Final Design Report

Ultrasonic/RF Positioning System

Team #1

Team Name: BS

Bret Lauer
(321) 537-5828
egpres@ufl.edu

Steven Medina
(407) 797-1299
sxm457@ufl.ed

Table of Contents

Appendices

List of Tables and Figures

Project Abstract

We are going to use a combination of RF and ultrasonic transceivers to design and build a system capable of locating the position of a mobile receiver inside an enclosed room. The system will consist of remote beacons mounted throughout the room, the object, and a handheld unit to display the location of the object to a user. The object will use an RF transmitter to request an ultrasonic tone from each of the beacons. The object will then measure the time delay from the transmission of the initial RF signal to the reception of the sonar pings. These time delays for each of the remote beacons will be stored by the microprocessor, and the microprocessor will then calculate the receiver's position relative to each of the remote beacons. When the system is set up, it will be calibrated with the location of the remote beacons so that the position relative to the beacons calculated from the time delays can be presented as an actual location within the room.

Features and Objectives

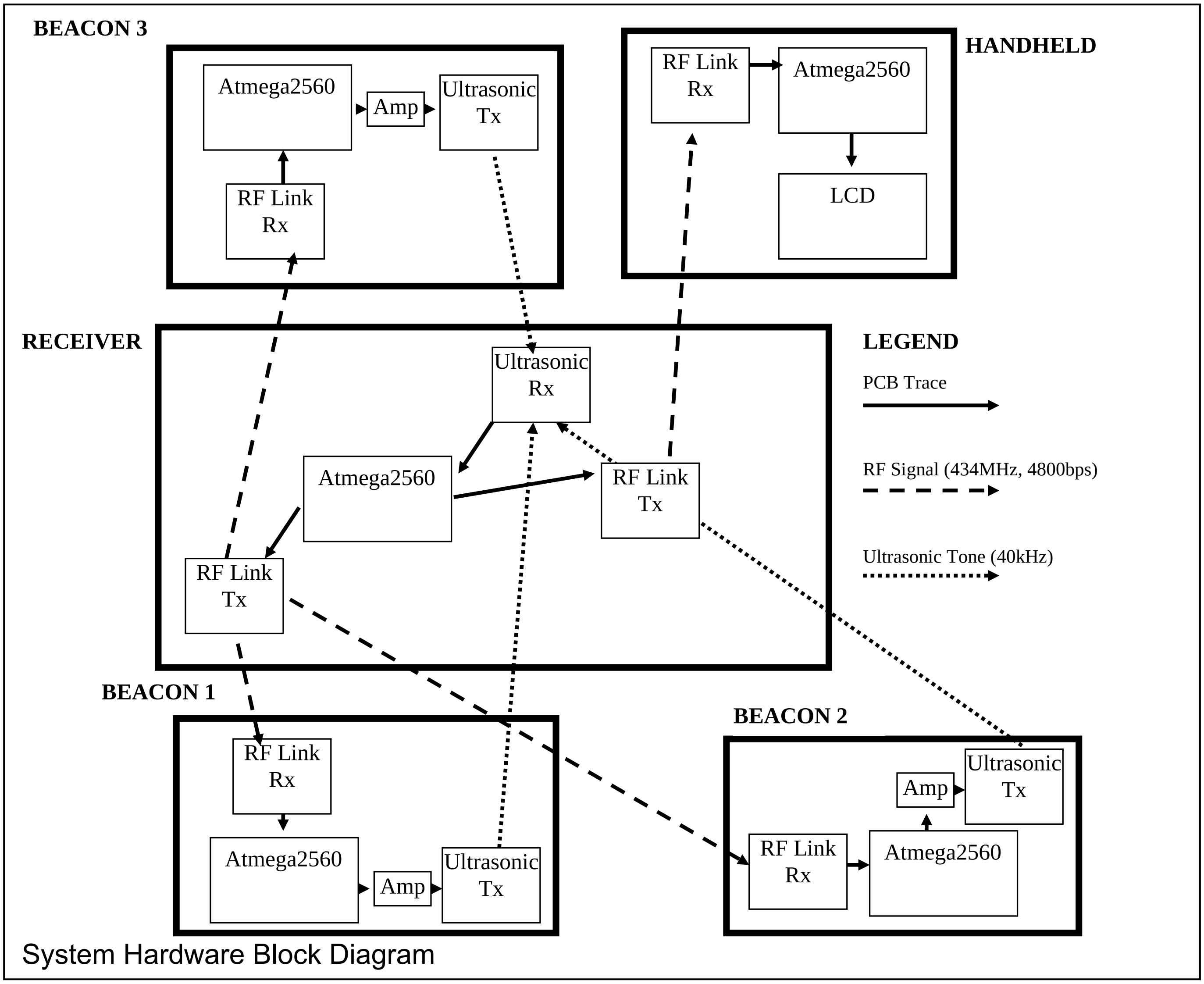
The ultrasonic/RF positioning system can satisfy a need in any application requiring the position of an object. One potential use is to determine the location of a robot within a room. In order to accomplish this overarching goal, the system must meet the following intermediate objectives:

- The object must know the location of each of the remote beacons in the room in order to use them as a reference in the calculation of actual position at the end
- Establish reliable RF communication between the object and each of the remote beacons with the ability of the object to communicate directly to a specific remote beacon when required
- Implement effective reception of ultrasonic tones by the object when they are emitted from the remote beacons
- Allow the object to obtain and store accurate timing information for the duration of time which passes between the transmission of the RF command and the reception of ultrasonic tones from specific remote beacons
- Perform calculations necessary to convert the recorded times into relative distances from specific remote beacons
- Resolve the geometry required to combine the relative distances from each remote beacon with the calibrated positions of each beacon into a specific point in the room which is the location of the object
- Establish reliable RF communication between the object and the handheld LCD device so that position information can be displayed for the user

These objectives can be met with varying degrees of success and reliability, which will determine the performance of the system. This will be displayed by characteristics such as effective range of the system, response time in updating the objects position, and accuracy of the displayed position. Concrete specifications in these performance areas were not set, but it is the goal of the team to obtain optimum performance in each one of these areas in the allotted time.

Technical Concepts/Project Architecture

There are numerous technical concepts that we will need to overcome in order for this system to function properly. Two way communications between the remote beacons and the object is imperative. The object must be able to command a specific remote beacon to emit an ultrasonic tone through the RF link, and the object must also be able to successfully receive that tone when emitted. Due to the effective receive area of the ultrasonic receivers; beacon placement will be very important to ensure that we get maximum coverage of the room. Communications on the beacon side of the system must also be handled reliably. The beacon needs to correctly receive and interpret the RF signal commanding it to emit its tone. A detailed layout of the object, beacon, and handheld device is shown below.

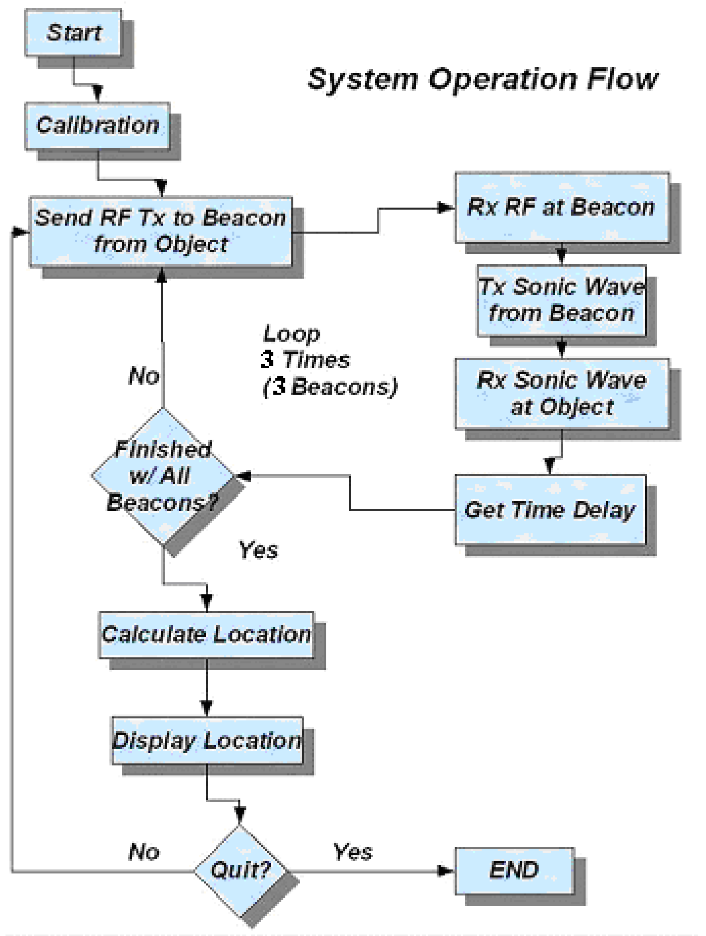


System Hardware Block Diagram

The second issue is receipt of the tones by the object. Since the object does not know where it is at to start, it must provide full coverage to receive an ultrasonic tone from anywhere in the room, from any direction relative to its orientation. This will be overcome through experiment to find the optimal placement for the beacons so that the object can receive the ultrasonic transmissions from anywhere within the room. Since the object's timer is implemented with the input capture function of the 2560, echoes are not an issue. An echo will have a longer distance to travel than that of the straight line path. Since the timer is looking for a rising edge, the first edge it detects must come from the tone which traveled the shortest distance (the direct line one). Any further tones received will be ignored since the input capture timer will have already stopped.

