EJAE Receivers: A GPS Receiver System

submitted to:
Professor Joseph Picone
ECE 4532: Senior Design I
Department of Electrical and Computer Engineering
413 Hardy Road, Box 9571
Mississippi State University
Mississippi State, Mississippi 39762

November 21, 2003

prepared by:
Cord Reynolds, Robbie Holt, Jason Pacher, Brian Barnes
Faculty Advisor: Dr. Rose Hu
Industrial Advisor: Matt McNeece, Entergy Services, Inc.
Department of Electrical and Computer Engineering
Mississippi State University
413 Hardy Road, Box 9571
Mississippi State, Mississippi 39762
Tel: 662-325-3912, Fax: 662-325-2298
email: {cjr1, reh2, jap1, beb5}@ece.msstate.edu

EXECUTIVE SUMMARY

GPS systems have found their way into many people's everyday life. Hiking in the woods, sailing on the ocean, fishing in a lake, and driving in a car are a few of the useful applications of GPS receivers today. However, current GPS receivers have problems that make them impractical for certain applications. One problem is the tendency for most GPS receivers that have a good accuracy to also have a high price. Prices on the Internet ranged from a little over $150 to well over $1000. This limits the use of GPS receivers for many simple tasks. Another problem for many current GPS receivers is the complexity of using them. They often have multiple screens and numerous buttons that may confuse the user. This limits the number of people that use GPS systems and limits their use in simple scenarios. The extra features also play a part in raising the cost, which also limits the number of users. Many applications of GPS receivers require nothing more than the user location.

Our GPS receiver will be a handheld unit with a competitive accuracy that is as simple as possible to operate. For our receiver to be competitive, the most important design constraint for the GPS receiver is accuracy. The GPS receiver will provide the user with a position accurate to within 15 meters 99% of the time. This means that the user will know their location within a sphere that has a radius of 15 meters. The movement of the user also affects accuracy. Essentially, the time to calculate a position versus velocity introduces a small error into the receiver. The accuracy will not decrease more than 1 meter per 22 mph. This determination is based on not allowing for more than 100ms for the receiver to manipulate current information and displaying it on the LCD. Since the GPS unit will be a handheld, portable unit, power consumption will also be a very important constraint. The GPS receiver will consume no more than 2 watts of power. After being turned on the GPS receiver will be capable of displaying a user position within 4 minutes. Upon initial display of position, user position will be updated every 10 seconds. This will provide the user with a close to real time measurement of position, while also allowing the user to record the current position if desired.

Two of the key design constraints, simplicity and low cost, compliment each other in implementation. To make the unit as simple as possible there will simply be one user input and one user output. The input will be a power switch and the output will be an LCD that shows longitude and latitude. This creates the simplest user interface possible, as well as helps reduce costs of the GPS receiver. The calculation of user position will rely on four main components; antennae, radio frequency front end, correlator, and an embedded processor. Using an embedded processor will also help minimize cost by avoiding the use of external RAM and ROM. We will use the processor to control the correlator in order to search the incoming satellite signals and determine which satellites we are receiving data from. The correlator will then provide the processor with rough satellite information that our algorithms contained in our processor will use to generate satellite locations, account for errors, and generate user position. We will then output the user position to the LCD. To be able to generate user position in an appropriate amount of time the processor will operate at a frequency of 66 MHz. Power consumption is reduced by use of integrated circuits and an LCD without a backlight. This also will help to keep product size small.

Our design will provide the consumer with a low-cost, accurate, easy to use GPS receiver. Not all GPS applications require a receiver to be complex and difficult use. Some applications require a simple user position output and simple interface. Our receiver will provide the consumer with a low-cost GPS receiver that simply provides location. This receiver will provide a product in the GPS market that is not readily available already. Potential ways to improve the receiver in the future would be to lower the costs wherever possible and add additional user information that would not significantly increase product cost, such as time, speed, and altitude. The only additional hardware that would be needed to display this information would be a larger LCD screen, or use of a user button to select which information to be displayed on the existing LCD. Either alternative would incur little additional costs or design.

TABLE OF CONTENTS

1. PROBLEM 1
2. DESIGN REQUIREMENTS 2
2.1 TECHNICAL DESIGN CONSTRAINTS 2
2.2 PRACTICAL DESIGN CONSTRAINTS 4
3. APPROACH 5
3.1 HARDWARE DESIGN 5
3.2 SOFTWARE DESIGN 11
4. EVALUATION 17
4.1 TEST SPECIFICATIONS 17
4.2 TEST SPECIFICATIONS-SIMULATION 17
4.3 TEST SPECIFICATIONS-HARDWARE 17
4.4 TEST SPECIFICATIONS-SOFTWARE 18
5. SUMMARY AND FUTURE WORK 19
6. ACKNOWLEDGEMENTS 19
7. REFERENCES 20
APPENDIXE A:………21

1. PROBLEM

A man is walking in the woods when he falls down a steep ravine. As he falls, he is knocked unconscious. Upon regaining consciousness, he realizes that he doesn't know where he is. Then, he remembers the GPS receiver and map he has in his back pocket. Armed with these tools, he proceeds to find his way back to his original destination. Consider two friends on a road trip, traveling in the deserts of New Mexico. They run out of gas as soon as they are between two towns over 100 miles apart. Stranded in the basting sun with no one around, they are desperate to get help. With the help of their GPS receiver and a cell phone, they are able to relay their exact position to authorities. The rescuers are able to go directly to their location without wasting any time searching for them. Once again, GPS saves the day. These situations may be drastic, but they may very well occur. Simpler, everyday uses of the GPS system also occur. For instance, normal hiking through the woods, or sailing on an ocean or lake can be made a lot safer and pleasant with the help of a GPS receiver.

Over time, the knowledge of navigation and tools to assist with navigation has grown: from the Chinese using compasses during pre-recorded history for navigation through fog-laden battlefields to command centers using GPS to track troops' positions during the War on Terror[1]. These strides in navigation techniques and tools were the result of people desiring more and more accurate ways of navigation. Recently, in this increasingly technological age, the world has been fascinated with new advanced technology, whether navigation-related or not. With the world wanting to know more and do more, all while making life easier, the GPS receiver falls right into place. Thus, the use of GPS receivers has skyrocketed in the past couple of years. The GPS receiver allows the user to accurately determine the user's position from any where on the planet. Knowing that alone opens up thousands of possibilities for GPS use.

The GPS system was first formed for use by the government as a means to "satisfy the requirements for the military forces to accurately determine their position, velocity, and time in a common reference system, anywhere on or near the Earth on a continuous basis"[1]. The idea of GPS has come far from the 1970's and has been used for countless civilian uses. When the intentional degradation of the civilian signal, known as Selective Availability, ended in 2000, clearly showed the change of the United States government's original intention of GPS being primarily for military applications.

The main feature that this design focuses on is the cost to accuracy ratio. By concentrating on this aspect, the receiver will be practical for many potential applications that were previously impractical. A potential application would be to simply record destinations for someone using a GPS navigation system. Other applications could include using the GPS receiver in conjunction with a system that needs a GPS receiver. However, for this receiver to be used in these applications a good accuracy should be obtained for the cost.

