meaning are shown in Table 19, below. One thing to note is that the VIN tests required changing the headers sent by the ELM322 to the OBD II system from "68 6A F1" to "6C 10 F1". Further, it takes 3 separate commands to get the entire 17 digit VIN.

Test	Data Sent	Data Received	Meaning
Vehicle Speed	01 0D	41 0D 00	Speed is 0 kph
Number of DTCs	01 01	41 01 81	There is 1 DTC, and the Malfunction Indicator Lamp has been lit for it
Request DTCs	03	43 01 13 00 00 00 00	The DTC stored is 0113, which translates to P0113. This means that the Intake Air Temperature circuit is malfunctioning
VIN Part 1	3C 01*	7C 01 00 31 47 31 4A 46	The first 5 digits of the VIN are 1G1JF. The VIN is output in ASCII code.
VIN Part 2	3C 02*	7C 02 35 32 34 34 57 37	The next 6 digit of the VIN are 5244W7.
VIN Part 3	3C 03*	7C 03 32 36 37 39 30 39	The last 6 digits of the VIN are 267909.
Clear DTCs	04	44	The DTCs have been successfully cleared.
Number of DTCs After Clearing	01 01	41 01 00	There are no DTCs stored.

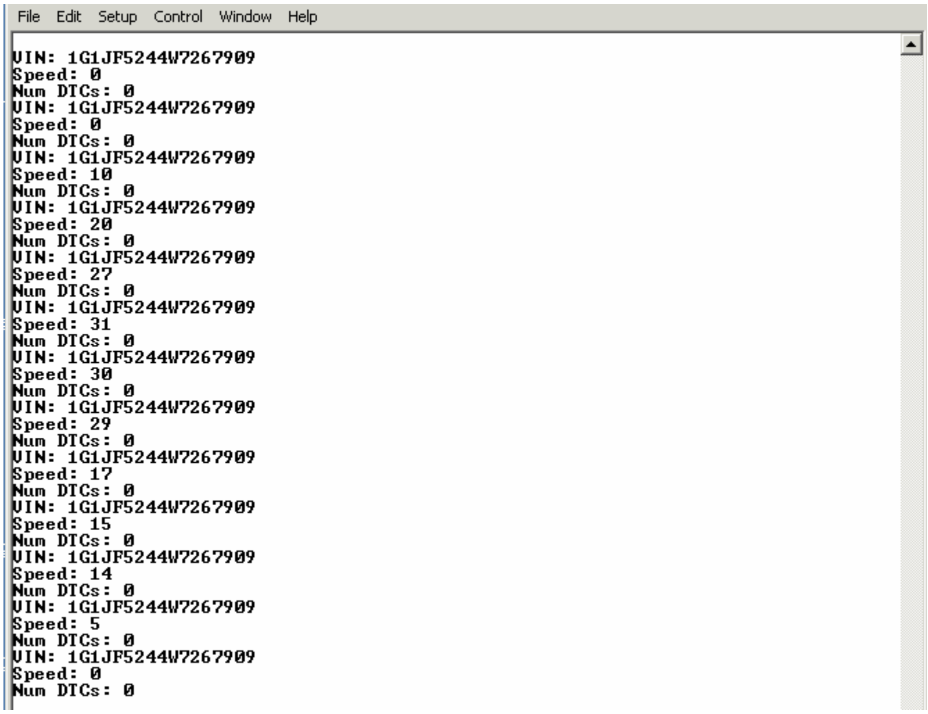
Table 19: ELM322 tests and results, through HyperTerminal

The next tests done were also done HyperTerminal. The ELM322 by default echoes characters received from the terminal (or in the case of the VEC from the Rabbit 3000). It also outputs extra line feeds, and translates the data received from the OBD II system into ASCII. All of this is done to make the ELM322 easily compatible with programs like HyperTerminal, but only serve to slow down a system using a microcontroller. To speed up communication, the echo, line feeds, and ASCII translation can all be turned off.

In order to know what data to look for on the Rabbit 3000 side of the interface, the same tests performed in HyperTerminal on the ELM322 as above were performed again, but with the sped up communication. The results were logged to several text files, and read with a hex viewer, to see the binary data sent and received. This allowed the creation of code for the Rabbit 3000 to perform the required tasks.

The last tests were done using the Rabbit 3000. Code was generated from the results of the previous two tests, and was tested using two of the serial ports from the Rabbit 3000 as output. Serial port C was used to access the ELM322, and serial port E was transmitted to HyperTerminal, to verify that the Rabbit 3000 and ELM322 were operating and communicating correctly. Since serial port E is also used for the PDA, this had the added benefit of testing that port's function.

After several failed attempts, mostly due to coding errors, the code passed the test. A screen capture of the HyperTerminal window is shown in Figure 18, below, and the final code developed is shown on the attached CD.



```
VIN: 1G1JF5244W7267909
Speed: 0
Num DTCs: 0
VIN: 1G1JF5244W7267909
Speed: 0
Num DTCs: 0
VIN: 1G1JF5244W7267909
Speed: 10
Num DTCs: 0
VIN: 1G1JF5244W7267909
Speed: 20
Num DTCs: 0
VIN: 1G1JF5244W7267909
Speed: 27
Num DTCs: 0
VIN: 1G1JF5244W7267909
Speed: 31
Num DTCs: 0
VIN: 1G1JF5244W7267909
Speed: 30
Num DTCs: 0
VIN: 1G1JF5244W7267909
Speed: 29
Num DTCs: 0
VIN: 1G1JF5244W7267909
Speed: 17
Num DTCs: 0
VIN: 1G1JF5244W7267909
Speed: 15
Num DTCs: 0
VIN: 1G1JF5244W7267909
Speed: 14
Num DTCs: 0
VIN: 1G1JF5244W7267909
Speed: 5
Num DTCs: 0
VIN: 1G1JF5244W7267909
Speed: 0
Num DTCs: 0
```

Figure 18: HyperTerminal output of the results of the final ELM322 with Rabbit 3000 test.

