Territory Tracking and Restriction System

Final Report

Top Dog Technologies

Denise Cuppett

John Kaczmarek

Michael Stewart

Chris Wesp

Department of Computer Science

Texas A&M University

December 11, 2007

Table of Contents

1 Executive summary

The Need, the Goal and the Objectives

The current commercial products available for tracking and deterring pets from specific zones are very limited in functionality. Current indoor deterrents involve a single setup to where a user can deter a pet from only one area, but there is no simple way to document how often the pet enters a zone or violates an off-limit zone without constant supervision. Our intent is to correct this problem by developing a deterrent system that can have multiple off-limit locations and keep track of the time and place of every violation so that the user can be kept up to date with his or her pet's habits. Multiple pets may be tracked by the system and all of the information can be viewed from one simple and easy to understand program. Our goal is to build this new tracking and deterrent system and make it not only cost-affordable, but effortless to use and setup. The Territory Tracking and Restriction System that Top Dog Technologies is developing will have higher quality than any other products currently on the market because there is a need to have a deterrent and tracking system that monitors and documents when pets enter specific zones.

Our team has researched the need of having a deterrent and tracking system and has defined the goal of its new product to create a receiver and transmitter pair that can record the general location of a pet and deter it from the off-limit areas. There has been a lot of research into the implementation of our project because there are some very important objectives that must be met to keep the product competitive in the market. The cost of the product is a very important aspect that must be kept to a minimum. The system must function well in an indoor environment without hindering or harming animals or people. The system must be easy to set up, use and adjust. With these objectives in mind, Top Dog Technologies has decided to use current RF technology to create the Territory Tracking and Restriction System.

Final Design and Implementation

The complete design of our project involves a series of RF receivers and transmitters. However, due to certain restrictions we have created only one transmitter and receiver pair. The transmitter will be the device placed on or around the location a user wishes to make a zone of interest. On it there will be an input that will allow the user to adjust the range of transmission, or deterrent settings. The range can be adjusted on transmitters by controlling the voltage input of a particular transmitter pin. The range can vary from a 1 foot radius to a 10 foot radius. This allows the system to track and potentially deter a pet from different locations ranging from a small area to an entire room. The pet will be equipped with a specialized collar that will have a receiver module on it. Whenever the receiver picks up the transmitted signal, the transmitter's id and a time stamp are stored to built-in memory. If the zone is designated as an off-limit zone, the pet will be deterred until it leaves the zone. When the pet's owner wishes to see the recorded tracking information, he or she can remove the device from the collar and connect it to a computer via USB. An advanced software suite will display the information in a user friendly fashion. The ability to keep track of when and where your pet is violating off-limit zones will be a new technology to the market. The user can look at all the documentation recorded for their pet and follow its progress over time.

Results

Our final result was one of success. Our team accomplished the goal of designing and building a transmitter and receiver that tracks zone breaches and signals if the zone is off limits. Our transmitter is fully built and runs off of AA batteries. It transmits a signal that includes its individual ID as well as the current deterrent settings. The transmitter ID is hard-coded into the transmitter software, but is easily changed at the time of programming. The deterrent settings are set on the fly by the end user with switches corresponding to each pet collar. Another successfully implemented feature is the adjustable transmitter range. The range at which the transmitter broadcasts the signal is fully adjustable using a dial which controls a potentiometer. This allows our system to achieve the result of different zone sizes.

The second successfully implemented component of our project is the receiver for the pet collar. The collar detects incoming signals and monitors for the beginning sequence of our transmitter. Once this is picked up we can process it for both the deterrent setting corresponding to the pet collar and the ID of the transmitter. The receiver module can then store the ID, the deter setting for the pet and a timestamp. All of this data is then used later when the user connects the receiver module to a computer and runs the client software.