Current GPS receivers have problems that make them impractical for certain applications. One problem is the tendency for most GPS receivers that have a good accuracy to also have a high price. Prices on the Wal-Mart's website and other retailers ranged from a little over $150 to well over $1000. This limits the use of GPS receivers for many simple tasks. Many modern GPS receivers prove to be difficult to operate due to the extra functions they offer. Often, they have multiple screens and numerous buttons that may confuse the user. The extra features seem more like overkill to the person wanting just the basics. This limits the number of people that use GPS and limits its use in simple scenarios. The extra features also play a part in raising the cost, which also limits the number of users.

Our GPS receiver will be able to obtain a good price to performance ratio by providing the user with the most basic information, location. This will limit the price by avoiding the use of additional components such as input buttons and higher resolution LCD screens. By limiting user interface components, it will be possible to concentrate on determining an accurate user location. This will be achieved by accounting for many possible errors in the calculation of user position. Errors that may be accounted for range from relativistic effects to atmospheric refraction. Many of these errors may be accounted for in the software design.

We propose to eliminate the complexity of using our GPS receiver by making it as simple as possible. All a user must do to obtain a location is simply turn the receiver on. After the GPS receiver has obtained the appropriate amount of satellite information the user's location will be displayed on the screen. Up until this point a message will be displayed on the screen telling the user that satellite signals are being acquired. Using the simplest interface possible also compliments designing the GPS receiver to be available at low costs.

GPS receivers may serve a variety of purposes, from leisure usage to life-and-death situations that require immediate and accurate location determination. In leisure situations the extra, unneeded overhead can be more of a nuisance and take away from the fun of navigating. In the life-and-death situations, the time spent sifting through the extra overhead can possibly be the difference between getting out of the situation alive and leaving in a body bag. Our GPS receiver will give users the ability to use a GPS receiver, accurate to within 15 meters, without the unneeded extra overhead, but with life-saving capabilities at a very reasonable price of approximately one hundred dollars.

2. DESIGN REQUIREMENTS

Many varieties of Global Positioning System receivers are available in today's market. Designing a competitive receiver will require a well-balanced use of resources. Our GPS receiver will be a handheld unit with a competitive accuracy that is as simple as possible to operate. For our receiver to be competitive, it must be capable of being manufactured at a relatively low price. To do this, we are simultaneously limiting the input/output devices and creating a simple user interface. The design constraints created by the properties necessary for our GPS receiver are outlined in the following two sections.

2.1 Technical Design Constraints

To make a viable GPS receiver, several technical design constraints must be met. We have decided on five technical design constraints that we will implement in our design of the EJAE GPS Receiver. These constraints are listed in the following Technical Design Constraints table, Table 1, and are then explained in detail immediately after Table 1.

Name	Description
Accuracy	The GPS receiver will provide the user with a position accurate to within 15 meters 99% of the time.
Accuracy vs. User Velocity	The accuracy will not decrease more than 1 meter per 22 mph.
Power Consumption	The GPS receiver will consume no more than 2 watts of power.
Initial Position Calculation Time	After being turned on, the GPS receiver will be capable of displaying a user position within 4 minutes.
LCD Updating Frequency	Upon initial display of position, user position will be updated every 10 seconds.

Table 1: Technical Design Constraints for EJAE GPS Receiver.

2.1.1 Accuracy

The accuracy of the GPS receiver is very important, second only to the safety of the receiver. Two different aspects of the accuracy constitute two of the design constraints. The most simple is that the GPS unit will provide an accuracy of 15 meters at rest. This means that the user will know their location within a sphere that has a radius of 15 meters. The movement of the user also affects accuracy. Essentially, the time to calculate a position versus velocity introduces a small error into the receiver. This occurs because the produced result was for the location of the receiver when it first started the calculation, so any movement incurred during the calculation time is not accounted for. By using the constraint of losing 1 meter's accuracy per 22 mph, the position must be calculated within 100ms.

2.1.2 Power Consumption

Since the GPS unit will be a handheld, portable unit, power consumption will be a very important constraint. The consumption will be limited to a maximum value of 2 watts. By meeting or surpassing this constraint, the unit will be capable of being operated for a useful amount of time on typical alkaline batteries. The unit will also be capable of operating on a practical number of batteries to continue making the receiver viable for portable use.

2.1.3 Initial Position Calculation Time

Any GPS receiver that solely uses the information sent by the satellites to determine user location takes a substantial amount of time to acquire the first user position. This is because the receiver must obtain ephemeris data from the signal broadcast by the satellites. In a worst-case scenario, this will take 30 seconds per satellite, with a minimum of four satellites being required to obtain user position. After initially acquiring the user position this information can be reused and periodically updated to keep accurate information. This GPS unit will be capable of providing the user with a signal within four minutes.

2.1.4 LCD Updating Frequency

To allow the user to be able to see and record positions on the receiver the display will be updated at 10 second intervals. Otherwise, the numbers would change too frequently for them to be read. However, when the display is updated the most recent position calculation must be used.

2.2 Practical Design Constraints

In addition to the required technical constraints, practical constraints must also be taken into consideration. These constraints are listed in Table 2 and are explained in more detail immediately following Table 2.

Type	Name	Description
Safety	Safety	The unit casing and wiring will be designed as to prevent electric shock to the user.
Economic	Cost	The retail price will be under $100.
Simplicity	User Interface	The GPS receiver will allow the user to obtain a location by only having to press a single button, the power switch.
Sustainability	Reliability	This system will be designed to operate over a two-year period without failure. The expected battery life is 3.5 hours and is the only part requiring regular maintenance.
Manufacturability	Size and Weight	The physical dimensions will be 2” high, 3” wide, and 6” long and weigh less than 2 pounds.

Table 2. Practical design constraints for the EJAE GPS Receiver.

2.2.1 Safety

Safety for the user is more important than any feature that the GPS receiver has. Although the receiver will be using typical alkaline batteries, which are not considered dangerous, it *will* be using capacitors, which can store dangerous charges. All components of the receiver will be properly grounded. Safety also makes the packaging of the receiver important, shielding the user from any live contacts within the receiver. Special considerations will be made on the construction and placement of the power switch on the receiver.

2.2.2 Cost

Cost of the GPS receiver is a very important feature of the receiver. For the receiver to be marketable, it will be designed to cost under $100. This will limit the variety and type of components that can be used in the system. Before purchasing any components for the receiver, it must first be validated that the component is the most cost effective solution. The size of the components will also increase the cost as the components get smaller, therefore consideration must be taken not to make the design so small that it will drive up the cost.

2.2.3 User Interface

This GPS receiver is designed to also be marketed on its ease-of-use. By having only the bare essentials, a power switch and a LCD display, the receiver will need no user input. This makes the receiver as simple as possible to operate, while also assisting in keeping the costs of making the receiver lower. To try to obtain a location, all a user must do is turn the receiver on. The unit will then attempt to acquire a signal and produce a result. After that, the receiver automatically updates without further input from the user. It will continue this process, changing the output depending on whether the user goes from a bad signal area to a good signal area, good signal area to a bad signal area, or if the user turns the receiver off.

2.2.4 Reliability

For the design of this GPS receiver to be practical, the unit must be able to operate over a period of five years without failing. By this time, the user should essentially have gotten their money's worth for the services provided. The only part of the system that will require active user maintenance is battery replacement. The GPS receiver will use standard alkaline batteries. Using standard batteries will make it easy for the user to buy replacements and allow the user to operate the unit continuously without having to wait for recharge time. Using alkaline batteries also contributes to keeping the costs low for the receiver.