In order to provide the maximum possible range for the object, we must include an amplifier on the object so that the incoming signal can be amplified to the necessary 5 Vpp square wave which is optimal for the input capture. This is accomplished using a Texas Instruments TL072 operational amplifier. We selected this amp because it has a Gain Bandwidth Product (GBW) of 3 MHz. This will provide us with enough gain to get to the second stage whereas an amp with a lower GBW would not give us enough gain at 40 kHz. The second stage of the receive amplifier is the Fairchild Semiconductor CD4093 NAND Schmitt Trigger. The Schmitt Trigger takes the amplified sine wave as its input and outputs a 5 Vpp square wave with high value when the input voltage is above the threshold.

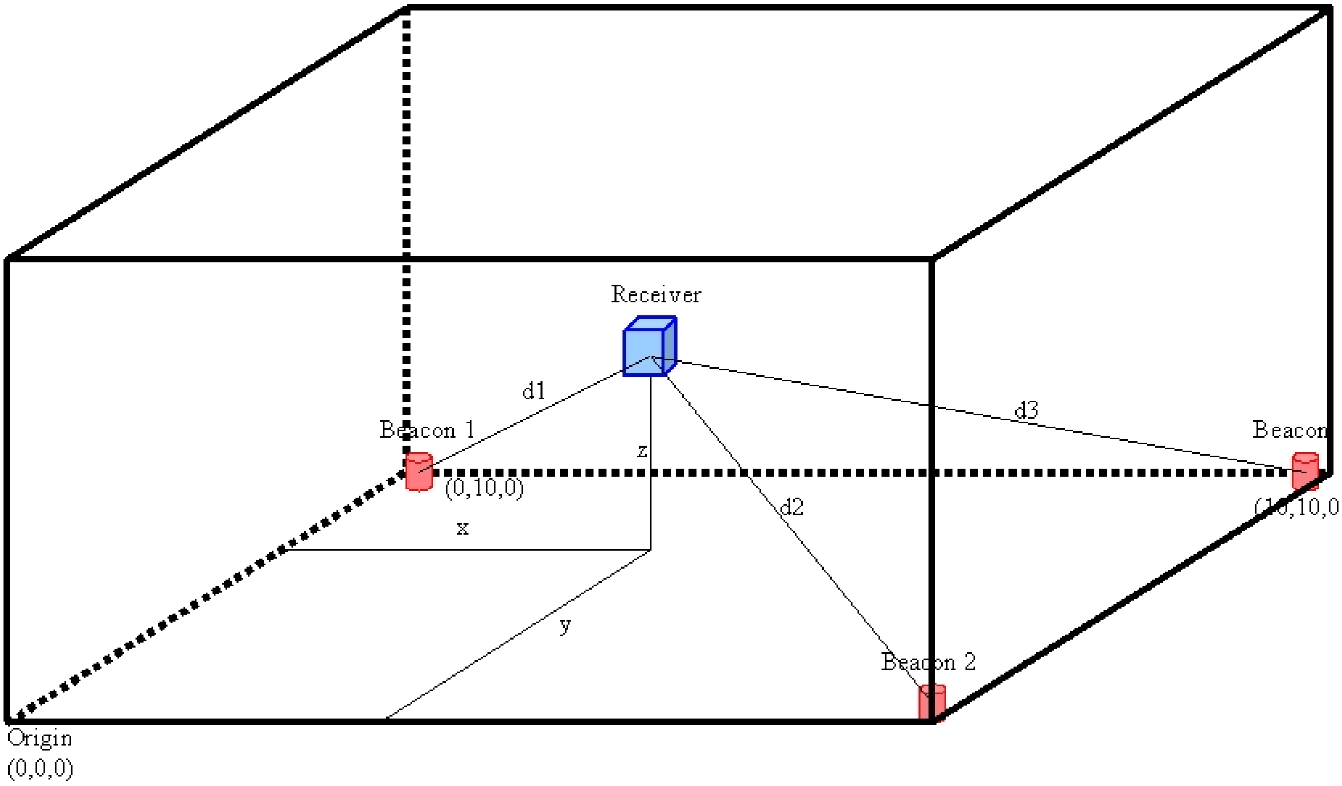
The other area to optimize in order to improve the range of the system is the ultrasonic transmission from the beacon. As it is provided from the 2560, the 40 kHz square wave is only 5 Vpp. The ultrasonic transmitters can take up to 20 Vpp, so if we do not amplify the square wave from the 2560, we are wasting three quarters of the capability provided by the ultrasonic transmitters. When designing this amplifier, it had to have the proper bandwidth to handle a 40 kHz signal, and also needed to provide enough power (high current) to adequately drive the ultrasonic transmitter. The high output current (1.5 A), low output resistance, and ability to run off a single 18 V supply made the TC4428 a perfect choice for this application.



Flowchart of system process

Third, the issue of timing must be successfully resolved. Since all of our location calculations will be based on the timing measurements obtained by the reception of ultrasonic pulses these times must be very accurate. The delay on the remote beacon between reception of RF signal and emission of ultrasonic tone must be kept constant so that it can be accounted for in the calculations. Also, the processor in the object needs to be able to run multiple timers and store values from them. This will allow each of the remote beacons to be timed coming into the ultrasonic receiver on the object. A chart of the entire process is shown on the left.

` Next, the geometry involved in the position calculation must be specified completely for the processor. Even after all of the above challenges have been successfully addressed we will be left with a list of distances from the object to each of the remote beacons in the room. These distances, together with the known positions of each of the remote beacons (coded into the system after the optimal position of the beacons has been experimentally determined) will be enough information to supply a definitive location in the room. Specifying the relationship between the relative distances, beacons positions, and location of receiver is pure geometry, but not necessarily simple. A preliminary room layout for the system (beacons) is shown below. A detailed explanation of the underlying mathematical concepts as well as the equations implemented and methodology to arrive at them is provided in a later section of this report.



Sample layout of room, object in arbitrary location and beacons placed in three corners

Finally, we must present our information in an easy format for a user to read so that when the system is implemented it will present the location of the object inside of the room in which the system is deployed. There are a few ways this can be accomplished. We have elected to go with a simple handheld unit with an LCD. The handheld object will have an RF Link receiver on it, through which communication with the object can be established. This is the simplest choice since it uses the same component already included in the object and we use the same method for communication here as was implemented between the object and the beacons. A microcontroller on the handheld unit, another 2560, will control the RF Link receiver and the LCD.

Power and size were not major limiting considerations for our design because we are implementing a standalone system, and thus were not concerned with any physical constraints arising from integration into an existing platform. That said, the beacons and object needed to be reasonably small. The beacons are designed to sit unobtrusively within the room, and the object needed to be small enough that in the future this system could be applied to an existing application to give location functionality. The handheld device, being that it is handheld, must be small enough for the user to hold. To make things straightforward, we elected to go with 9 V batteries which could be attached to the board by soldering on standard snap connectors purchased from Radio Shack.

Design Choices – Concept and Technology Selection

The driving factor behind the choices made during the design of our project was performance first and foremost. What methods and components would give us better reception, greater range, more accurate calculations; better ease of use, etc. The choices of the hardware components are explained in more detail in the following section below. This section is for design choices pertaining to the overall makeup of the system.

The make up of the system itself was the first and most important design choice addressed during the process. How many beacons are needed? How are results displayed? What “pieces” does the system actually consist of? We knew from a basic algebra standpoint that we are solving for three variables (position in three dimensions), so we would need three equations. This meant that at a minimum we needed three beacons and an actual object for our system. We elected to go with the three beacon minimum in this case since our goal was to make a scaled-down proof-of-concept type system due to the limited time of one semester. Because of this we were not worried about needing to cycle through multiple beacons in order to find three within range that we could receive ultrasonic tones from. The scale we are operating at means we are able to see all three beacons from any reasonable location within our “room”.

The user interface was the next issue tackled as choices made here would have a large impact on how the system actually ended up taking shape. What sort of output are we going to provide? Does the user give any input? In what configuration is the system initially set up?

For output multiple options were considered. We looked at wirelessly sending the position to a computer for display with in a simple UI on screen. We considered sending the measured distances from the beacons to the computer, having the computer solve the position equations, and then display the position. Finally, we explored the possibility of a separate handheld unit with an LCD screen to display the position of the object. This was the solution that we went with for multiple reasons. First and foremost was ease of use for the user. With a handheld device the user can be anywhere doing anything and still receive feedback from the system. With the other two ideas, the user is tied to the computer in order to see results. It was determined that this would hamper the systems applications later. The second area which made the handheld LCD attractive was ease of design and implementation. Using the same microcontroller connected to the same RF Link receivers we were already using to drive and LCD we were already familiar with was much more desirable than tackling the complexities involved with interfacing the object wirelessly to a computer. In this case, ease of use and ease of design both pointed to the same solution, making the choice very easy.