5.3 PDA

During the development of the PDA software, the code was tested each time something new was added. First, the user interface was designed and tested on the Palm OS emulator. The final user interface can be seen in Figure 19 below.

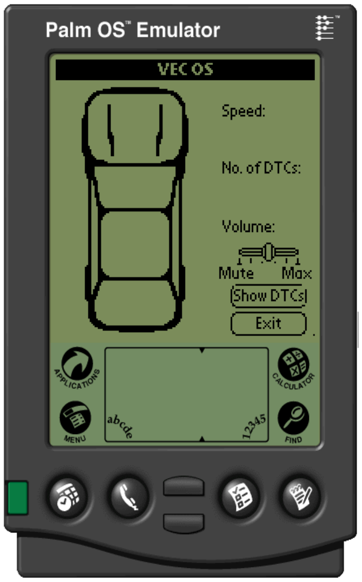
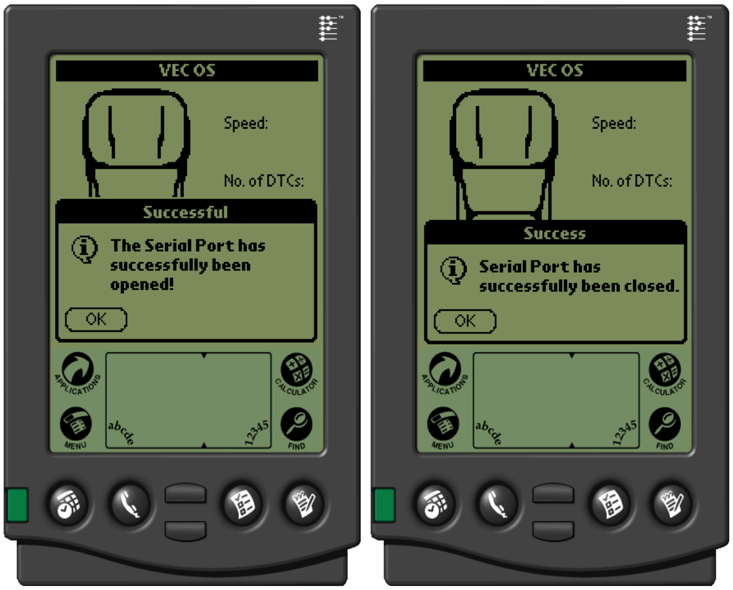


Figure 19: PDA User Interface

Once that was completed, code was written to control the serial port. This code could not be tested on the emulator, because the emulator does not have a serial port, and had to be installed on the PDA after every compilation. The first thing tested on the serial port was the opening/closing of the port on the PDA. The PDA displayed messages stating whether the port had been successfully opened or closed. These messages can be seen in Figure 20, on the next page.



Successful

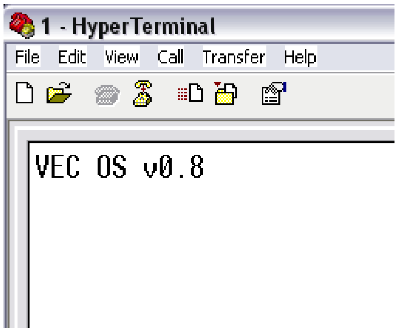
The Serial Port has successfully been opened!

Success

Serial Port has successfully been closed.

Figure 20: Serial Port Success Messages

When this was completed, code was written to send data out of the serial port. This code was tested by connecting the PDA to the serial port of a computer and using HyperTerminal, or any other terminal emulator, to output the characters the PDA sent. An example of the code output can be seen in Figure 21 below.



```
VEC OS v0.8
```

Figure 21: Serial Port Send Test

Code was then written to receive data through the serial port, and was tested in almost the same manner. The PDA displayed a message stating what characters it had received through the serial port. When it was determined that all functions of the serial port were working correctly, functions were written to manipulate and handle the data.

When it was determined that the serial port was operating correctly, the serial port on the PDA was connected to a serial port on the Rabbit 3000 (via a max-3232 chip). Code was written for the Rabbit 3000 that waits until it receives a byte from the PDA, and the sends out the 64 byte sequence described before. Initially, the Rabbit 3000 sent the PDA fixed numbers to verify the connection it was working correctly.

5.4 Winbond WTS701EF/T

Once the PCB for the WTS701 was completed and the chip mounted, testing of all connections began. First before any power was applied to the PCB, all pins were checked for short circuits with their neighboring pins. Due to the size of the package, short circuits between pins were a major concern for the team. After it was determined no short circuits among pins were present, connections between each pin and its corresponding header on the PCB were tested to ensure proper connectivity. A requirement of 4 Ohms or less was necessary to guarantee connection. After these connections were tested, the PCB and other necessary hardware (capacitors, crystal, etc) were placed onto the project board and connected separately to a power supply. Voltage and current measurements were taken then to check correct wiring methods. Once current draw was found to be within allowable limits, the WTS701 PCB was wired into the general power circuitry of the VEC. The PCB containing the WTS701EF/T as mounted to the VEC main board appears in Figure 22 below.