2.2.5 Size and Weight

Another feature of the GPS receiver is its portability. To be able to maintain this property, certain practical size and weight constraints must be maintained. For a unit to be considered "handheld", the size must not be allowed to get too large. This GPS receiver will be limited to being 2" tall, 3" wide, and 6" long, for 36 cubic inches of space. The weight of the receiver must also be kept in mind to avoid overburdening the user. Two pounds was chosen for the weight limit on the receiver. These constraints also include the size and weight of the batteries and antennae that will be used on the receiver.

3. APPROACH

The following section will go into details about the hardware and software used to create the EJAE receiver. Although more actual software design went into the building of the receiver, there were still considerable hardware considerations. The hardware that has been chosen is more likely to stay the same in the final product, while the software will more than likely have considerable changes on the final implementation.

3.1 Hardware Design

The most challenging part of the hardware design was choosing hardware components that only performed the necessary operations, without having too much additional functionality. Many complete chipset solutions for use in GPS receivers exist in the market already, so this task proved difficult. Figure 1 shows the basic components that are necessary in a GPS receiver. The EJAE receiver has the components broken up into the following categories: antenna with Low Noise Amplifier (LNA), Radio Frequency (RF) front end, correlator, embedded processor, LCD, and miscellaneous hardware. Each category will be gone over in detail as to the function it provides and how it was implemented.

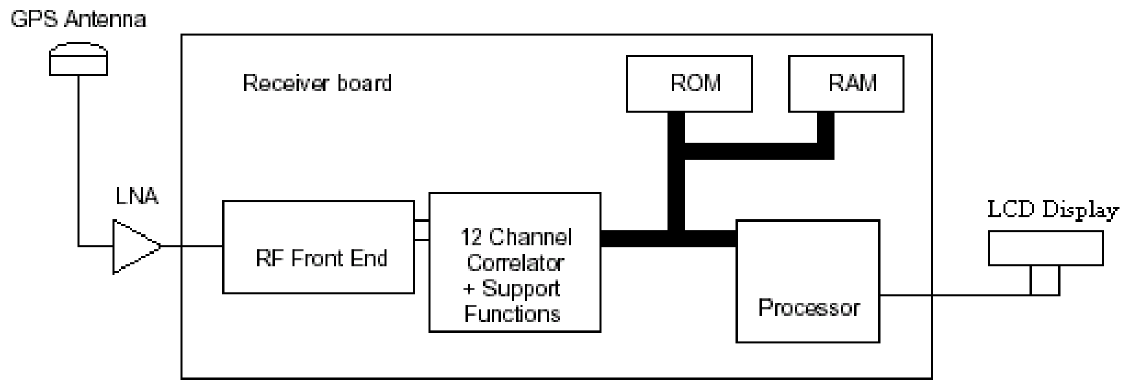

Figure 1 : The physical parts of the GPS receiver.

3.1.1 Antenna with LNA

The GPS system generates two signals for users to determine their location. These are referred to as the L1 and L2 signals. The L2 signal is encoded and is intended primarily for military use [6]. Although some characteristics of the L2 signal can still be used to increase the accuracy of a civilian receiver, this is not practical for a low cost design, such as the EJAE receiver. Because of this, it is only necessary to use an antenna that receives the L1 signal, which is centered at 1575.42 MHz. The signal received is a spread spectrum ranging signal. The signal contains the navigation message, as well as a 1.023 MHz Pseudo Random Noise (PRN) Coarse/Acquisition (C/A) code sequence that provides ranging information [6].

Since the satellites orbit at an altitude of approximately 20,200 km, the signal strength on the surface of the earth is obviously weak [2]. Because of this it is desirable to have a LNA used in conjunction with the antenna to help boost the signal. For simplicity reasons it was determined to use an antenna with a built in LNA for the prototype.

The antenna with a LNA that was chosen for the EJAE receiver prototype is the Micro Pulse 19100 antenna. This antenna has several useful features that were advantageous in use during the creation of the receiver prototype; however it will likely not be used in the final product. This antenna has an integrated 27 dB LNA that runs off of 5V. The two features that made this antenna particularly useful in the prototype were a long coaxial cable and a durable interface. This allowed for testing to be done on a workbench near a window, so the receiver could receive a signal.

The Micro Pulse 19100 antenna has unfortunately proven to have a few characteristics that make it impractical for consideration in a low-cost, handheld receiver. The most alarming problem discovered was that it has a magnetic mount. That, in itself, is cause enough not to use the antenna in a handheld receiver. Other problems encountered were a higher than desired price and a somewhat bulky package.

A patch antenna is currently being considered for use in the final EJAE receiver. A patch antenna is relatively simple in design and is therefore not very expensive. It consists mostly of a groundplane with a square conductor mounted over it. The antenna is actually simple enough to be constructed from scratch. Packaging considerations would probably make this an impractical idea however, so the EJAE receiver will use an existing, mass-produced patch antenna. Price will be the ultimate determination in whether an integrated or separate LNA will be used in conjunction with the patch antenna. The information on which antenna to use will increase during development of the final product, in which cost and packaging are major considerations.

3.1.2 RF Front End and Correlator

The next component in the GPS receiver is the RF Front End. Essentially the RF Front End is an Analog to Digital converter for the GPS receiver. The output of the antenna is a 1575.42 MHz signal input into the RF Front End and transformed into an Intermediate Frequency (IF) digital signal around 4.31MHz. The RF Front End filtering of the IF stages occurs off chip with a discrete 175.42MHz IF filter, 35.42MHz SAW filter, and several on chip filters. Once the signal is stepped down it is input to the Correlator as a sign and magnitude signal. Choosing the Front End was based on a variety of characteristics, such as power consumption, performance, cost, documentation, and interface.

The correlator in a GPS receiver performs several functions. The signal that is delivered to the correlator is a digital signal at an Intermediate Frequency, but it still contains all the separate signals transmitted by the individual satellites. The correlator takes the signal and separates the signal into up to 12 different channels that is able to track 12 separate satellites simultaneously. The correlator can be programmed to use, as many or as few channels as needed in order to save power but still provide needed accuracy. Once the correlator separates the different signals into the selected amount of channels it then steps the 4.31MHz signal down to the final 1.405MHz signal. This final down conversion is needed to keep the correlator in line with the satellite long enough to track it and receive the correct information from it. Once the signal is found we must control the correlator to select a Gold code Psuedo Random Noise (PRN) number from the list stored in the correlator to compare to the signal that it has received. The correlator checks the signal to see if it is the correct satellite if it is then it continues on. However, if the signal does not match it chooses a different PRN number and compares it again. The correlator continues this until it identifies which satellite signal that it has received. Once it has identified which satellite it is looking at the correlator then processes the information about that satellite into raw data that is output to the microprocessor. This occurs for all 12 or however many channels we choose to use. The correlator also has the ability to interrupt the processor to provide it with the most updated satellite information. Choosing the correlator was based on a variety of characteristics, such as power consumption, performance, cost, documentation, compatibility with RF Front End, and interface.

After doing some limited research it was found that RF Front Ends for GPS receivers are plentiful and not difficult to find. As far as power consumption and performance were concerned, nearly every Front End was applicable for use in a low-cost GPS receiver. The biggest problem was finding a Front End receiver that was compatible with an available correlator.