From the standpoint of system set-up and user interface simplicity was the deciding factor. By setting the position of the beacons in advance based on what works best experimentally we have simplified many potential difficulties. If the user placed the beacons in arbitrary locations, there would be no guarantee of full coverage of the room. With the position of the beacons not known in advance, there would need to be a way for their locations to be entered into the system after it is deployed. This would have required user input on the handheld device. When implementing our mathematical solution into the software, we remove 9 variables (the positions in three dimensions of each of the beacons) allowing us to simplify the solutions we place in the code. This significantly reduces the time required to complete the calculations. The advantages from a math standpoint are better explained in the math section below.

The system, based on the choice of specifying beacon locations in advance and hard coding these in, is now quite simple to use. The three beacons are labeled, one, two, and three. The position that each of them must be placed in is specified. Upon placing the beacons in each of their locations and providing power to all the components the system immediately begins displaying position information on the LCD. No further input or set up is required.

Before moving into the task of deciding what specific components would provide the best performance for each piece of the system, we decided to use as many of the same components as possible for similar tasks. For example, if we can find one microcontroller than will accomplish the necessary tasks on ALL of the pieces, that would be preferable to using a different microcontroller on the object form the one on the beacon and a third one on the handheld device. Consistency is assembly as well as the advantage of the economy of buying in bulk was the driving force behind this decision. We are implementing the same scheme to supply power to all the pieces; we use the same microcontroller in all the pieces; the same RF communication components are implemented in the object-beacon link as the object-handheld link.

Hardware Components

Moving down to a lower level than the system level choices detailed in the previous section, following is an explanation behind choices made pertaining to specific hardware components chosen to meet system choices outlined above. The ultrasonic/RF positioning system will require the finely-tuned coordination of multiple components: namely, microcontrollers for the beacon and object, RF transmitters and receivers, ultrasonic transceivers, and other supporting circuitry.

THE MAIN INGREDIENTS

Microcontrollers

For the microcontrollers, we have decided upon the Atmega 2560. The biggest concern was having enough UARTs and timers (for capturing the time delay of the reception of ultrasonic tones). We could have found other comparable microcontrollers, but the Atmels provided the best balance of features and ease-of-use. They can be programmed in C, and there are ample resources available on the web for Atmel microcontrollers. The Atmega 2560 will be used in all of the pieces of our system, the beacons, object, and handheld LCD device. We made this choice since it is easier and cheaper to purchase more of one type of chip and we only need to become familiar with one microcontroller to accomplish all our design objectives.

We have 4 UARTs, which will allow for flexibility in designing the PCB and multiple timers which will be used both for input capture and to generate the square wave at the beacons. Using a 2560 to drive an LCD is straightforward

RF Transmitters/Receivers

To send the instantaneous command to each beacon (and to receive the command) we needed some form of RF transmission and reception. Originally, we were looking at Xbee development kits to handle the RF. The ease-of-use in tandem with the vast array of options made them highly appealing, but to our chagrin we found that they were actually too high tech. Our design relies on our ability to reliably time the transmission of a sound wave. This means that our RF connections have to be as consistent (time-wise) and as fast as possible. The Xbees have multiple built-in controllers that handle the RF connection, but unfortunately, the timing for transmission may be inconsistent due to higher-end error-correction schemes and counters—our need is much simpler.

We found RF Links on sparkfun that were very stripped and economical, providing only what we needed. With a data rate of up to 4800bps and a range of 500ft, the RF link is just a simple RF transmitter receiver pair that connects to a UART. The small size and cheap price will help make multiple beacons more feasible.

Ultrasonic Transducers

Also from sparkfun, we found ultrasonic transducers that were inexpensive (great for scaling to multiple beacons). They offer decent range directly dependent on the applied voltage (up to 20 Vrms). The transducers operate at 40 kHz and provide a wide enough beam width that with smart choice of placement we will get adequate coverage of the room from our beacons.

Handheld Display

The LCD we are using on the handheld device is the Hitachi HD44780U. We selected this device because we were given them for the homework assignment at the beginning of the semester and sample code to run them off of an Atmel microcontroller was readily available. It is perfectly suited to our application since it provides two lines on the display, and we are using it to display the location of the object.

THE ADDITIONS

Amplification

In order to obtain maximum range of our system, we need to provide the ultrasonic transmitter with as close to its maximum accepted voltage as possible. Additionally, we will need to amplify the received signal at the object so that we have a rising edge that we can recognize with the input capture timer. This also raises another issue, as voltage supplied to the amplifiers is going to be much higher that the 5 V required by the microcontrollers and the RF links. Power considerations are addressed below.

To generate the 40 kHz square wave which the ultrasonic transmitter sends, we are using a timer in the 2560s. The square wave generated has a magnitude of 5 V. In order to get this 5 Vpp square wave as close to 20 Vpp as possible, we are using the TC4428 Power MOSFET driver connected to an 18 V supply. The MOSFET driver is ideal for this application because it has a very low output impedance (7 Ω) and a high output current (1.5 A), ensuring that we deliver enough power to the ultrasonic transmitter.

On the receive end we needed an amp that would handle a 40 kHz signal and we also needed to be sure that we were getting a square wave to the input capture of the microcontroller, since it looks for a rising edge. To accomplish this, we used three Texas Instruments TL072 operational amplifiers (GBW = 3 MHz) cascaded and the Fairchild Semiconductor CD4093 NAND Schmitt Trigger. The op amps amplify the incoming signal (which is a sine wave) and the Schmitt Trigger turns it into a square wave. With this configuration we are able to take an input of approximately 50 mVpp and send a 5Vpp square wave to the input capture.

Power

Both the RF Link and the Atmega2560 require 5 V supplies. However, our amplifiers described above require higher voltages. To properly supply the amps, each component of the system (beacons, object, and handheld) will be powered by two 9 V batteries. To bring this down to the 5 V needed for the RF Links at the 2560s we are using the ST Microelectronics L7805 voltage regulator. This is combined with capacitors to give our microcontroller and RF Link a steady 5 V supply. Since the regulators are making a large drop in voltage, heat sinks are required to prevent damage to the IC.

To keep consistency and to make things easy on use during board population, we are going to use the same circuitry and components for the power supply on all of the boards.

Bret Lauer
(321) 537-5828
egpres@ufl.edu

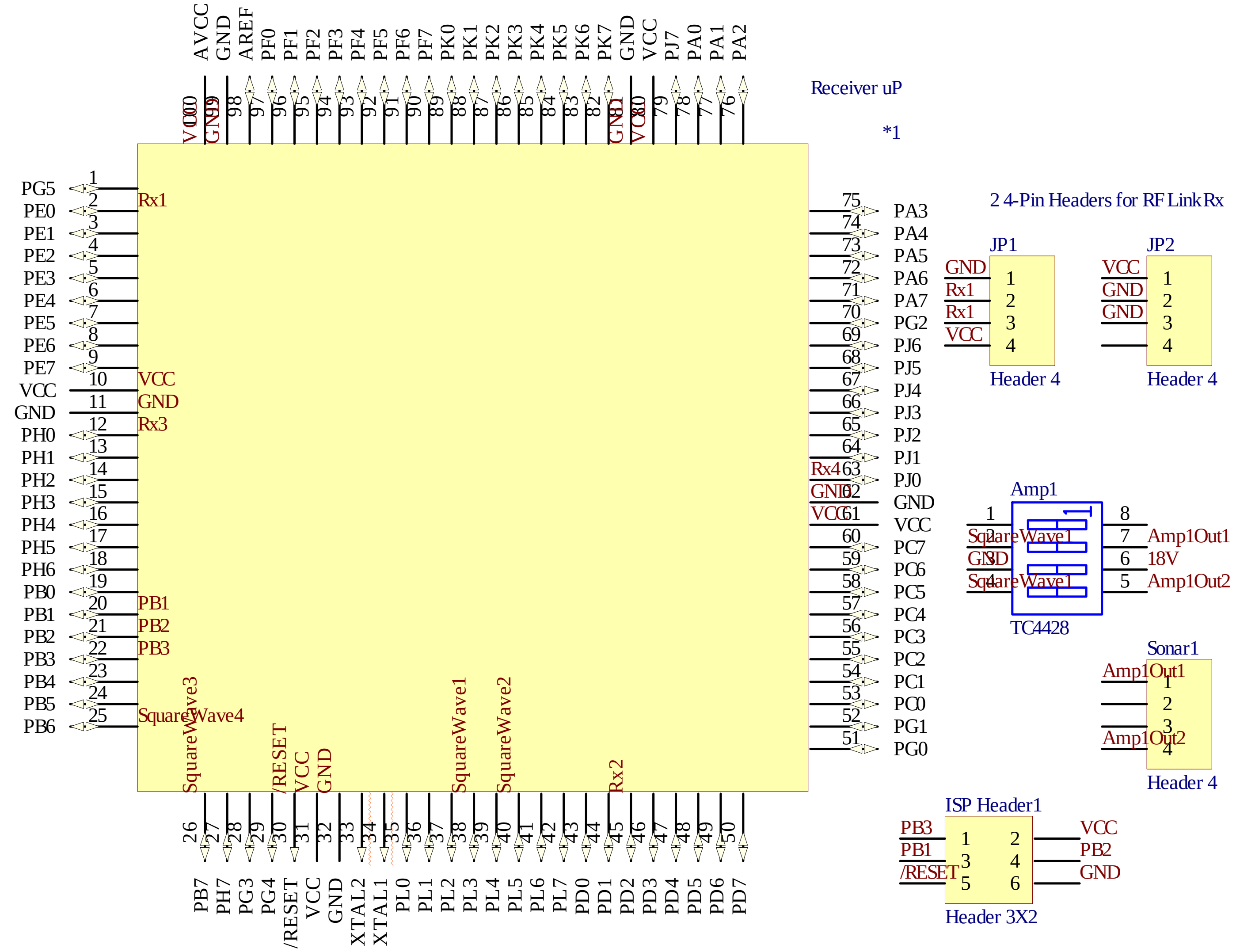
Steven Medina
(407) 797-1299
sxm457@ufl.ed