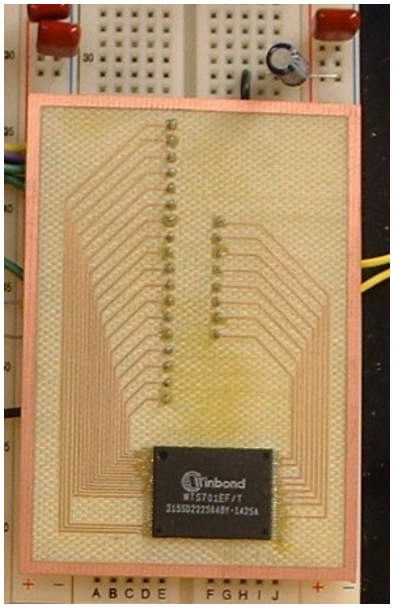


Figure 22: Winbond WTS701EF/T

After this testing was complete, the operations of the WTS701 were still left unchecked. In order to test its functionality, a small software program was written for the Rabbit 3000 to theoretically cause the WTS701 to output the popular phrase "Hello World." Device initialization could be monitored by the usage of print statements in the software causing the Rabbit 3000 to output to its STDIO screen the data received from the WTS701. In order for the WTS701 to speak, several operations had to be preformed. The chip is initially in power down mode. Commands must be sent via SPI to configure the clock register of the WTS701 for the correct clocking method. Expected status register after this command is hexadecimal 80, which displayed via STDIO. The power up command must then be sent to the WTS701 and a start up time of one millisecond is required. Expected status response from the WTS701 is again hexadecimal 80 via STDIO. Following power up, the volume level on the WTS701 must be set, and finally the CONV command followed by "Hello World" and ending with Hexadecimal 1A is sent to the chip. The ultimate conclusion of the test was successful upon hearing a woman's voice speak "Hello World" as expected through the speaker.

5.5 Polaroid 6500

Testing and verification first started with attaching the Polaroid 6500 sensor to a power supply, frequency generator, and an oscilloscope. Initial lab testing allowed for The VEC team to watch the ECHO line go high and vary with the changing distance of an object from the sensors. Laboratory results revealed to the VEC team a power supply issue. When a "ping" (INIT line is sent high having the Polaroid 6500 sensor send out a 50 kHz sound signals) is sent from the Polaroid 6500 sensor; 2 amps are briefly drawn. Since this draw only occurs from the "ping" it can be solved by using a 470uF capacitor [8]. The VEC team has one 1000uF capacitor with the Polaroid 6500 sensor and with the main bread board. These two capacitors deal with any failure rate of just one capacitor and 1000uF capacitors were used instead of 470uF due to availability.

After hardware verification was completed, testing was started on software application. This started as a simple frequency generation program from the Rabbit 3000, and then progressed to a program to output the time for the signal to be received. Upon signal sensor operation, testing was done with several small and large objects, both stationary and moving.

These were tested both near (1-3 feet) and far (10-20 feet). After all detection testing, code was adapted for all 5 sensors. Similar laboratory tests were conducted with the PDA, Winbond WS701EF/T, and all 5 sensors attached. Then testing moved to the first on car testing. The VEC team then tested on a 1998 Chevy Monte Carlo LS.

5.6 Combined Systems Testing

Once it was determined that all separate subsystems for the VEC were functioning correctly individually, integration of the systems was started, and testing of these systems working cooperatively was required. First, the VEC team worked on and tested the communications between the PDA, the OBD II, the ELM322, and the Rabbit 3000. In order to do this the team was required to move testing into the 1998 Chevy Monte Carlo LS. The PDA and Rabbit 3000 code pertaining to the OBD II systems and the PDA communications were singled out and used for these tests. All other systems remained commented out of the VEC system. It was discovered during this period that communication between the PDA and Rabbit 3000 was not strictly adhering to the voltage level requirements, and buffer circuitry was necessarily added to the VEC system board to adjust the voltage levels accordingly. The PDA, Rabbit 3000, the OBD II, and ELM322 were tested again, and it worked perfectly.

Each sensor algorithm was tested individually. To do this, a team member walked around the perimeter of the car. As he made his way around the vehicle, each sensor detected him and determined he was too close to the vehicle. This caused the PDA and Winbond WTS701EF/T to output warnings. He then tested the warning levels by moving toward and away from the vehicle. When he was in the range that the sensors determined to be a major warning, the PDA and Winbond WTS701EF/T both output a major warning. Likewise, when he was in the range that the sensors determined to be a minor warning, the PDA and Winbond WTS701EF/T both output a minor warning. When he was out of the warning ranges, no warnings were output.

The next tests performed were complete system tests. The sensors were mounted to a 1998 Monte Carlo, and the remaining components were installed inside the vehicle. The parking test was conducted first, by backing up to a light post. This caused the Winbond WTS701EF/T and PDA to output warnings as the vehicle approached, as expected. Next, the VEC was tested by pulling the vehicle next to another vehicle, which again caused warnings to be output. The last parking test performed was to pull into a spot with vehicles in front and on both sides, causing warnings to be output for all three of the corresponding sensors.

The last tests performed were on Interstate 90. These tests showed that the VEC system automatically changes modes from parking all the way to Autobahn. When we tailgated someone the VEC system warned us of the impending danger. When we started moving left to pass a slower moving vehicle, the left sensor alerted us of a hazard in the left lane. This demonstrated the VEC's ability to expose blind spot hazards.

6. Realization of Requirements, Constraints & Standards

6.1 Rabbit 3000

The Rabbit 3000 was tested to ensure that it complied with applicable requirements, including: system power consumption, internal component operating temperature range, shock and vibration tolerance, main unit size, and main unit weight.

The Rabbit 3000 has a minimal current draw of 10mA to a peak draw of 160mA. It was tested extensively under both light and heavy loads. This peak draw allows for the remainder of the system to pull a maximum of 840mA, if needed, and still fall within the required range.

According to the technical specifications for the Rabbit 3000, it can withstand temperatures ranging from -40°F to +185°F. This range is well within our specified target range of -20°F to 150°F.

The Rabbit 3000 development board that was used measured 2.10 inches by 1.20 inches and weighed 5 ounces. These measurements leave sufficient space for other components to meet the weight and size requirements.

Shock and vibration tolerance could not be tested due to the fact that we did not have time to fabricate a PCB and any vibration could loosen wires on the breadboard.