Due to the compatibility problems that were found in almost all front end and correlator combinations, it was eventually determined that it was necessary to find a correlator and front end that were manufactured by the same company. Availability was also an issue, as many companies needed a lead time of at least four weeks, only sold their product to select customers, or only sold in large quantities. Eventually a manufacturer was found that produced a standalone correlator and front end that were available in small quantities. Zarlink Semiconductor produces a front end and correlator available through Insight Electronics in small quantities. Because of this the GP2010 front end and GP2021 correlator were chosen.

Other considerations that led us to choose the GP2010 and GP2021 were compatibility and limited functionality. The correlator must be able to interface with a processor that will perform the mathematical calculations. The GP2021 claims that it is compatible with most 16 and 32 bit processors, which made available a wide variety of microcontrollers and microprocessors to perform the processing of the receiver. Another reason these components were chosen was because of their limited functionality. Many other front ends and correlators could not be purchased without purchasing an entire chipset that performed all receiver functions. Obviously, this would have taken away almost all design in the project.

3.1.3 Embedded Processor

The most difficult part to choose after selecting the correlator and front end was the type of processor to use. There were many different variables to be considered in the selection of a processor. Among these were speed, math capabilities, instruction size, number of programmable pins, architecture type, communication abilities, and onboard memory. Due to the large number of processor manufacturers, this was the part with the most selection to choose from.

The first three properties that had to be determined were the speed, number of programmable pins, and math capabilities desired. Due to the complex nature of the math required to calculate satellite and use positions, it was determined that a processor that could do high precision arithmetic was necessary. For this purpose, a 32-bit processor was decided to be optimal. This way complex math and trigonometry could easily be used in the programming. For the speed of the processor it was determined that a processor with a minimum frequency of 30 MHz would be appropriate, so calculations could be done quickly enough to be considered near real time. The number of programmable pins was also an issue, as a minimum of three pins were necessary, or as many as ten. The processor also needed to have a 16-bit data bus to communicate with the correlator. This began narrowing the number of potentially compatible processors.

Another important factor in choosing a processor was the genre desired. For this project there were two different type to be considered extensively. One is a standalone processor where it is necessary to provide the RAM and ROM used by the processor. The other is what is considered an embedded processor, in which the RAM and ROM are integrated onto the same die as the processor. For the intent of the EJAE receiver the embedded processor was deemed to be more practical. Some of the reasons that made an embedded processor more practical were lower overall cost, fast on-die memory, and less wiring. Although the amount of memory on an embedded processor is more limited, processor with enough were found for the EJAE receiver.

Even after narrowing the processor choice down to a 32-bit, 30+ MHz, embedded processor with several programmable pins, there were still a great number of choices available for the processor. Motorola had several promising embedded processors that could were considered based on their M*CORE. However, cost proved to be the most important factor in deciding which embedded processor to use. The Motorola evaluation board for the M*CORE processors cost at least $500. After doing some research a qualified processor was found for use in the EJAE receiver manufactured by the Atmel Corporation. They also had an evaluation board, the AT91EB40a, which cost only $200, which was much more practical. The processor chosen was the AT91FR40162, the processor on the evaluation board is the AT91FR40008. The only difference between these is that the 40162 has a 2MB flash chip on it. This is provided by the evaluation board for the 40008. What this means is that the programming used on the evaluation board should be very easily ported to the AT91FR40162 when the final product is made.

The AT91FR40162 embedded processor provides very good performance at a reasonable price. The pertinent features are listed in table 3.

AT91FR40162 Feature List	
Operating Frequency	0-70 MHz
Architecture Type	ARM – RISC
Arithmetic Capabilities	32bit ALU and dedicated multiplier
RAM Size	256 KB
ROM Size	2 MB
Programmable Pins	32 pins
Databus	16 bit

Table 3 [8]

The bundled software allowed the processor to be programmed in C, which was essential in being able to do the complex math used in calculating positions. This allowed for the easy use of matrixes, trigonometric functions, square roots, and other complicated math. The math performed on the processor will be gone into more detail in the software sections that explain what each of the different software modules perform.

3.1.4 LCD

The LCD was one of the simpler components to choose. Since the EJAE receiver's marketability is based primarily on the cost, it was necessary to choose a simple, low-cost, easy to use LCD. It was decided that a 2 line by 16 character display would be appropriate. A simple search on Digi-Key produced several compatible results. The MDL(S)-16265 model from Varitronix was eventually chosen.

This model uses a HD44780 control interface that is well documented and widely used. This interface features an 8 bit parallel interface that is fairly easy to control. Since the LCD will be used strictly used by the processor, the LCD will only be written to. This allows the use of one less programmable pin in programming the LCD. Choosing an LCD with this standard proved to be an excellent idea, as there was much support available for the controlling of the LCD, which greatly aided in the design of the software control.

In the future the least expensive LCD will be chosen to be used in the EJAE receiver. This model cost up to two or three times more than other models that had the same functionality and control interface. The LCD will still be a 2x16 LCD with an HD44780 control interface, so replacement of the LCD will require no change in the actual control of the display. Another improvement that is being considered is the addition of a way to light the LCD so that it can be seen at night. Two options that are being considered are either use of a backlight or using an LED to light a conventional display without a backlight. Either implementation would be an on-demand function where the user would have to use a button to activate the lighting. Adding lighting to the display should just add a little additional hardware design. The choice of which is used will be determined by the cost and quality of each.

3.1.5 Miscellaneous Hardware

This section contains parts the EJAE receiver needs in order to achieve the most optimal signals through the GP2010 RF Front End and the GP2021 Correlator. These are the filters and clocks that are used to reduce and eliminate the low noise that might be contained in the signal from the satellite.

3.1.5.1 1575.42 MHz Filter

There is a lot of interfering frequencies that might be received by the active antenna such as cell phone and other equipment in the near by area producing high frequencies. The 1575.42 MHz filter is used to reject the noise frequency that are produced by other sources [5]. The RF filter should reject frequencies that are out-of-band interference signals. The RF filter will be located between the LNA that is located in the antenna and RF input on the RF Front End. There will also need to have some capacitors that are specified by the Front End to block the +5 volts DC that is being supplied to the LNA on the antenna as seen in Figure 2.

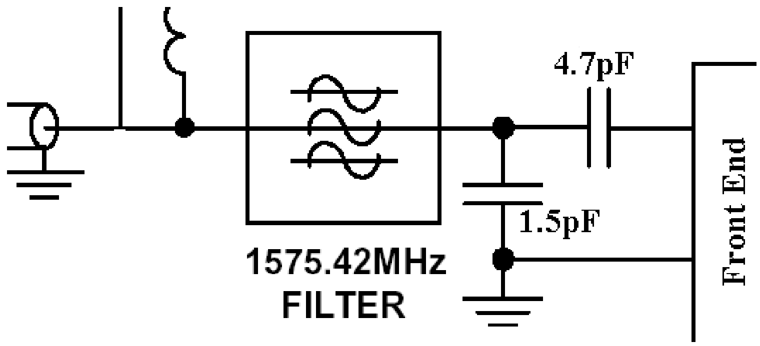

Figure 2: 1575.42 MHz Filter

3.1.5.2 175.42 MHz Filter

This filter is used to reject the intermediate frequency, IF, that comes out of the first stage of the RF Front End. Some of the noise rejection is achieved in the 1575.42 MHz filter but the interference needs to be rejected further. A 104.58 MHz frequency is produced from the first stage within the Front End [5]. Once the filter has rejected the interference noise contained in the signal, the signal is then passed into the second stage of the RF Front End. The noise is rejected by the IF filter to insure that there is still a noise free quality signal. The filter is comprised of capacitor and inductors. There will also need to be a couple of inductors used to pull-up the DC voltage. The circuit can be seen in Figure 3.