Our system consists of three basic separate components, each of which requires its own PCB. The three beacons will be on identical boards, another board is required for the object, and another for the handheld device. Using Altium Deisgner, the schematics for each of these pieces of the system are drawn and the PCB is then generated. We then place the components so that the traces on the board are as simple and straightforward as possible attempting to minimize the amount of vias required. Because of the physical characteristics of the 2560s we decided to send gerber files to have our boards milled by a company rather than in the lab here at school. This gave us the additional benefit of solder mask, silk screen, much higher quality, durability, and reliability. Due to the turn around time required for the boards to be made, we knew we only had one shot at making effective working designs and populating them when they came in.

Because we only had time to make one order, and thus got one chance at populating the boards when they came in, we designed multiple redundancies into the boards. Each of the boards for the pieces of the project was designed with three or four redundancies for each of the required components. On the beacon we have four places for an amp, ultrasonic transmitter, RF Link receiver set. On the handheld device, there are three places for RF Link receivers and LCDs. On the object there are four locations for the RF Link transmitter and ultrasonic receiver. In addition to the redundancies on each board, we ordered spare boards with our shipment. The boards were panelized; with four beacon boards placed on one PCB (three are needed). The second board we ordered was panelized with two object boards and two handheld boards (one of each required). This was done simply due to time constraints. Because everything must be finished at the end of the semester, there is no rime to make mistakes and make second order. The redundancies allowed for mistakes in the technician work required to assemble and populate the boards without creating serious issues for completion.

In all cases, due to the size and dimensions of the microcontroller, the trace width on the PCBs is 8 mils, with minimum spacing of 8 mils. We included solder mask on both the top and bottom and silkscreen on the top, since the boards were being sent away for milling.

Since we have three separate individual pieces in the system each one of them was designed individually. Below are the schematics for each of the pieces, their PCB layouts, and explanations for choices made in the design and the layout of the PCB.

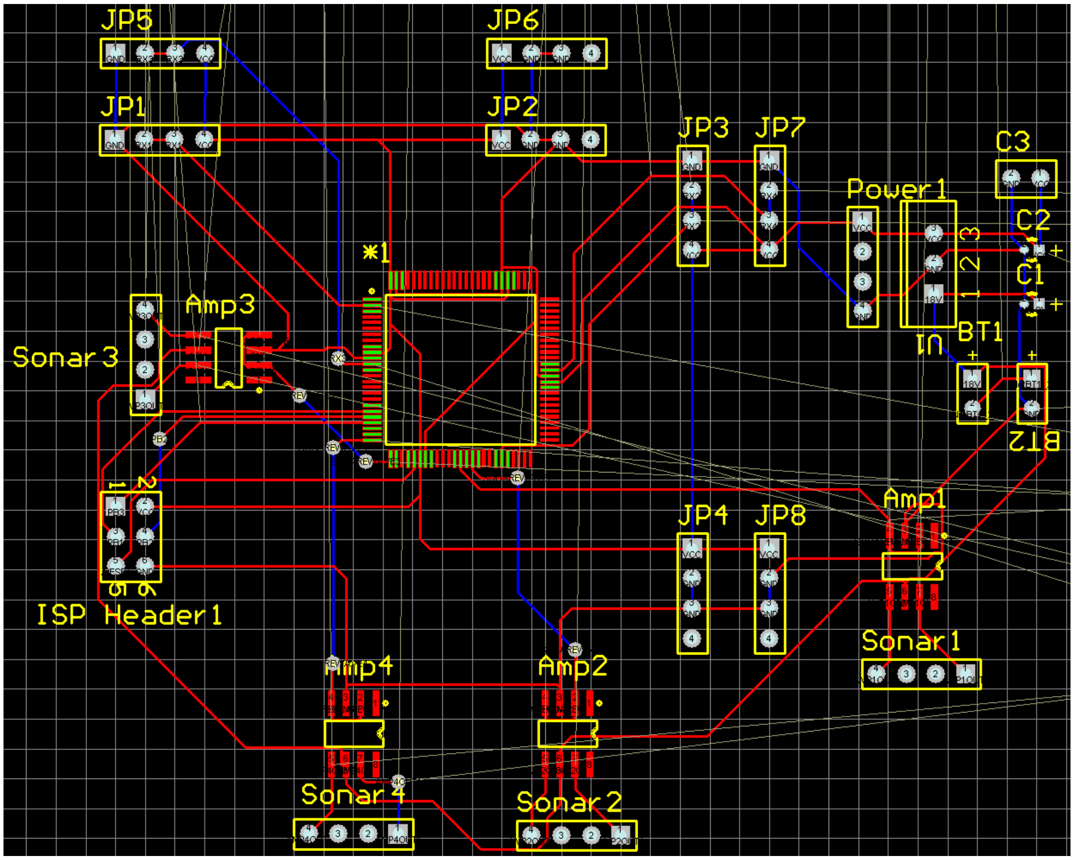


Beacon schematic copied from Protel Project file. The schematic includes the 2560, ISP header, RF Link RX headers, ultrasonic TX header, and amplifier

Our beacons must consist of a microprocessor, RF Link receiver, ultrasonic transmitter, amplifier to drive the ultrasonic transmitter, and an ISP header so that we can program the µP. The RF Link receiver must be connected to a USART, and the square wave is generated on one of the output capture pins of the 2560. Since we have 4 USARTs on our processor we can duplicate all of these traces 4 times, while only having to use one of them on the final board. In the schematic above, all of these traces are indicated on the 2560, but the components are only shown for one "system". The ISP header follows the standard 6-pin configuration. The nodes labeled VCC, GND, and 18 V come from the power supply circuitry, shown in the next section.

Once the PCB is generated we kept all of the components for each of the redundant systems next to each other in order to make for the simplest traces possible on the PCB. The RF Link receivers must be tied to the USART pins on the microcontroller, while the amplifier sits between the 2560 and the ultrasonic transmitter. Keeping the amp next to the ultrasonic transmitter important, as is placing the RF Link receivers next to the USART pins on 2560. Since there were more than 4 possible pins for the output of the square wave from the 2560, we had the flexibility to choose the pins that allowed for the most straightforward PCB design.

Using these guidelines as well as moving the various components once they were placed on the board to make the traces as direct as possible, we arrived at the following design for the PCB of the remote beacons

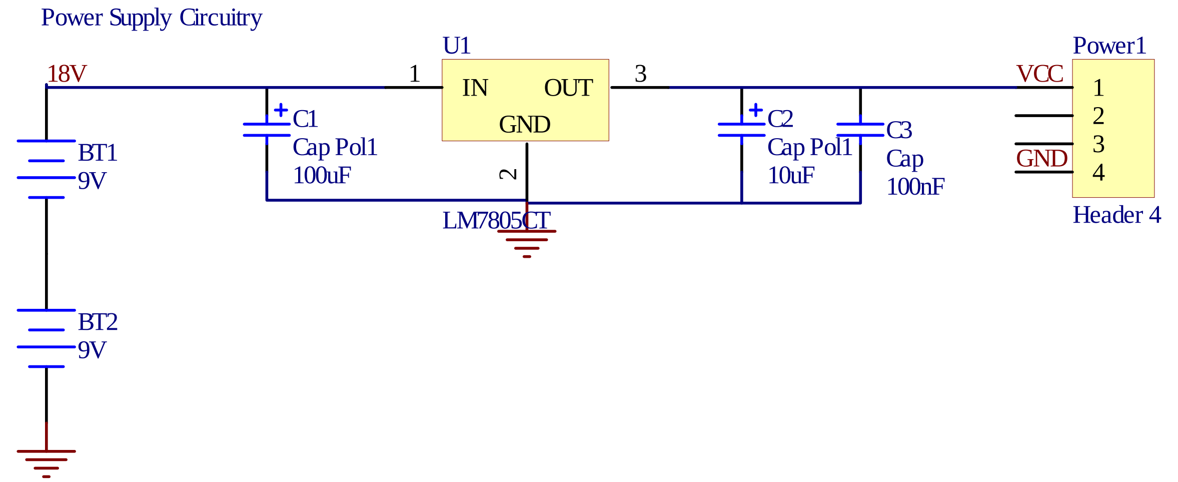


Protel PCB Layout of Beacon

The power supply is at the upper right and the ISP header is in the lower left, as close as possible to the pins that it needed to be connected to. The amps and ultrasonic headers are matched up with each other and placed around the microprocessor near their respective pins. Headers for the RF Link receivers are spaced apart so that they will fit directly onto the boards. For the final board, we have fit four individual beacons onto the same PCB so that we have a spare board. When populating the board, only one RF Link receiver, one amplifier, and one ultrasonic transmitter was attached. The other footprints are left empty.