6.2 ELM322 and OBD II

The ELM322 PCB was tested to ensure that it complied with any constraints applicable. The first was power consumption. The ELM322 PCB was found to have a peak draw of 10mA, which is well below the maximum system power consumption of 1A, allowing plenty of current for other components. The circuit uses the 12V provided by the vehicle, and should operate between the required ranges of -20°F to 150°F. Although the last requirement could not easily be tested, the ELM322 itself has an operating range of -40°F to 185°F. The remaining components, resistors, diodes, and transistors, all have even larger operating ranges.

The ELM322 PCB also falls within the Main Unit Size requirement. The PCB is approximately 6”x3.5”x1” (L x W x H), which is less than 6” x 9” x 3” as required. The PCB also falls easily in the weight range, less than 10 lbs. The PCB weighs less than one pound.

Shock resistance and vibration tolerance could also not easily be tested. It should be noted, however, that the ELM322 PCB has survived a drop from about four feet, and continues to function. Lastly, the ELM322 PCB cost about $50 to make, including the $20 OBD II to DB9 cable, so the PCB is significantly less than the $500 target sales price. This allows enough money for the other VEC components.

The ELM322 PCB also enables the VEC to realize some of the constraints identified. The main constraint was identified under the manufacturability category. The VEC must be able to communicate with the OBD II system of vehicles made after 1996. The ELM322 is specifically designed for this purpose, and as such is ideal for this constraint. Also, since the ELM322 allows any data to be transmitted through it to the OBD II system that the OBD II standard allows, the VEC is able to change as new commands and codes are added by manufacturers. This helps make the VEC sustainable.

The standards identified in the Standards section that the ELM322 PCB conforms to are:

- J1962 – Standard OBD II Connector
- J1979 – Standard OBD II Commands and Timing
- ISO 9141 – OBD II Communication Protocols
- J1850 – OBD II Communication Protocols
- EIA-232 – Standard for Serial Data Communication

The OBD II standards are all conformed to by the ELM322 or the OBD II to DB9 cable used for the interface. The EIA-232 standard is also employed by the ELM322, through the interface between the ELM322 and the Rabbit 3000.

6.3 PDA

The engineering requirements that deal with the PDA are: time from object detection to output, number of buttons, object distance displayed refresh rate.

The maximum time from object detection to output on the PDA is 0.5 second. This meets the requirement of 0.5 seconds, but the Winbond WS701EF/T alerts the user much faster than the PDA, so the PDA does not need to update any faster.

The number of controls VEC uses is 3: 2 buttons and 1 slider. While the PDA itself has 7 buttons on it, none of these buttons are used. Instead, the controls for the VEC system are integrated on the touch-screen. The number of controls we use, thus, is less than the requirement of 5.

6.4 Winbond WS701EF/T

The Winbond WTS701EF/T main task for the VEC is to output the audio alerts of the system according to the particular state of the sensors. In order to do this effectively, it must operate with a fast response time, be audibly clear, and loud enough to alert the vehicle operator over the sound of the car. The speaker driver built into the WTS701 is 23 mW. Combined with using a speaker enclosed in a box, the sound output from the WTS701EF/T is adequate to be heard over the sound of most standard car engines and background noises.

The WTS701EF/T comes standard with the circuitry required to drive a standard eight ohm speaker. Therefore a secondary task of the WTS701EF/T is the volume control operation of the VEC. The PDA display has a volume slider containing eight separate levels. These levels correspond to the seven defined levels the WTS701EF/T driver circuitry has, and the final level is a mute.

6.5 Polaroid 6500

The Polaroid 6500 ultrasonic sensors were able to fit many of the engineering requirements that we had originally set out to accomplish. Size of external sensor unit was required to be under 6in x 9in x 3in. We have accomplished this goal with all outside sensors in

3in x 2in x 1in project boxes. The sensor accuracy was tested in the lab, and was found to be accurate to within less than 1%. In fact, they tested at less than 0.1% error. With the sensors updating every 250ms they more than comply with the engineering requirement of updating every half a second. The 4 Hz frequency of the sensors is to ensure the ability of the sensors to locate faster moving object through the sound waves.

6.6 VEC System

6.6.1 Requirements

As a whole, the system had to meet requirements for reliability, system power consumption, shock resistance and vibration tolerance, sales price, main unit size, main unit weight, and data logging. Reliability of the system could not be determined because reliability data could not be found for many of the components, including fabricated ones. Shock resistance and vibration tolerance also could not be tested as the equipment to do so is unavailable.

Reading from the digital multi-meter in the senior design lab, the maximum current draw for the entire system was 400mA, which is well below our limit of 1A.

Our target sales price was $500. According to the cost table, Table 21 in Section 8, the cost of the components of the VEC is $415.43. The sales price of the unit would not include the cost of the Rabbit 3000 programming cable or the PDA cable. Subtracting these from the price lowers the cost to $370.43, which is under the required amount.

The main unit size was not met, but only due to lack of time. The components, which are currently on a breadboard, could be placed on a PCB, which would be smaller than 6”x9”x3”. The main unit weight is significantly less than the required ten pounds, weighing approximately 4-5 pounds including the breadboard.

Data logging was not implemented in the system because we realized there was no advantage to logging data. As such, logging data would simply increase the complexity and cost of the system, which would be unnecessary.

6.6.2 Constraints

The VEC meets the constraint of being profitable by being inexpensive to build, relative to the cost of an accident. The cost to build the VEC was $370.43, while an accident could potentially cost thousands of dollars.

While the VEC can operate within the required temperature ranges, it does not meet the environmental constraint of weatherproofing. This could be resolved by purchasing weatherproof sensors, which were unavailable to us [12].

To our knowledge, the VEC does not violate any patents or copyrights. Although it does not currently meet wiring standards for vehicles, this could be easily remedied by using automotive wire and harnesses. Also, the VEC does not impair a vehicle's current operation.