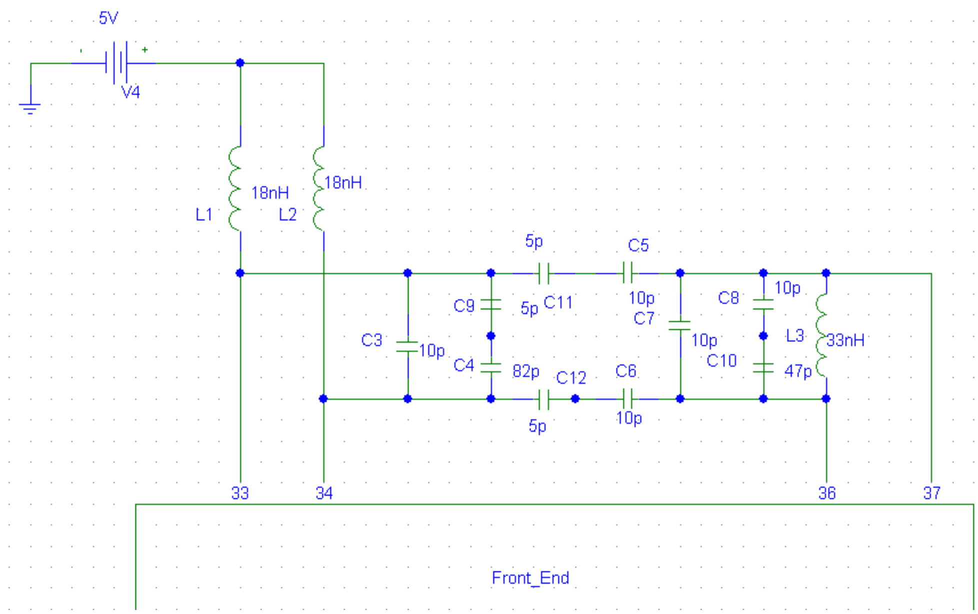

Figure 3: 175.42 MHz Filter

3.1.5.3 35.42 MHz SAW Filter

The SAW filter receives the output signal from the second stage on the RF Front End. We decided to use a filter that has been a well-tested frequency bandwidth is the Dynex DW9255. The filter is used to insure that there is a 2 MHz bandwidth to receiver the Course-Acquisition code signal. The bandwidth has to be maintained to guarantee that the GPS signal has been maintained. The filter rejects a anything that is not contained within that allowed bandwidth.

3.1.5.4 10MHz Crystal Oscillator

There is a built-on phase-locked loop on the RF Front End that controls the local oscillator signals. A 10MHz Crystal Oscillator is connected in with the phase-locked loop to provide frequency stability [5]. Figure 3 is a picture of how the 10 MHz Oscillator is connected to the Front End phase-locked loop.

3.2 Software Design

The software design made up the bulk of our design content. Though we used chips to bring in the signal, this signal was useless unless we controlled how, when, and where to bring in the signal. An overview of the flow of data in the software is shown in Figure 4. The software's role starts with controlling the correlator by telling it when and how to track the satellite signals it has received from the front end. Once it has decided which satellite's information it has, it calculates the raw data and sends it to the processor. Once the data is sent, the software reads the accumulate data and measurement data from the correlator. The software collects the raw data and changes it to ephemeris data. The ephemeris data is then stored for later use.

At this point we have all the necessary data we need and begin making calculations. The first calculation is for corrections on the incoming satellite measurement and time data. As the GPS L1 signal passes through the atmosphere, its traveling speed is affected by its passing through the ionosphere and troposphere. This delay in signal reception contributes the second most influential error to the GPS receiver. The software makes these calculations and stores the new measurement and time values back into the variables. The corrected ephemeris data is now used to calculate the satellite pseudo-range and position. The satellite position uses an iterative approach to a section of the code which determines the true anomaly from the mean anomaly. The resulting satellite positions are then used in the most influential part of the error correction of the GPS receiver. We used geometric dilution of precision (GDOP) to select the four satellites that were farthest apart. Now that we have the coordinates in x, y, z coordinate system, we are able to calculate the user position from the pseudo-orange and the given satellite positions. The function which calculates user position uses an equation that contains four unknowns, which it forms into four linear equations. This function also uses an iterative approach, which attempts to converge the calculated user position to as close as possible to the actual user position. The LCD output control concludes the software part of the EJAE GPS Receiver. The function which outputs to the LCD takes a line number, string to output, and the number of characters to output.

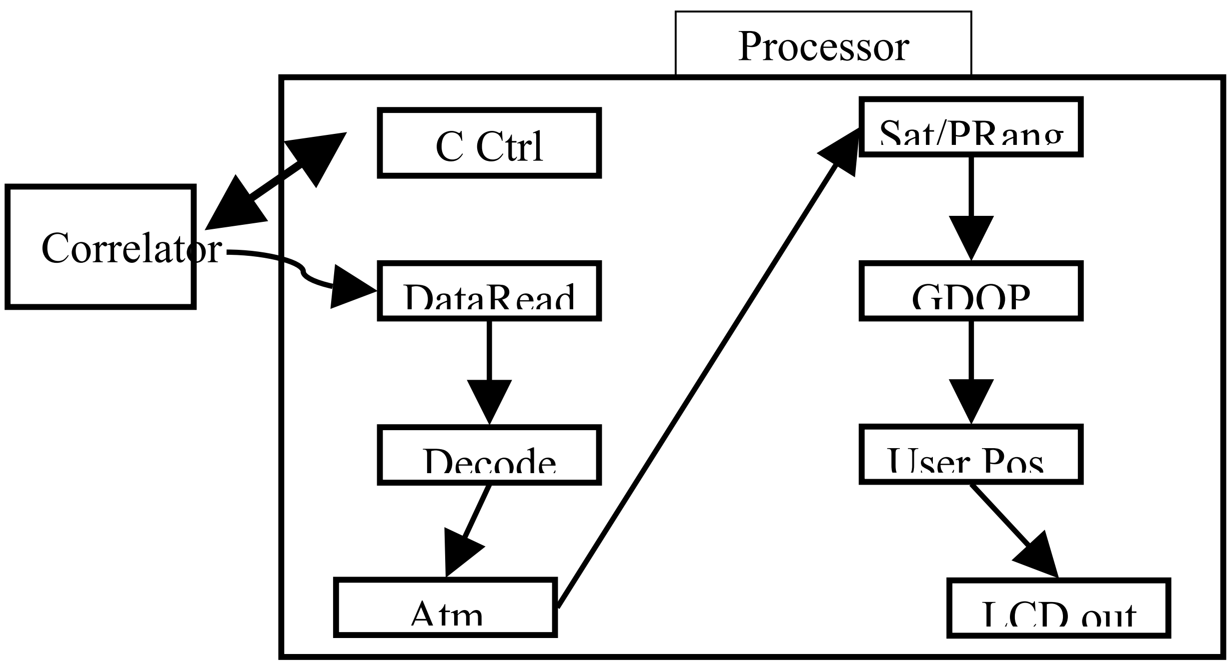

Figure 4: Software Control Flow Chart

The following sub-sections outline the code and equations we used for the different calculations and controls within the software. They are divided into controls, calculations, and corrections.