THE POWER SUPPLY

Except for the amplifiers, all of the components on our boards require 5 V supplies. The amplifiers are given 18 V so that we could squeeze as much performance as possible out of the design. To make things simple, we are using just a single supply on each of the boards (18 V) and using a voltage regulator to drop this down to the 5 V necessary to power the microprocessor, RF Links and the LCD on the handheld device. For the 18 V required for the amplifiers, we can just run a trace directly from the battery connectors. To supply the smoothest 5 V we can to the other components of the board, we employed the standard configuration of capacitors between the supply voltages and ground to eliminate ripple. These included a 100µF and 10µF electrolytic capacitors and a 100nF ceramic capacitor. The schematic for the power supply including batteries, regulator,



Power Supply schematic from Protel project file

caps, and a 4 pin header is shown here.

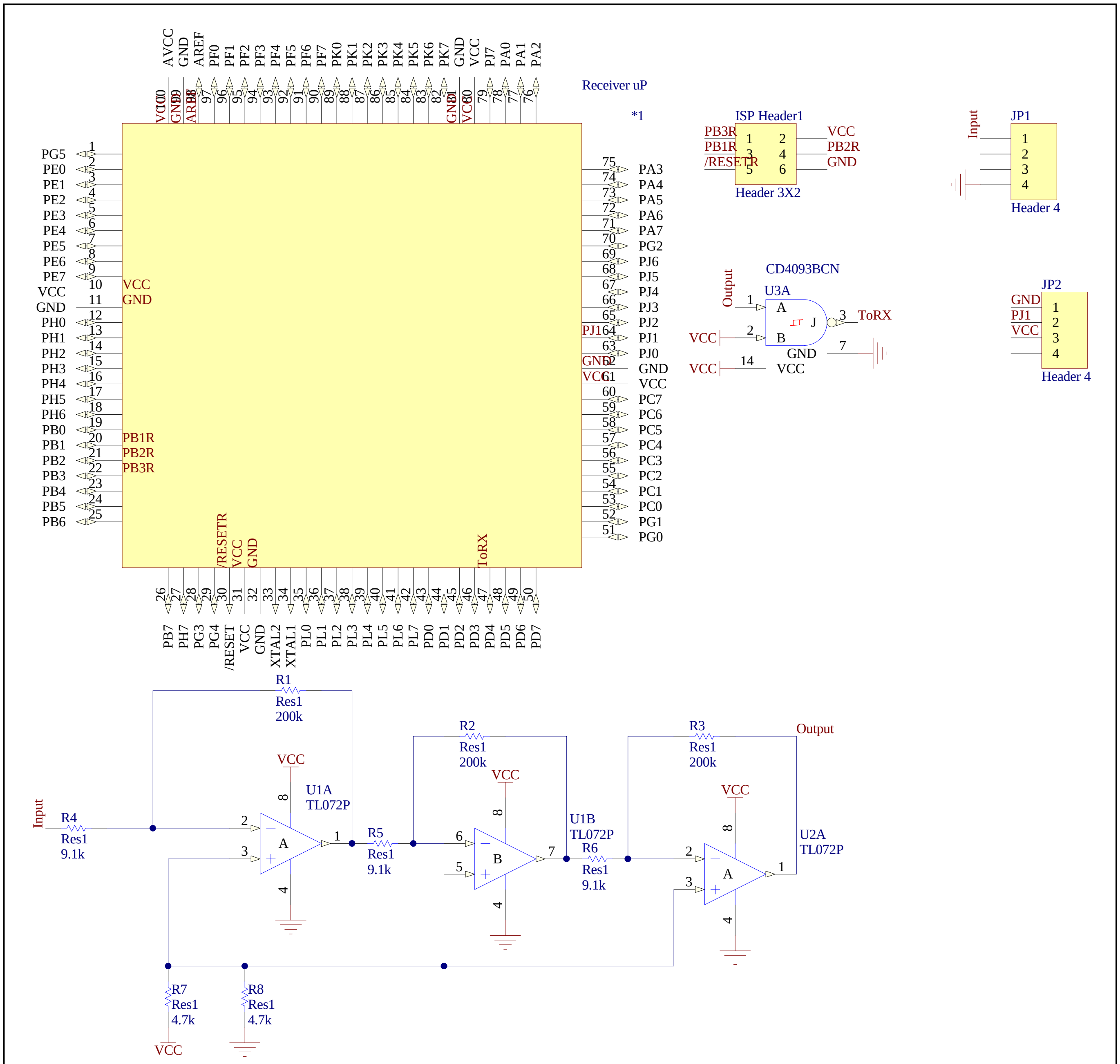
The header is included just to provide another easy access to Vcc after the boards have been milled and populated by just wire-wrapping right onto the header. The schematic for the power supply was taken from the 7805 datasheet as a suggested topology for providing a steady voltage on the node attached to pin 3. This is duplicated on all five PCBs in the system (three beacons, object, handheld device).

THE OBJECT

The object PCB is essentially the opposite of the remote beacon PCB. It includes an RF Link transmitter and an ultrasonic receiver along with the microprocessor, and power supply circuit. The receive amplifier is different from the one on the beacon, since it is dealing with a sine wave from the ultrasonic receiver. The receive amplifier is included on the board with object. The ultrasonic receiver is connected to the input of the amplifier configuration, and the output of the Schmitt trigger is fed to the input capture pin of the microprocessor.

The same method of multiple redundancies was utilized on the object board as the beacon boards. The schematic on the following page shows only the first of each of the redundant systems. The four pin header, JP 1 represents the header where the RF link transmitter will connect, Sonar 1 is the connection to the receive amplifier, which is on a separate board to allow for optimal positioning of the ultrasonic receiver to have adequate visibility of all three beacons. The header which connects to the amplifier feeds the voltage limiter described above, and this output (the voltage across the zener diode) is sent to the input capture pin of the microprocessor. The voltage limiter is very important here because the input capture is looking for the rising edge of a square wave. In order to be sure we can clock the edge correctly we want the receive amplifier to make the peak to peak voltage of the square wave as high as possible. However, the microprocessor can only take a voltage up to Vcc (5 V). To meet both of these requirements, we have set the gain of the receive amplifier very high to be sure we overshoot the 5 V. The Schmitt trigger and the op-amps are all supplied with a Vcc of 5 V so that the square wave that gets to the input capture pin cannot exceed the 5 V that the processor can handle.

For the design and layout of the PCB, the same process was used here as for the beacon PCB. Redundant pieces were kept close to their associated components and each of these “systems” could be placed in the best spot as a group. The four pin header associated with the power supply is utilized here providing an easy connection to tie the receive amplifier to the same power supply as the rest of the components on the object board