In its current state, the VEC does not conform to the health and safety constraints because it is not properly encased. The production model, however, would conform to this constraint as it would be properly enclosed.

The physical characteristics of the VEC meet the manufacturability constraints. The VEC interfaces with the OBD II connections from 1996 to present, the size of the unit does not interfere with operation of the vehicle, and the software for the VEC is coded in accordance to the chosen processors language.

The social constraint of the VEC was that it should have a simple interface that allows targeted audiences to easily use the product. This constraint is satisfied, as the VEC has only one plug to connect.

As the lifetime of the electronic components are longer than the average lifetime of a car, the VEC meets the sustainability constraint. It should be noted, however, that the system is not transferable between vehicles, as we previously thought that it might be, due to mounting issues.

6.6.3 Standards

The VEC follows the following standards:

- EIA-232 – Communication between the ELM322 and Rabbit 3000
- J1850 – Communication between the ELM322 and the OBD II System
- J1962 – Standard OBD II connector
- J1979 – Standard OBD II commands and timing
- C/C++ - Coding standards on both Rabbit 3000 and PDA

7. Project Management Plan

7.1 Jeffery Betts:

Jeffery's major contribution to the VEC project is the operation of the Polaroid 6500 ultra sonic sensor. The Polaroid 6500 sensors come with one transducer and one circuit board. These components were adapted for team VEC use. This included removing the original flat flex connector and replacing with wire that attached to an ATX connector. The Polaroid 6500 sensors were placed in a 3in x 2in x 1in project box for external mounting. Cable and connectors similar to phone wire were used for connectivity. Another project box was used for a "connectivity center" which contained 5 phone jacks (one for each sensor) and two DB-9 connectors for the PDA and EML322. Jeffery also helped integrate the components into the VEC system.

7.2 Roger Grayson:

The main part of the VEC that Roger worked on was the PDA. He wrote the majority of the software for the PDA including the user interface, event handling, and serial port communication code. Stephen provided the code for the LUT's, which deal with the OBD II error codes. Roger also designed the layout for the user interface on the PDA and created a top-view picture of a car to display in the user interface.

To connect the Rabbit 3000 to the system, Roger desoldered an IDE connector from a motherboard, soldered wires to each header, and wired the IDE connector to a breadboard. He then tested each connection to verify its integrity, and then connected the Rabbit 3000 via an IDE cable.

Roger tested the serial connection on the PDA by connecting it to a computer and transmitting data to and from hyper-terminal. He and Stephen developed code for the Rabbit 3000 to interact with the PDA through the serial port. He then tested the Rabbit 3000 code through HyperTerminal before connecting the PDA to the Rabbit 3000. He also helped Stephen initially debug software for the Rabbit 3000 and test the serial ports. Roger also helped with the integration of the VEC system.

7.3 Stephen Haug

The main accomplishment Stephen made was the ELM322 PCB and related code. Stephen designed and built the ELM322 PCB, after testing the circuit on a breadboard. He also tested the PCB and wrote the code for the Rabbit 3000 to interface to the ELM322.

Stephen also researched the DTCs that could possibly be output from the OBD II system, and wrote code for the PDA to generate a string containing the text of each of the 1585 possible codes. This code takes the DTC from the OBD II output, and converts it to the appropriate string.

Stephen helped test and debug the Rabbit 3000, the interfaces to the Rabbit 3000, and helped develop the Rabbit 3000 code for other components. He and Roger also investigated costates and determined the proper usage of costates.

Stephen also helped integrate the components into the VEC system. Lastly, like all of the members of the VEC team, Stephen helped write many papers. Many, many papers.

7.4 William Wykoff:

William's major contribution to the VEC project is the operation of the Winbond WTS701EF text-to-speech processor. The WTS701EF is only available in a 56 pin TSOP package. This required the fabrication of a printed circuit board in order to use the WTS701 with the VEC project. A photograph of the circuit board and connections is shown in Figure 22. The PCB's sole purpose was to allow the easy connection of the TSOP package to the standard bread boards used in the project. The PCB was fabricated with all pins that could possibly need connected run to a header that would then plug into the breadboards. In this way, none of the functions of the WTS701 were sacrificed, and no extra connections (many of which are not permitted under any circumstance) are allowed. Most external parts required from the WTS701 (a crystal, several capacitors) fit nicely near the PCB board and all components occupy a small footprint on the main project board. William also helped integrate the components into the VEC system.