3.2.1 Correlator Control

The correlator is a very important part of the GPS receiver system. It tracks the satellites, separates them, and calculates the raw data needed from them to send to the processor. The correlator we chose to use in our project is the GP 2021 from Zarlink semiconductors. The GP 2021 has all the functionality we need to track and receive the data, however, it needs to be controlled by our software to function properly. The GP 2021 has the ability to access up to 12 channels to track 12 different satellites at the same time. We have set the correlator to track seven separate satellites. This will allow enough satellites to give an accurate signal within our 15-meter accuracy constraint while consuming less power than a correlator tracking all 12 satellites.

The first thing our software does is control the correlator to perform signal acquisition. There are several registers that our software sets to control the signal acquisition phase of the correlation process. The first two registers that our software has to set are the CHx_CARR_INC_LO and the CHx_CARR_INC_HI. These registers are set with the local oscillator frequency to bring the 2-bit signal from the front end down to the final 1.405MHz signal. The second set of registers CHx_CODE_INC_LO and the CHx_CODE_INC_HI are set with twice the nominal chipping rate of the C/A code about 4.05MHz. This value gives the typical length to be searched. The correlator then needs to know which satellite Gold code to search for. The software does this by programming the CHx_SATCNTL register. To program this register the software set the SOURCESEL bit to select the signal source, set the G2_LOAD bit to select which PRN number to search, and then release the channel to begin the correlation process. The correlator then begins dumping the recorded data into the new accumulated data register and then sets the status bit corresponding to that register high. These registers are mapped in consecutive addresses so that the processor can read them. Then the correlator decides if a GPS singal has been found if it has the processor moves into the pull in algorithm. If a GPS signal has not been found the processor must choose a new PRN number and continue searching until it determines it has found an actual signal. The correlator then moves into tracking phase. The phase alignments and Gold Code are still in error due to the continuous searching. Therefore the signal pull in algorithm uses successive small steps until the error is too small to notice. It then programs the carrier registers and the PRN register with the more accurate values. Once the satellites are aligned and tracked the correlator can begin to receive accurate raw data to send to the processor. The processor will use this raw data to come up with the ephemeris data used to calculate the user position.

At this point we have not yet been able to connect the data bus from the processor to the correlator. Therefore we have not been able to fully test this code to ensure it is working properly. Over the next several days we will have the correlator and processor fully hooked up and will begin testing this part of the software.

3.2.1 Calculations

3.2.1.1 Read Data

The readdata() function's job is to collect the accumulate and measurement data, which also contain the ephemeris data. When the processor activates the write function on the correlator, the correlator and the processor must be synchronized to send and receive data at the correct times. To solve this problem, we used the correlator's clock signal to synchronize the input data. The correlator's clock signal output is tied to one of the processor's input pins. A variable int corrclock keeps track of the current state of the input clock. Since the processor is running at 70 MHz it is able to sample the 20MHz signal from the correlator multiple times before it changes. Using this fact, we are able to detect when the edge-trigger is activated, by having a loop that constantly check the value of the input clock pin against corrclock. Since the correlator is falling edge triggered, we detect that new data is being sent when the value of corrclock is 1 and the new value of the input clock pin is 0. We then enter the conditional statement that actually collects the data. The process then returns to looping to detect a falling edge until all of the data has been received.

After new data being sent is detected, we use an array eph[1600] to collect this data. We chose a size of 1600 because the frames coming from the correlator contain 10 32-bit words, with 5 frames total. The function then perform a loop to go through the 16-bit wide databus. Inside the loop, eph[i] is able to scan each input of the databus to detect the 1's and 0's that indicate that the pins are high or low. These represent the bit value of the incoming 16 bits. The value of the corresponding element in the array is set to either "1" for HIGH or "0" for LOW. During the next falling edge-trigger, the next 16 bits are received into the eph[i] and so on, until the array's entire buffer is full. The function then relinquishes control to the control program which signals the correlator to stop writing and calls the decode function.

3.2.1.2 Decode

Once the data has been read into the array buffer, it is ready to be decoded. The decode() function separates the raw data into its separate ephemeris data variables. Each piece of ephemeris data is separated by parity bits. Our collection of the raw data into one array makes the conversion to separate data variable easy. For example, t_{oe} is located in bits 219-234. The function simply assign eph[219…234] to an array bitephToe[0…15]. Once it is separated, the function converts this bit representation into an actual number by using a loop to multiply the value of each bit-placeholder by 2 raised to the bit-placeholder. For example if the bit values were data[0]=0, data[1]=1, data[2]=1, and data[3]=0, on the first pass through the data[0] would be multiplied by 2^0, second pass data[1]*2^1, third pass data[2]*2^2, last pass data[3]*2^3. Each pass accumulates the total value of all passes. The final number is the double representation of the raw bit data input. This is done for each piece of ephemeris data, with the consideration taken for the length of the bit representation.

3.2.1.3 Satellite / Pseudo-Range

The satposition() function takes an input int satnumber and performs the calculations on that specific satellite number. The satnumber ranges from zero to 6 for a total of 7 satellites, the maximum number of satellites we will track. We used double arrays satx[], saty[], and satz[] to hold the values of the x,y,z positions for each of the satellites. The ephemeris data, which is global data, is used to perform the calculations to determine satellite position. The variables with prefix "eph" are ephemeris data that have been decode in the decode function. The variables with prefix "temp" are temporary variables that hold calculations done on the ephemeris data and other temporary values. The step by step order of calculation is presented as follows

```
tempTc=ephTOW-(ephDdat*.0000002)
tempN=sqrt(conU/(pow(ephsqAs,6)))+ephdeltaN
tempM=ephMo + tempN(tempTc-ephToe)
```

In the following 2 equations, we had the choice to either iterate until we crossed a certain threshold or to iterate between 5 and 10 times (research showed that that range was enough to produce a satisfactory value). We chose to iterate 9 times, but may modify later on to improve speed. We had to pick an initial value for tempE so we could iterate and we chose, upon suggestion from our research, tempM as our initial value.

```
tempEarray[0]= tempM
tempEarray[i+1]=tempM+ephEs*sin(tempEarray[i])
tempE = tempEarray[i]
tempR=(ephsqAs*ephsqAs)(1-ephEs*cos(tempE))
tempV1=acos((cos(tempE)-ephEs)/(1-(ephEs*ephEs)*cos(tempE)))
tempV2=asin((sqrt(1-(ephEs*ephEs))*sin(tempE))/(1-(ephEs)*cos(tempE)))
```

After calculating tempV2, if tempV2 is less than 0, tempV=tempV1*(-1). If tempV=0, tempV equals zero. If tempV2 is greater than zero, tempV=tempV1.

```
tempPhi=tempV+ephW
tempdelPhi=ephCus*sin(2*tempPhi)+ephCuc*cos(2*tempPhi)
tempdelR=ephCrs*sin(2*tempPhi)+ephCrc*cos(2*tempPhi)
```

tempdelI=ephCis*sin(2*tempPhi)+ephCic*cos(2*tempPhi)

tempPhi=tempPhi + tempdelPhi
tempR=tempR + tempdelR
tempdeltaTr=conF*ephEs*ephsqAs*sin(tempE)
tempdeltaT = ephAf0+ephAf1 * (tempTc - ephToc) + ephAf2(tempTc - ephToc)*(temptc - ephToc) + tempdeltaTr-ephTgd
tempT=tempTc-tempdeltaT
ephI=ephI + tempdelI +ephIdot*(tempT-ephToe)
tempOMer=ephOMe + ephOM*(tempT-ephToe) - conOMie * tempT