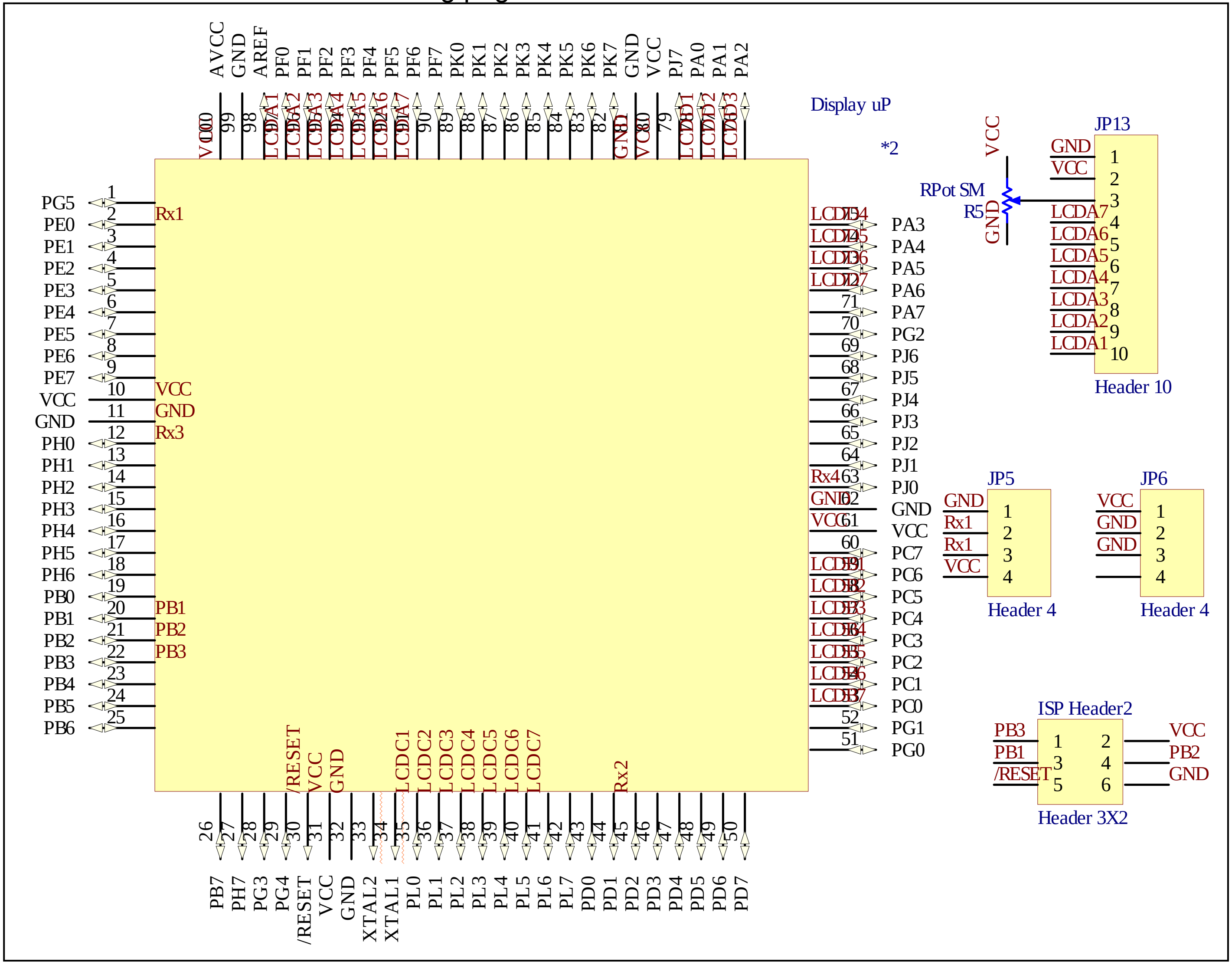
Object schematic copied from Protel project file

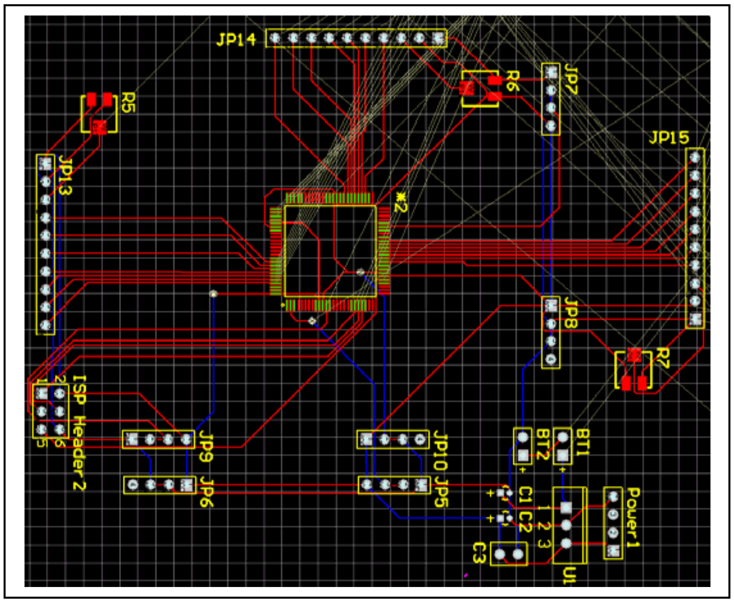
THE HANDHELD DEVICE

The handheld device looks like a beacon, but instead of an ultrasonic transmitter, the microprocessor is connected to an LCD screen. We used the LCD from the initial homework assignment because sample code was readily available, we were familiar with the device, it runs off of a 5 V supply, and it provided all of the functionality required for this application. The schematic, shown below, for the handheld device consists of these three basic components

Again, redundancy is built into this PCB as well. After allowing for each of the four USARTs to potentially be connected to the RF Link receiver, we took four of the remaining ports to allow for flexibility in what would drive the LCD. The LCD is tied to the same Vcc, with the contrast controlled by a surface mounted 10k potentiometer. The configuration of the headers for the RF Link receivers is identical to the one utilized for the beacon boards, with two four pin headers spaced at the measured distance to allow the receiver to be soldered directly onto the PCB.

Due to the amount of traces required to connect the LCD to the microprocessor, we reduced the number of LCD headers and RF Link receivers from the initial design of 4 down to three. This allowed for a much simpler layout of traces on the board as opposed to squeezing in four of each. Even though we sent away to get the boards milled, we wanted to utilize as few vias as possible just to reduce the number of places that we could encounter difficulties. The PCB design is shown below the schematic on the following page.





Previous Page: Protel Schematic of handheld device

Left: Protel PCB Layout of handheld device

THE RECEIVE AMPLIFIER

The receive amplifier is contained on the object board with the RF link transmitter and the ultrasonic receiver. This amplifier consists of two parts. We need to process the incoming ultrasonic signal in two ways. We are receiving a sine wave, and in order for the input capture function to work properly it must see a square wave. In order to accomplish this, the incoming sinusoid must be both amplified to provide adequate magnitude for detection, and it must be converted to a square wave. We are not concerned with the amplitude of the output from this stage, since it is received by the microprocessor through the diode voltage limiter.

The amplifier is implemented first. For this we are using three TL072 operational amplifiers from Texas Instruments. This was selected because it had a Gain Bandwidth Product high enough to provide good gain at 40 kHz. The three amplifiers are cascaded, with each of the three stages set to have a gain of approximately 22. This gives us more than enough amplification to end up with a 5 Vpp signal before we get to the Schmitt trigger. Once the sine wave has been amplified, we use a NAND Schmitt Trigger (Fairchild Semiconductor CD4093) in order to send a square wave to the microprocessor.

The same Vcc that powers the processor supplies power to all of the active components of the amplifier. Limiting these supplies to 5 V guarantees that the signal at the input capture pin does not exceed the 5 V that the processor can accommodate. This amplifier configuration, because it is able to successfully process such low power incoming signals allows us to achieve a wide beam width from the ultrasonic receiver and ample range to cover an entire room. Designing a good receive amplifier provided significantly better improvements to the range and beam width than amplifying the transmitted wave fed to the ultrasonic transmitter through the MOSFET driver.

Positioning Mathematics

After the object has successfully communicated with each of the beacons and recorded the time delay until reception of the ultrasonic pulse there a few important steps to take in order to resolve this data into a fixed position. We start out with values for a time, which are then multiplied by the speed of sound (the velocity of the ultrasonic tone traveling between beacon and object). The value used for the speed of sound we got from NASA's website through an applet which calculates the speed of sound for various conditions (elevation and temperature). The value we are using is for sea level and 80 degree Fahrenheit, which is 670 m/sec.

After the actual distance is calculated, this is converted into a number corresponding to our coordinate system. The dimensions of the room are broken into a Cartesian coordinate system with the origin at (0, 0, 0) and the opposite, upper corner at (10, 10, 10). Each unit corresponds to a distance through the following relation. Dimension/10 = distance per unit. Dividing our distance from the speed of sound calculation by this value of distance per unit gives the distance between the respective beacon and the object in these Cartesian units. This is where the location calculation will be done.

At this stage we have a distance in units from each of the three beacons to the object. This provides three distance equations:

```
d1^2=(x-x1)^2+(y-y1)^2+(z-z1)^2]
d2^2=(x-x2)^2+(y-y2)^2+(z-z2)^2]
d3^2=(x-x3)^2+(y-y3)^2+(z-z3)^2]
```

D1, d2, and d3 are the distances (in units) from the respective beacons to the object. (xi, yi, zi) is the position of the ith beacon within the room. (x, y, z) is the location of the object. This provides us with three equations and three unknowns. Solving the systems for x, y, and z, in terms of the distances will give us a general form of the solution that will allow the position to be calculated by measuring the distances and plugging those values into the solution formulas.

When arriving at a mathematical solution, a variety of methods were investigated. This is a problem that is tackled in many applications, the most well-known of which is the Global Positioning System. There are numerous papers available describing the matrix equations and their solutions which correspond to the GPS equations, as they are known. This is where we began, but there are problems with this method in our context for a variety of reasons. In the GPS case, there are actually 4 unknowns, because there is an unknown time variable that must be resolved between the satellites and the GPS receiver. Because of this, a GPS receiver must be within sight of a minimum of 4 satellites. This is required to calculate the position of the satellites and provide the correct frame of reference for the GPS receiver. The added variable of the fact that the satellites are continuously moving while the receiver resolves its position adds another layer of complexity. On top of this, there is a fudge factor included in commercial GPS systems skewing their accuracy so that it is not better than about 20-30 ft.