7.5 Microsoft Project Task Sheet

ID	ℹ	Task Name	Predecessors	Start	Finish	Duration	Est. Hrs.	Resource Names
1		**MCU**		**Tue 12/9/03**	**Sun 2/29/04**	**83 days**		
2	▦	Select and purchase		Tue 12/9/03	Fri 1/16/04	39 days	8	Jeff,Steve
3		**Develop software**	**2**	**Sat 1/17/04**	**Sat 1/31/04**	**15 days**		
4	▦	Sensors		Sat 1/17/04	Sat 1/31/04	15 days	6	Jeff
5	▦	OBD II		Sat 1/17/04	Sat 1/31/04	15 days	14	Steve
6	▦	Winbond Chip		Sat 1/17/04	Sat 1/31/04	15 days	6	William
7	▦	PDA		Sat 1/17/04	Sat 1/31/04	15 days	10	Roger,Steve,Jeff,William
8	▦	Computations		Sat 1/17/04	Sat 1/31/04	15 days	6	Roger
9	▦	Breadboard and test	3	Sun 2/1/04	Mon 2/9/04	9 days	13	William,Jeff,Steve,Roger
10	▦	Debug	9	Tue 2/10/04	Sun 2/29/04	20 days	32	Jeff,Steve,Roger,William
11		**OBD II**		**Mon 12/8/03**	**Fri 1/23/04**	**47 days**		
12	▦	Purchse ELM chip		Mon 12/8/03	Fri 1/16/04	40 days	2	Steve
13	▦	Breadboard and test	12	Sat 1/17/04	Fri 1/23/04	7 days	8	Steve
14		**Power Supply**		**Fri 1/16/04**	**Thu 1/29/04**	**14 days**	**25**	
15	▦	Design circuit		Fri 1/16/04	Thu 1/22/04	7 days	3	Roger
16	▦	Breadboard and test	15	Fri 1/23/04	Thu 1/29/04	7 days	6	Roger
17		**PDA**		**Mon 12/8/03**	**Thu 2/19/04**	**74 days**		
18	▦	Select PDA		Mon 12/8/03	Fri 1/16/04	40 days	5	Jeff,William
19	▦	Test serial connection	18	Sat 1/17/04	Fri 1/23/04	7 days	10	Jeff,William
20		**Develop software**	**18**	**Sat 1/17/04**	**Tue 2/10/04**	**25 days**		
21		User interface		Sat 1/17/04	Tue 2/10/04	25 days	25	Roger,William
22		Data Logging		Sat 1/17/04	Tue 2/10/04	25 days	25	Roger,Steve
23	▦	Display		Sat 1/17/04	Tue 2/10/04	25 days	25	Roger,Jeff
24		Test and debug	20	Wed 2/11/04	Thu 2/19/04	9 days	17	Roger,William
25		**Voice Synthesis**		**Mon 12/8/03**	**Fri 2/20/04**	**75 days**		
26	▦	Purchase chip and speaker		Mon 12/8/03	Fri 1/16/04	40 days	2	William
27	▦	Breadboard and test	26	Sat 1/17/04	Wed 2/11/04	26 days	14	Jeff,William
28		Debug	27	Thu 2/12/04	Fri 2/20/04	9 days	9	William,Steve
29		**Sensors**		**Mon 12/8/03**	**Tue 2/10/04**	**65 days**		
30	▦	Purchase sensors (steal from Coulston)		Mon 12/8/03	Fri 1/16/04	40 days	2	Jeff,Roger
31		Bradboard and test	30	Sat 1/17/04	Tue 2/10/04	25 days	17	Jeff
32		**Integration**	**1,11,14,17,25,29**	**Mon 3/1/04**	**Wed 3/31/04**	**31 days**		
33	▦	Breadboard and test		Mon 3/1/04	Wed 3/31/04	31 days	100	Roger,Jeff,Steve,William
34		**Mounting**		**Mon 2/2/04**	**Thu 4/1/04**	**60 days**		
35	▦	Design	2,12,18,26,30	Mon 2/2/04	Sun 2/29/04	28 days	25	Jeff,Roger,Steve,William
36	▦	Implement	35	Mon 3/1/04	Thu 4/1/04	32 days	50	William,Steve,Jeff,Roger
37								
38	▦	Team Design Review		Mon 2/23/04	Fri 2/27/04	5 days		
39	▦	Final Presentation to Faculty		Mon 4/19/04	Fri 4/23/04	5 days		
40	▦	Senior Design Conference		Sat 4/24/04	Sat 4/24/04	1 day		
41	▦	Team Member Evaluations		Mon 2/2/04	Mon 2/2/04	1 day		
42	▦	Team Member Evaluations		Mon 3/1/04	Mon 3/1/04	1 day		
43	▦	Team Member Evaluations		Mon 4/5/04	Mon 4/5/04	1 day		
44	▦	Team Member Evaluations		Mon 5/3/04	Mon 5/3/04	1 day		
45								
46		**CSIDC Milestones**		**Fri 2/20/04**	**Tue 6/29/04**	**131 days**		
47	▦	Interim Report Due		Fri 2/20/04	Fri 2/20/04	1 day		
48	▦	Final Project Report Due		Fri 4/23/04	Fri 4/23/04	1 day		
49	▦	World Finals		Sun 6/27/04	Tue 6/29/04	3 days		