Now we are able to calculate the position of the current satellite number:
satx[satnumber]=tempR*cos(tempOMer)*cos(tempPhi)-tempR*sin(tempOMer)*cos(ephI)*sin(tempPhi)
saty[satnumber]=tempR*sin(tempOMer)*cos(tempPhi)-tempR*cos(tempOMer)*cos(ephI)*sin(tempPhi)
satz[satnumber]=tempr*sin(ephI)*sin(tempPhi)

3.2.1.4 User Position

The userposition() function picks up after the GDOP function has completed. The known elements are the 4 satellite positions satx[i] saty[i], and satz[i], with i ranging from 0-3, and the distances between satellites and user position satp[i]. The unknown elements are the user point positions userx, usery, and userz, and the user clock bias userb and are double data types. The variables used to store the results from the intermediate calculations done on each satellite are of the double array data type.

The algorithm to calculate user position starts by choosing initial conditions for the user positions points. We chose "0" as the initial condition since we knew that the values for user position could not be negative. Then we select a desired threshold (v) for the error amount. Careful consideration went into the determination of the desired threshold. It had to be low enough to give an accurate answer, but high enough so that the algorithm did not loop infinitely or for a long period of time, trying to calculate the user position. We chose a threshold of 0.1 after numerous test runs that helped us narrow down the range we needed. Next, the algorithm calculates a pseudo-range calcsatp[i] for each satellite based on the equation calcsatp[i] = sqrt(pow((satx[i]-userx),2)+ pow((saty[i]-usery),2) +pow((satz[i]-userz),2)) +userb. It then use those values to get the difference dsatp[m] in the calculated pseudo-range and the actual pseudo-range with the equation dsatp[m]=calcsatp[m]-satp[m]. Now the algorithm differentiates satp[i] in order to get the linear equations of a 4x4 matrix alphsat4[x][y]=(satx[x]-userx)/(calcsatp[x]-userb). During this loop to fill the matrix, satx[x] and userx are replaced by saty[x] and usery when y=1, satz[x] and userz when y=2, and alphsat4[x][y]=1 when y=3. The code then passed the array to the inverse function we created so it could perform the inverse on the matrix. Our code is for a 4x4 matrix but can be easily When the inverse of this matrix is multiplied by the 1x4 matrix dsatp[i], we are able to obtain duserx, dusery, duserz, and duserb. It then uses these values to calculate the error amount dv in the equation dv = sqrt(pow(duserx,2) + pow(dusery,2) + pow(duserz,2) + pow(duserb,2)). The algorithm then compares the desired threshold to the calculated error amount. If dv is less than v, the algorithm halts. If dv is greater than v, we get new initial conditions for userx, usery, userz, and userb, using the equations userx=userx+duserx,…, userb=userb+duserb. The algorithm then repeats itself using the new initial conditions. It continues this process until it crosses the desired threshold. The final values for userx, usery, userz, and userb are the desired user positions.

Now that the user positions are obtained, the function converts the x,y,z coordinates into longitudinal and latitudinal coordinates, then converts these coordinates into a form suitable for input into the lcdPrint function. The equations for converting to (double) latitude and (double) longitude from x, y, z points are longitude = atan(userz /(sqrt(pow(userx,2) + pow(usery,2))) and latitude = atan(usery/userx). To convert the coordinates into a suitable format for the lcdPrint function, we used the sprintf() function to turn the doubles into character arrays. We then padded the input by adding NULL characters and spaces to even the spacing around the numbers.

3.2.2 Corrections

3.2.3 Atmospheric Delay

The atmdelay() function corrects the time values of the ephemeris data to reflect the actual time required for the signal to reach the antenna after delay caused by traveling through the ionosphere and troposphere. This calculation is only for error correction and does not drastically change the accuracy of the receiver; therefore we will not implement this function until 2nd semester.

3.2.4 GDOP

The gdop() function goes through each possible combination of 4 satellite sets to decide the 4 farthest apart satellites. It uses the distance formula d=sqrt((x1-x2)^2 +(y1-y2)^2 +(z1-z2)^2) to get the total distance apart for the 4selected satellites. One all are examined, the 4 farthest apart are selected then used in the user position calculation. This calculation is also only for error correction and does not drastically change the accuracy of the receiver; therefore we will not implement this function until 2nd semester.

3.2.5 LCD Communications

The LCD communications uses the HD44780 standard. Fortunately, this is a well-documented standard with plenty of examples. This interface is an 8-bit parallel communication with a read/write line, instruction/memory line, and enable line. For the purposes of the EJAE receiver, the LCD will only be written to. Due to this, the read/write line is tied to ground so it is always set for writing. The instruction/memory line tells the LCD whether the given 8-bit is a memory access or an instruction. For a memory access or instruction access to occur the enable line must go from low-to-high and high-to-low. The values present on the input pins while the enable line are input.

The HD44780 is relatively easy to understand, therefore the software for controlling the LCD was developed entirely by the EJAE receiver team. What turned out to be the hardest part of managing to accomplish this was essentially a hardware problem. The pinout diagram given for the processor didn’t directly correlate to the way it was laid out on the board. Until the proper pin locations were discovered, it was impossible to test how to change pin values to control the LCD. Once this was discovered and documented by the team, progress on the LCD communications progressed rapidly.

The LCD communications program provides an easy way to display values to the display. The program was coded in C. A call to the program requires three values, the string, the length of the string, and which line to print the string on. By doing this it is possible to write to each individual line, without having to write to both. All ASCII characters can be passed to the program and displayed on the LCD.

4. EVALUATION

4.1 Test Specification

In making sure our GPS receiver meets our specified design constraints we have prepared thorough tests that will validate our claims. Also, all hardware will be tested for proper operation. The tests will test the GPS receiver's accuracy, accuracy versus user position, power consumption, initial position calculation time, LCD updating frequency, and ability to obtain a satellite signal. We will test these using various software, hardware, and simulation-testing tools provided by the department.

Testing for the correlator and front end will be accomplished by using an oscilloscope on various pins. The general values are given in the documentation that should be present on the pins. By comparing these values it will be possible to see if the parts are functioning normally. In depth testing will only be possible when the data bus is connected and can talk to the correlator.

Software testing will be accomplished by running the simulations on a PC and then comparing the obtained values to what is calculated on the GPS receiver. The results will be gotten from the LCD of the GPS receiver, which will have a general module to allow communication with the LCD.

4.2 Test Specification – Simulation

Due to the nature of this project, very little simulation can be used. Most parts will simply have to be hooked up to see if they work properly and then compare them to values given in specification sheets of the products used. This is the case for the antenna, correlator, front end, processor, and lcd.

4.2.1 Power Consumption

The GPS receiver will consume no more than 2 watts of power. We will test the power consumption of the receiver using Pspice. We will add the components and their required voltage, amperage, resistance, capacitance, and inductance in a simulated circuit to test that the combined parts require no more than 2 watts of power.