All of these contributing factors made us realize the complicated matrix algebra was not only beyond the scope of our project, but very likely beyond the scope of our ability to program the necessary code into our microcontroller. A simpler solution was needed. Since we did not need to account for moving beacons, and unknown time variable, or design a certain amount of inaccuracy a simpler take on the math would be more than sufficient to accomplish our goal. We went back to the three distance equations. The difficulty here is in the squared terms, which prevent us from taking an eighth grade algebra approach to the solution. Care must be taken not to eliminate the second degree from the polynomial, as this could be the solution that we are interested in. Due to the framework within which our system works, solutions to these equations are restricted to x, y, z having values between 0 and 10. We are going to retain both solution sets, during simplification and later select the correct physical one at the end.

The method for solving (x, y, z) in terms of (x1, y1, z1), (x2, y2, z2), (x3, y3, z3), d1, d2, and d3 is explained step by step below:

```
d1^2=x^2-2x*x1+x1^2+y^2-2y*y1+y1^2+z^2-2z*z1+z1^2       (Eq 1)
d2^2=x^2-2x*x2+x2^2+y^2-2y*y2+y2^2+z^2-2z*z2+z2^2       (Eq 2)
d3^2=x^2-2x*x3+x3^2+y^2-2y*y3+y3^2+z^2-2z*z3+z3^2       (Eq 3)
```

Subtracting Eq 1 from Eq 2 and 3 gives:

```
d2^2-d1^2=x2^2-2x*x2+y2^2-2y*y2+z2^2-2z*z2+2x*x1-x1^2+2y*y1-y1^2+2z*z1-
z1^2                                                    (Eq 4)
d3^2-d1^2=x3^2-2x*x3+y3^2-2y*y3+z3^2-2z*z3+2x*x1-x1^2+2y*y1-y1^2+2z*z1-
z1^2                                                    (Eq 5)
```

Moving the constants to the right side of both of these equations and leaving all of the terms with factors x, y, and z on the left gives:

```
2x*x2+2y*y2+2z*z2-2x*x1-2y*y1-2z*z1=x2^2+y2^2+z2^2-x1^2-y1^2-z1^2+d1^2-
d2^2                                                    (Eq 6)
2x*x3+2y*y3+2z*z3-2x*x1-2y*y1-2z*z1=x3^2+y3^2+z3^2-x1^2-y1^2-z1^2+d1^2-
d3^2                                                    (Eq 7)
```

Substituting new variables for the constants on the right hand side gives:

```
2x*x2+2y*y2+2z*z2-2x*x1-2y*y1-2z*z1=B                   (Eq 8)
2x*x3+2y*y3+2z*z3-2x*x1-2y*y1-2z*z1=A                   (Eq 9)
where
```

$$B := x2^2 + y2^2 + z2^2 - x1^2 - y1^2 - z1^2 + d1^2 - d2^2$$

$$A := x3^2 + y3^2 + z3^2 - x1^2 - y1^2 - z1^2 + d1^2 - d3^2$$

Next, factoring out x, y, and z from their common factors and setting Eq 8 and 9 equal to zero gives the following:

```
x(2x2-2x1)+y(2y2-2y1)+z(2z2-2z1)-B=0                    (Eq 10)
x(2x3-2x1)+y(2y3-2y1)+z(2z3-2z1)-A=0                    (Eq 11)
```

Declaring the following variables allows us to write equations 10 and 11 as a 2x4 matrix of linear equations in three variables. Variables followed by the matrix:

$XA := 2x2 - 2x1$ $\quad YA := 2y2 - 2y1$ $\quad ZA := 2z2 - 2z1$

$XB := 2x3 - 2x1$ $\quad YB := 2y3 - 2y1$ $\quad ZB := 2z3 - 2z1$

```
[XA  YA   ZA   B]
[XB  YB   ZB   A]
```

Next, we want to reduce this matrix to reduced row echelon form. This will provide us with solutions for x and y in terms of z. We use the following steps to reduce the matrix to rref. The results of each of the steps are not shown, only the final matrix.

Divide row 1 by xA

row 2 = row 2 – xB * row 1

Divide row 2 by (yB - (xB * yA) / xA)
Simplify the last two terms of row 2
row 1 = row 1 – (yA / xA) * row 2
The resulting matrix is:

$$\begin{bmatrix} 1 & 0 & \left(\frac{ZA}{XA}\right)-\left(\frac{YA}{XA}\right)\cdot\left[\frac{(XA\cdot ZB-XB\cdot ZA)}{(XA\cdot YB-XB\cdot YA)}\right] & \left[\frac{(XA\cdot ZB-XB\cdot ZA)}{(XA\cdot YB-XB\cdot YA)}\right] \\ 0 & 1 & \left(\frac{B}{XA}\right)-\left(\frac{YA}{XA}\right)\cdot\left[\frac{(XA\cdot A-XB\cdot B)}{(XA\cdot YB-XB\cdot YA)}\right] & \left[\frac{(XA\cdot A-XB\cdot B)}{(XA\cdot YB-XB\cdot YA)}\right] \end{bmatrix}$$

From the above matrix, we declare 4 additional variables to simplify writing out the solutions. The new variables are below:

$$\Delta := \left(\frac{ZA}{XA}\right)-\left(\frac{YA}{XA}\right)\left[\frac{(XA\cdot ZB-XB\cdot ZA)}{(XA\cdot YB-XB\cdot YA)}\right] \qquad \Sigma := \left[\frac{(XA\cdot ZB-XB\cdot ZA)}{(XA\cdot YB-XB\cdot YA)}\right]$$

$$\Omega := \left(\frac{B}{XA}\right)-\left(\frac{YA}{XA}\right)\left[\frac{(XA\cdot A-XB\cdot B)}{(XA\cdot YB-XB\cdot YA)}\right] \qquad \Pi := \left[\frac{(XA\cdot A-XB\cdot B)}{(XA\cdot YB-XB\cdot YA)}\right]$$

This leaves us with two equations for x and y in terms of these declared constants and z. We can take these two equations:

$$x := \Omega - \Delta\cdot z \qquad y := \Pi - \Sigma\cdot z$$

And plug them back into Equation 1, which will yield a quadratic equation with z as the only unknown. Everywhere x and y occurs in equation 1 is replaced by the respective equation above. This is the resulting equation:

d1^2=Ω^2-2*Ω*Δ*z+Δ^2*z^2-2x1*Ω+2x1*Δ*z+x1^2+Π^2-2Π*Σ*z+Σ^2*z^2-2*y1*Π+2y1*Σ*z+y1^2+z^2-2z*z1+z1^2 (Eq 12)

Collecting common terms and factoring out z^2, z, and collecting all the constants gives the following equation:

0=z^2(Δ^2+Σ^2+1)+z(-2Ω*Δ+2x1Δ-2Π*Σ+2y1*Σ-2z1)+(Ω^2-2x1*Ω+x1^2+Π^2-2y1*Π+y1^2+z1^2-d1^2)

Declaring three more variables:

$$\alpha := \Delta^2 + \Sigma^2 + 1$$

$$\beta := -2\Omega\cdot\Delta + 2\cdot x1\cdot\Delta - 2\cdot\pi\cdot\Sigma + 2y1\cdot\Sigma - 2z1$$

$$\gamma := \Omega^2 - 2\cdot x1\cdot\Omega + x1^2 + \Pi^2 - 2\cdot y1\cdot\Pi + y1^2 + z1^2 - d1^2$$

Gives the following quadratic

equation, which gives two solutions for z:

$$z := \frac{\left(-\beta + \sqrt{\beta^2 - 4\cdot\alpha\cdot\gamma}\right)}{2\alpha} \qquad z := \frac{\left(-\beta - \sqrt{\beta^2 - 4\cdot\alpha\cdot\gamma}\right)}{2\alpha}$$

Once the value of z is determined, it is placed into the previous equations:

$$x := \Omega - \Delta\cdot z \qquad y := \Pi - \Sigma\cdot z$$

In terms of the physically correct solutions, we are taking the value of z which is between 0 and 10, as this is the one which falls within our defined coordinate system. This value of z is placed into the x and y equations and now all three coordinates are known. This solution to the location equations gives x, y, and z in terms of arbitrary beacon locations and the three measured distances. When we specify locations for the beacons, the equations are dramatically simplified from to expressions which only involve d1, d2, and d3. For this simplification we use Mathematica. The initial location equations are declared in Mathematica using the Dynamic command and the complete solutions are solved. Going back and declaring values for the beacon locations, the solutions are updating reflecting the specified values and simplified, yielding the expressions which we code into the system to solve for the location of the object.