Table 20: Microsoft Project Task Sheet

7.6 Microsoft Project GANNT Chart

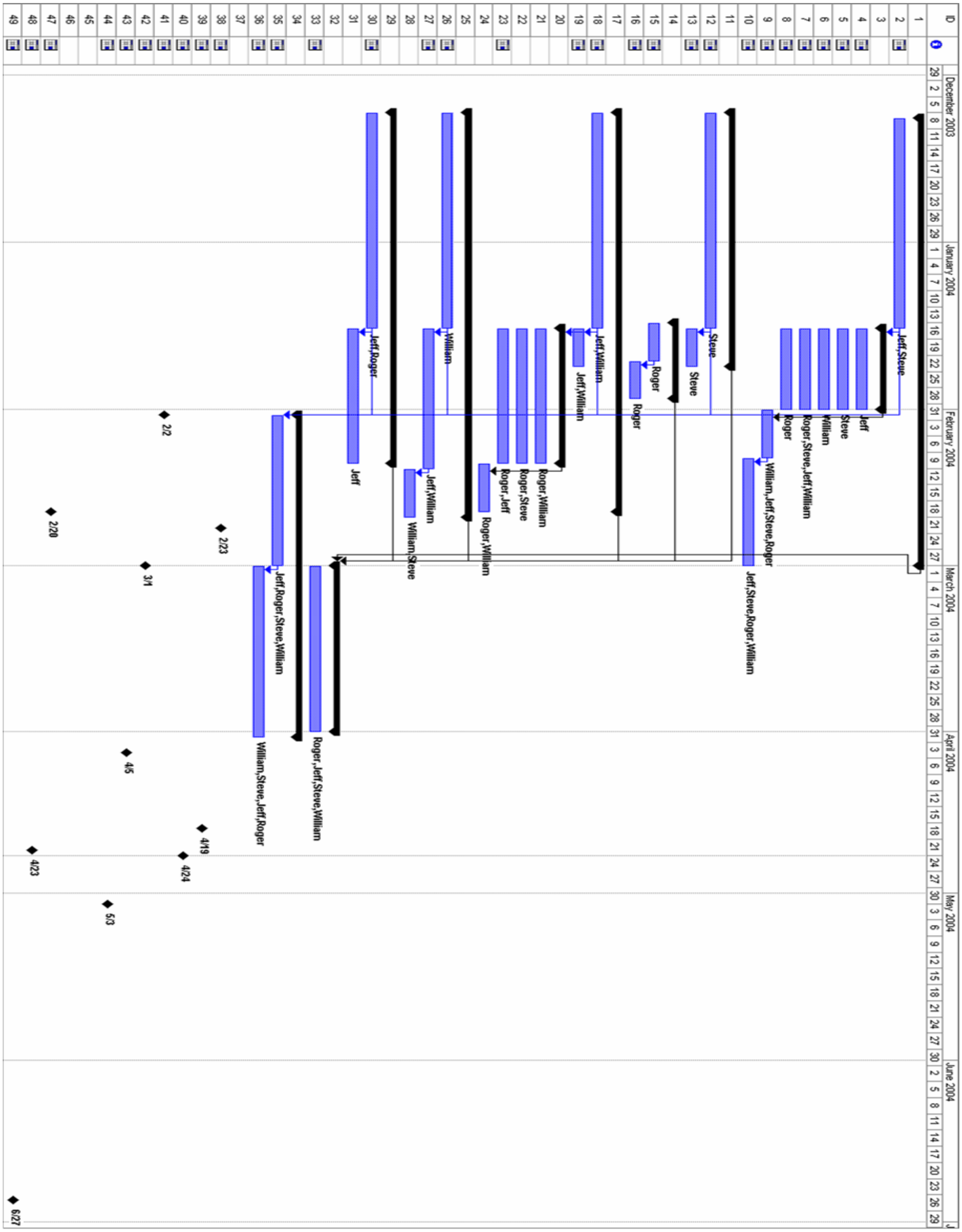


Figure 24: Microsoft Project GANNT Chart

7.7 Project Plan Discussion

The project was executed nearly according to plan. The members listed for each part of the plan in section 7.5 did actually do the work that was scheduled to them. In many cases, we helped each other by being available for reference and moral support even on parts of the project that were not our own.

The timing of the plan was initially delayed since we did not get our project budget until mid-January. For this reason, we could not purchase any parts, which according to the project plan should have been purchased at the beginning of December.

Another reason the timing was not followed exactly was because we needed to find a programming cable for the Rabbit 3000. We initially tried to save $25.00 by making our own, but eventually had to buy a pre-made cable, since we could not find the correct components. Once the cable was obtained, the project proceeded in a timely fashion.

A third timing issue was the Winbond WTS701EF/T PCB fabrication. The instructions for designing a PCB did not include enough information on non-standard parts. It took us a few extra days to locate the correct libraries and settings. Again, once this was resolved, the Winbond WTS701EF/T portion of the project continued as projected.

Each aspect of the project was completed in the allotted amount of time, however, because of the aforementioned delays, the start and finish dates were skewed by approximately 2-3 weeks.

Through this experience, we learned how crucial it is to complete predecessors of the tasks on time, and how one part can delay the entire project. This showed us how important it is to write a project plan out and refer to it during the design process to stay on task. We also learned how difficult it can be to find a time when all members can meet.

8. Development Costs

	Item Cost	Quantity	Total Cost
Polaroid 6500 Range finder	$46.00	5	$230.00
Rabbit RCM3610 Core Board	$39.00	1	$39.00
Rabbit 3000 Programming Cable	$25.00	1	$25.00
Winbond WS701EF/T	$36.30	1	$36.30
3V voltage regulator	$0.96	1	$0.96
Crystal 24.576 MHZ	$0.88	1	$0.88
Crystal 3.579545MHZ	$0.58	1	$0.58
OBDII Cable	$19.95	1	$19.95
ELM322	$12.76	1	$12.76
PDA Cable	$20.00	1	$20.00
Mounting Supplies	-		$30.00
Total			$415.43

Table 21: Development Costs

9. Conclusions

The VEC system has been an exercise in design planning, communication, and finally system integration. From the very beginning, it was necessary to organize such a large project into sections. For the VEC, these sections were very explicit, which made the divisions seem natural. The major sections the VEC was divided into are the OBD II system, the PDA system, the Winbond WTS701EF/T operations, and the Polaroid 6500 sensors operations. All of these were coordinated through the Rabbit 3000. Despite these divisions and the capability of operating all subsystems interdependently, it was important to remember while developing the systems that they would eventually have to interact with each other.

The forced interaction of these systems brings into focus the element of communication that the development of the VEC system requires. Not only were standard methods of communications between the individual systems required to be developed, but communication between the members of the team itself was necessary to orchestrate such large scale interaction. For this reason, the team utilized team meetings on a weekly basis, used file sharing utilities provided by the University (ANGEL), instant messaging, and email.

The integration of the individual systems of the VEC was a task that required many carefully planned steps. In order for this operation to be completed as smoothly as possible, the team chose to integrate the systems in smaller groups first, and then integrate these groups of systems together. For this reason the ELM322, PDA, and Rabbit 3000 were first designed to work together, and the Winbond WS701EF/T was designed to operate with the sensors. This left the integration of these two parts as the main and final task, and minimized possible problems that could have resulted from doing too much at once.

The team believes that the high level of our success rate is a direct consequence of the planning and thought put into the project from the beginning, along with good communication skills that have been developed, improved, and used throughout the project.