4.3 Test Specification – Hardware

Hardware testing will be the most critical part of the testing on the receiver. This is the way that most parts can be tested. Measurements of values will be taken by hooking up the components to digital multimeters and oscilloscopes.

4.3.1 LCD Updating Frequency

Upon initial display of position, user position will be updated every 10 seconds. We will test the frequency of displaying the data by giving the receiver sample position coordinates to display and timing how long it takes to update with the correct information.

4.3.2 Obtain Satellite Signals

The antenna must receive the satellite signals and relay them to the RF front end, which converts the signals from analog to digital and send them to the correlator. The correlator separates the signals into separate satellites signals containing the ephemeris data. We will test each component of this section then combine them together and test them.

4.3.3 Antenna

We will test the antenna using the following steps:

1. Connect the antenna to a power source.
2. Connect the antenna to the oscilloscope
3. Measure signal being received
4. Compare to L1 (satellite) signal value of 1.57542GHz

4.3.4 RF Front End

We will test the functionality of the RF Front End using the following steps (after testing of the antenna):

1. Connect the Front End to the antenna
2. Connect both to a power source
3. Connect Front End to the oscilloscope
4. Measure the signal being received from the front end
5. Compare to 4.309MHz signal required to send to correlator

4.3.5 Test Satellite Signal Obtaining Unit

We will test the combined functionality of the pieces, which form the Signal Obtaining Unit (Antenna, RF Front End, and Correlator) using the following steps:

1. Connect antenna to RF Front End
2. Connect RF Front End to Correlator
3. Connect all to a power source
4. Verify digital signal from Correlator

4.3.6 Power Consumption

The GPS receiver will consume at most 2 watts of power. We will test the actual output using a digital multimeter. The expected battery life of the GPS receiver is 3.5 hours. We will test the receiver by turning it on with a fresh set of batteries. We will then time how long it takes to run the battery down.

4.4 Test Specification – Software

Software testing will mainly be accomplished by performing runs on a PC and then comparing values to what is output on the LCD of the receiver. The LCD has an easy to use software module that allows printing to the screen to aid in testing.

4.4.1 Accuracy

The GPS receiver will provide the user with a position accurate to within 15 meters 99% of the time. We will test the accuracy of the EJAE receiver by comparing it with a retail GPS receiver with equal or greater accuracy ratings. We will also find a spot with known latitude and longitude and compare our readings with the known location.

4.4.2 Accuracy vs. User Position

The accuracy will not decrease more than 1 meter per 22 mph. To ensure this, the correlator must separate satellites within a 100ms time frame. We will test to ensure this by letting the processor keep track of the number of times the correlator does its calculations within 5-second window. Dividing 5 seconds by this number will give the time it takes to do one of these calculations.

4.4.3 Initial Position Calculation Time

After being turned on, the GPS receiver will be capable of displaying a user position within 4 minutes. We will test this functionality by turning the GPS receiver on and waiting to see how long it takes for the receiver to display its first reading. This will be done 10 times, each time recording the time it takes and comparing it to the desired <4 minute mark.

5. SUMMARY AND FUTURE WORK

Several components of the EJAE receivers are at a functional state already. Unfortunately, there is one critical part that is not working yet and has delayed having a fully functional prototype. The data bus between the correlator and processor has not been fully implemented. However, the front end and correlator have been hooked up to verify that they propagate a signal. The communications between the LCD has been successfully implemented and tested.

There are several problems that are causing the data bus to not be hooked up yet. One problem is that they have different voltage ratings for the data bus. The correlator has a 5V high value, while the processor has a 3.3V high value. Because of this the data bus has not been hooked up. Currently, a bidirectional voltage translator has been ordered to make safe communications possible. Also, during the course of this semester, there was simply not enough time to accomplish all the necessary design. So not much time has been spent yet on the software necessary to communicate with the correlator on the processor side. The research and design is currently progressing and will be worked on over Christmas break, so the prototype will be functional before the start of Senior Design II.

The LCD is at a fully functional state, with no future work needing to be done to it. The correlator and front end are in a somewhat indeterminate state currently. It has been verified that several signals are propagating through when voltage is applied, however much of it cannot be tested until communication between the correlator and processor is achieved. At this point it will be possible to read from and write to the registers in the correlator. Then in-depth testing of the correlator and front end will be possible.

There are several ideas to enhance the GPS receiver in the future. Lighting will be added to the LCD in some fashion. One idea is to actually add a backlit LCD and another is to use an LED to illuminate the existing LCD. Cost will probably be the deciding factor in which implementation is chosen. Either implementation will be done in an on-demand way, so the battery will not be drained by constantly lighting the display. Other future work will be to increase the accuracy via the software, so no additional costs are incurred. Also, any way to reduce the cost of making the receiver will be examined. This can be done by finding less expensive parts that provide the same functionality or by finding less expensive solutions to implement functions.

6. ACKNOWLEDGEMENTS

We wish to acknowledge Matt McNeese of the Entergy Corporation for his input and advice as a technical advisor. We would also like to thank our advisor Dr. Rose Hu for her hard work and support on this project. We also acknowledge Dr. Picone for this great opportunity to participate in this class and for his continuing support of our project and ideas. We would like to show our appreciation to Jordan Goulder for his advice on our documents. We would also like to thank

Balaji Venkatesan for his help in the lab trying to test our antenna and for all the other help he gave us on our project.

7. REFERENCES

[1] J. Bao-Yen Tsui, *Fundamentals of Global Positioning System Receivers: A Software Approach*. New York, NY: John Wiley and Sons, Inc., 2000.

[2] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, *GPS Theory and Practice.* New York, NY, Springer-Verlag, 2001

[3] Ahmed El-Rabbany, *Introduction to GPS: The Global Positioning System.* Boston, MA: Artech House, INC., 2002

[4] Zarlink Semiconductor Inc., "GP 2021 GPS 12-Channel Correlator," January 2003, http://assets.zarlink.com/DS/zarlink_GP2021_JAN_01.pdf.

[5] Zarlink Semiconductor Inc., "GP 2010 GPS Receiver RF Front End," February 2002, http://assets.zarlink.com/DS/zarlink_GP2010_FEB_02.pdf.

[6] United States Coast Guard, "Global Positioning System Standard Positioning Service Performance Standard," October 2001, http://www.navcen.uscg.gov/gps/geninfo/2001SPSPerformanceStandardFINAL.pdf

[7] William Stallings, Operating Systems, Internals and Design Principles, Prentice-Hall, Inc., Upper Saddle River, New Jersey, USA, 2001.

[8] Atmel Inc., "AT91EB40A Evaluation Board: User Guide", 2002, http://www.atmel.com/dyn/resources/prod_documents/DOC2635.PDF

APPENDIX A: PRODUCT SPECIFICATION

EJAE Receivers

Features

- This device meets FCC E911 accuracy specifications
- Accuracy determined by simultaneously tracking four distinct satellites
- Low cost will not hurt budget
- Longitudinal and latitudinal position automatically displayed on LCD screen
- Light weight for easy transportation
- Low power consumption
- Great device for outdoor survival
- Supplies location in an emergency situation which is crucial for an emergency operator

Product Specifications

Accuracy:
- 67 % at 50 meters
- 95 % at 150 meters

Display:
- Two line LCD screen

Antenna:
- Multipath Reduction Antenna

DC Powered:
- Less than 5 W

Cost:
- Less than $100

Size and Weight:
- 2 x 3 x 6 in.
- 16 oz.