In order to be sure that the object can see all of the beacons from every location within the room, the beacons are placed in the upper corners of the room. This will allow them to transmit downward into the room, and no matter the vertical location of the object, the beacons will always be visible to the ultrasonic receiver, which points up from the object board. In terms of the Cartesian coordinates, beacon 1, 2, and 3 are placed respectively in the following locations:

$$\{x1, y1, z1\} = \{0, 0, 10\}$$
$$\{x2, y2, z2\} = \{10, 0, 10\}$$
$$\{x3, y3, z3\} = \{10, 10, 10\}$$

Mathematica then simplifies the location equations, given these beacon locations, providing the solution for the object location, in terms of the three measured distances to the beacons. The coordinates are below:

$$x = 1/20\ (100+d1^2-d2^2)$$
$$y = 1/20\ (100+d2^2-d3^2)$$
$$z = 1/20\ \left(200-\sqrt{20000\ \ 200\,d1^2\ \ d1^4\ \ 2\,d1^2\,d2^2\ \ 2\,d2^4\ \ 200\,d3^2\ \ 2\,d2^2\,d3^2\ \ d3^4}\right)$$
$$\text{or } 1/20\ \left(200+\sqrt{20000\ \ 200\,d1^2\ \ d1^4\ \ 2\,d1^2\,d2^2\ \ 2\,d2^4\ \ 200\,d3^2\ \ 2\,d2^2\,d3^2\ \ d3^4}\right)$$

The code will calculate both values of z, accepting the one which is between 0 and 10, as this is the value which places the object location within the room. This math has been as simplified as possible in order to minimize the number of clock cycles required by the processor to crunch the numbers, thereby reducing the time required to display the location.

Competition

Smart Robots, Inc is a company which sells autonomous mobile robots in a number of different configurations. They offer a navigator version of their SR-4 robot which has a triangulation base and two remote beacons. This unit can only be obtained as an addition to the professional model. Total cost is approximately $6500, of which $1000 is added on for the navigator functionality which includes the base and two remote beacons. Our system will function independently and be available on its own. Its applications will be limited only by what can receive its output, and our cost will be significantly less than this alternative.

The main application of ultrasonic transmitters and receivers is in ultrasonic range finding. This is different from our application, as the goal with an ultrasonic range finding system is to detect the distance from an object, not the actual position of the object which the transceivers are attached to. This is what provides robots with collision detection, or drives the parking assistance systems in modern cars. Conceptually, there are some similarities in that we are also determining distance. However, we are going a step further in that we have determined the distance from multiple known points and then apply the trilateration equations to these distances and points to give an absolute location.

Division of Labor

The following is a rough division of labor,

Steven	***Bret***
RF Transmission/Reception	Positioning Mathematics
Ultrasonic Transmission/Reception	Amplifiers
Object Atmel programming	Beacon Board
Beacon Atmel programming	Object Board
Handheld Atmel programming	Handheld Board
Prototype Construction	Prototype Construction
Testing/Debug	Testing/Debug

Future Extensions

This system, while effective, has ample room for improvement in future iterations and extensions of the design. The performance limitations were due to time constraints more than anything else, as the project had to be completed from start to finish during a single semester. Once the initial idea and concept phase at the beginning is removed, and time has been factored in to allow for turnaround in the ordering of parts and getting PCBs milled, there is actually fairly little time in which the project must be completed. For this project especially it would have been very easy to bite off more than we could chew by specifying over-ambitious goals for range, response time, and accuracy.

All of these factors turned this project into a proof-of-concept type application of ultrasonic positioning. Our system works on a small scale with limited components, and a very specific feature-set to accomplish the goal intended. There is little flexibility in the set up, and the scale is small. It is in these areas that future efforts would be focused in order to expand the system from where it currently stands to a more robust, long range, customizable location solution for the applications discussed in the introduction.

Through optimizing the components, which would be accomplished through more extensive testing and research, the range of the system in its current configuration could be increased. Time required us to use the first components which provided adequate performance as opposed to optimizing each stage along the way. Better amplifiers on both sides, and the possible addition of filters to provide for cleaner transmit and receive signals would increase the effective range of the system. The limiter currently is the ultrasonic transceivers, and finding a better choice for these would be a very easy way to provide more range.

The second big change would be the addition of a basic user interface to allow for input. This would enable the user to specify where the beacons are freely placed and having the microprocessor adjust the location equations on the fly based on these input coordinates. Along with this, allowing for the freedom to place beacons anywhere but still insure that the object can see them would be to integrate multiple ultrasonic receivers onto the object board. They would be focused in different directions allowing for more complete coverage of a larger room. To accomplish this, the software on the microprocessor would have to be rewritten to accommodate for the processing of multiple input captures and the correct selection of which one is relevant/correct.

The addition of a user interface allowing for input would require the addition of an input device, most likely a basic keyboard on the handheld device. Since there will now be more information on the screen, upgrading to a more advanced LCD allowing for a larger display would also be required for this.

In all cases, we would not have to make a hardware change to the microprocessor since it is very expandable beyond its current use in our system. We have numerous unused pins of each necessary variety, and the boards are not overly crowded. Refining the PCBs to make them smaller, more compact and not having extra components on them would also be a fairly straightforward improvement. During our design the main concern was making sure we could end up with working pieces after a single PCB run. With more time there would not be such a press to get gerber files sent off, and we could afford to minimize the PCB for only required components without the redundancies we included for mistakes. We would also use four layer boards as opposed to two so that everything could be better organized and neater.

As the system is, there is not a fallback in case the object loses sight of one or more beacons. Including more than three beacons would allow for the object to cycle through all of them and then calculate its position as soon as it received responses from any three. The added complexity here, in addition to milling and making more beacons is purely software. The microcontroller would need to keep track of which beacon responded and then select the associated equations and adjust the formulas accordingly based on which responses were received successfully. The added benefit to a system such as this would allow the object to obtain more data to resolve its position multiple times using different sets of three beacons. This would greatly increase the accuracy of the system.

The pieces by themselves are not complex, however designing and implementing a position system is not trivial since there are a lot of factors to account for. Timing is crucial, and with the ultrasonic method, directivity is important. Also, the math involved is not as simplistic as it looks at first glance. Our system had things greatly simplified just by the fact that we set where the beacons were located. Allowing the beacon locations to be free variables in the system adds nine unknowns to the equations, and the microprocessor would then have the added task of simplifying the location equations based on input positions as opposed to having the location of the beacons decided in advance so that the equations can be simplified beforehand. Any of these mentioned areas would provide an interesting and challenging follow-on to this project

Sources and References

www.sparkfun.com	--Source for components of the system. This is where we ordered the RF Links and ultrasonic transceivers
www.digikey.com	--Good source for a wide variety of electrical components. This is where we ordered our processors, IC amps, pots, voltage regulators and heat sinks
www.ti.com	--Datasheet for TL072 and ordered samples of this part for inclusion into the system
www.pcbnet.com	--Company where we ordered our PCBs from as they provided a reasonably fast turnaround and the boards included solder mask and silkscreen at no extra charge
Dr. Eisenstadt’s Lecture Notes	--E Circuits 2 notes were very helpful as a reference for the analog circuit parts of the project
www.atmel.com	--Datasheet for Atmega2560 and sample code for a variety of the tasks we needed the processor to complete
www.AVRfreaks.com	--Another reference for all things Atmega, including sample code, common problems, and troubleshooting advice
www.datasheetcatalog.com	--Datasheets for all of the important components of the project including pin outs, performance characteristics, and limitations

APPENDICES

Appendix A – Software Used

Altium Designer 2004	This is the software we used to design all of our PCBs and schematics. Neither one of us had much familiarity of experience with this software coming into senior design, so we were learning as we went. Taking advantage of the automatic placement and routing features was very beneficial. We had to design our own part for the microprocessor as there was not a pre-made on in any of the libraries and Atmel did not provide one. It was very easy and straightforward to generate the necessary output files to send away for milling as gerber and drill files are generated with a couple clicks.
Mathematica	Like Altium, we had no experience with Mathematica coming into this, and did not even plan on or expect to need to use it. We created a

	workbook file which contained the general form of our position equations, and it would provide the simplest algebraic solutions for them in terms of the distances, when provided with the beacon locations. We used this, after inputting the final beacon locations to determine the equations which were programmed into the microprocessor in the final implementation of the project.
Mathcad	Mathcad is what we started with to verify the arithmetic when we came up with solutions to the distance equations. Since Mathcad was fairly (based on our expertise) limited in what it could do symbolically, we used it to double check our algebra after we were finished. When we solved the system of equations by hand, through multiple substitutions and changes of variable, the easiest way to check this work was to declare all the variables in terms of each other then enter known values with expected solutions into Mathcad and verify the output based on the equations we entered.
PSPICE	Since we were intimately familiar with PSPICE thanks to all of the coursework up to this point all of the simulations of our analog circuitry was done in PSPICE. This is where we verified our voltage limiter, power supply, amplifiers
AVR Studio 4	As the easiest and most common package for developing C code for Atmel chips, there was no reason to look anywhere else for our code. As it provided support and additional help for each specific atmel chip, we could tailor our code and optimize it to work on the Atmega2560. While waiting on breakout boards and final boards we could swap code over and test for performance using the breadboard compatible 324s while using the same development environment.