Several improvements to the final version of the VEC would help improve the marketability of the system. In order to reach the product stage of development the team desires, the major standard of weatherproofing the system would need to be implemented. This requires the use of different sensors that although available, were not practical in the budget of the team. Secondly, developing support for other forms of display besides the PDA would greatly enhance the flexibility of the system and increase the customer base. This could be achieved by porting the software of the PalmOS to other handheld PC systems such as WindowsCE. Finally, an alternative LCD display and buttons or touch screen option could be developed as an option to the VEC system which would eliminate the need for a PDA while still maintaining most system features.

The VEC system in its present state displays a conceptual model of a system that with some improvements could be turned into a marketable product. Several original design requirements were unable to be met due to budgeting issues, however, the operation of the system is sufficient enough to prove the feasibility of the system as originally designed. With the right amount of time and funding, a finished product could be developed that the team believes would be marketable and profitable.

10. References

10.1 Citations

[1] Bruce Devisser, “Touchscreens in the automotive environment”, *Electronic Products*, < http://www.electronicproducts.com/ ShowPage.asp?SECTION=3700&PRIMID=&FileName=JUNFUJ1.JUN2003 >, accessed Oct. 12, 2003.

[2] Rabbit 3000 Microprocessor User's Manual, Rabbit Semiconductor < http://www.rabbitsemiconductor.com/documentation/docs/manuals/Rabbit3000/ UsersManual/index.htm >, accessed Dec. 23, 2003

[3] Technical Specifications for 6500 Series Sonar Ranging Module, Polaroid OEM Corporation, < http://www.polaroid-oem.com/pdf/6500series.pdf > accessed Feb. 21, 2004

[4] Brian Trappett, PrkNav home page, < http://www.lowtension.com/parknav.htm >

[5] Winbond WTS701EF data sheet, Winbond Electronics Corp, < http://www.winbond.com/e-winbondhtm/partner/PDFResult.asp?Pname=843 >, accessed Oct 15, 2003

[6] Electronic Industries Alliance (EIA), The EIA Home Page, < http://www.eia.org/ >, accessed Oct. 21, 2003.

[7] Max3323 data sheet, Maxim Integrated Products, < http://pdfserv.maxim-ic.com/en/ds/ MAX3322E-MAX3323E.pdf >, accessed Feb. 13, 2004

[8] ELM322 data sheet, Elm Electronics, < http://www.elmelectronics.com/DSheets/ELM322DS.pdf >, accessed Oct 2, 2003

[9] OBDIIJ1962 Replacement AutoTap Cable data sheet, B&B Electronics Mfg Co, < http://www.bb-elec.com/bb-elec/literature/obdiij.pdf >, accessed Jan. 21, 2004

[10] VW info page, Hex Microsystems, < http://www.wwwauto.net.cn/wk/2001-5/imags/wxxzst7.gif >, accessed Feb 21, 2004

[11] Rabbit Semiconductor, Dynamic C Embedded Software, <http://www.rabbitsemiconductor.com/products/dc/DC8/index.shtml>, accessed Dec 25, 2003

[12]Trading Direct Ltd., Veba Recess Sensor < http://www.parkingsensors.co.uk/veba.htm >, accessed Feb 11, 2004

10.2 Non-cited References

10.2.1 Needs Assessment and Background Research References

1. Auto Tap, The OBD-II Home Page, < http://www.obdii.com >. accessed Oct. 2, 2003.

2. ELM Electronics, ELM home page, < http://www.elmelectronics.com >, accessed Oct. 2, 2003.

3. EarthLCD, The EarthLCD Homepage, < http://www.earthlcd.com/touch.htm >, accessed Oct. 12, 2003.

4. Touch Window, "How Touch Screens Work", < http://www.touchwindow.com/miva/merchant.mv?Screen=CTGY&Store_Code=TSI&Category_Code=How+Touch+Screens+Work accessed Oct. 12, 2003.

5. Devisser, Bruce, "Touchscreens in the Automotive Environment", < http://www.fcai.fujitsu.com/touchpanel/RTP_automotive_environments.asp >, accessed Oct. 12, 2003.

6. Winbond WTS701, Winbond Homepage, < http://www.winbond-usa.com/products/tts/datasheets/wts701.pdf >, accessed Oct 15, 2003.

7. Winbond Single Chip Voice Record/Playback Devices, Digikey Homepage, < http://dkc3.digikey.com/PDF/T033/0300.pdf >, accessed Oct 15, 2003.

8. EDP Co. Sona Switch, Ultrasonic Sensors, < http://www.edpcompany.com/pdf/1650.pdf >, accessed Oct. 12, 2003.

9. Motorola Inc., M68HC11 Family, < http://e-www.motorola.com/webapp/sps/site/taxonomy.jsp?nodeId=03t3ZGpnLn84498635 >, accessed Oct. 11, 2003.

10.2.2 Engineering Requirements References

1. Electronic Industries Alliance (EIA), The EIA Home Page, < http://www.eia.org/ >, accessed Oct. 21, 2003.

2. International Organization for Standardization (ISO), The ISO Home Page, < http://www.iso.ch/ >, accessed Oct.21, 2003.

3. National Fire Protection Association (NFSA), The NFSA Home Page, < http://www.nfpa.org/ >, accessed Oct. 21, 2003.

4. Society of Automotive Engineers (SAE), The SAE Home Page, < http://www.sae.org/ >, accessed Oct. 21, 2003.

5. Underwriters Laboratories (UL), The UL Home Page, < http://www.ul.com/ >, accessed Oct. 21, 2003.

6. American National Standards Institute (ANSI), The ANSI Home Page, < http://www.ansi.org >, accessed Oct. 21, 2003

Appendix 1

DEFINITIONS/ACRONYMS

ASCII – American Standard Code for Information Interchange is the numerical representation of characters.
OBD II – On Board Diagnostics, level II
PDA – Personal Digital Assistant
SCI – Serial Communications Interface
VEC – Vision Enhanced Car
ECU - Engine Control Unit
PWM - Pulse Width Modulation