Our third completed result was creating a software application that could intuitively show a pet owner the data the system had recorded. The software downloads all of the stored information from the memory in the pet collar module and automatically stores it. A user is also able to view this information through several types of charts and graphs. The most useful of which is a timeline display of the zones and which pets have entered them and when. The software program is also able to apply several filters to the data to allow the user to customize the information they see.

The primary goals of the project were fully accomplished by our team and we are happy with the final results. Given more time there are a few improvements we would have liked to attempt. Primarily, we would like to test and implement a multiple transmitter multiple receiver system. In which we could overcome possible interference and fault recovery issues.

Project Management Summary

Overall our team experience was a very good one. We managed out time well getting benchmarks accomplished quickly so that we would not be rushed to complete something. We had an effective team leader that set goals and delegated responsibility. The team as a whole was able to get along well and utilize individual strengths to achieve results in a limited timeframe. Our team ran into several technical issues throughout the project timeline, such as bad support issues, confusing sample code from manufacturers, and hardware malfunctions. However, proper and effect team-based decision making was used to overcome these challenges and in the end we succeeded.

2 Project background

Many pet owners have trouble keeping their pets out of specific areas of their house. For instance, they may want to keep their pet off valuable furniture or keep it away from certain rooms and areas. When pet owners are absent from their houses, they are unable to control where their pets are allowed to go. The Territory Tracking and Restriction System will allow owners to track and control their pet’s movement. When a pet enters user determined zones, the system will record that the pet entered the zone and deter the pet if it is an off limit zone. This is done by putting RF transmitters in off limit zones, and giving the pet a collar with a RF receiver. When the pet enters a zone, the receiver will receive the RF signal from the transmitter containing the zone id and deterrent settings. The receiver then responds according to the signal received, deterring if necessary. At any time, the owner can use a USB connection to get the data from the receiver to see what zones the pet has been entering and at what times.

2.1 Needs statement

There is a need to have a pet deterrent system that tracks pet movement throughout the house 24/7 by monitoring and documenting when a pet enters certain areas of interest and deters the pet when needed.

2.2 Goal and objectives

The goal our group has set for this project is to create a network of receivers and transmitters that can record the general location of a pet and deter it from the off-limit areas.

Here is a list of objectives that need to be taken into consideration when designing our Pet Deterrent System:

- The prototype system must cost less than $500 to meet the project's budget constraints.
- The commercial system should cost $50 or less per receiver and $30 or less per transmitter to be competitively priced on the market.
- The system must use a power source accessible to the public, such as a battery, and the power source must last at least 1 month without being replaced.
- The system must not harm animals or people.
- The system must function well in a typical indoor environment.
- The collars should be light, less than 1 pound, and comfortable for the pet.
- The system must be easy for the user to set up which is defined as the set up time taking less than 30 minutes.
- The system must be easy to use and adjust; any adult with basic computer knowledge should be efficient with the computer software after 1 week.
- The system should have a variable range that covers an area with a 1 foot radius to an area with a 10 foot radius.
- The system should document the zone and time when a pet violates a restricted location; should also record when a pet enters an allowable zone.
- The recorded information should be displayed to the user in an organized and understandable fashion.

2.3 Design constraints and feasibility

The economic constraints of this system fall into two groups. First the prototype of the deterrent system should cost less than $500. If the development of the project goes over $500 then the project will go over budget. Secondly, this system is being designed with the intent of being a consumer product. The final system needs to be priced competitively. In order for the system to be affordable, the receiver should cost $50 or less, and the transmitter should cost $30 or less. Another commercial economic constraint is the power for the system. The power source needs to be something that is easily accessible and cheap to the consumer. Also, the power source for the system needs to be efficient, so that it does not have to be replaced often.

The physical constraints on the system are in the collar/receiver and transmitter design. The receiver needs to be lightweight (preferably less than one pound) and not hurt the pet. In addition to being lightweight, the receiver needs to be relatively small; a bulky receiver could hinder the pet's movement. The receiver also needs to not produce to much heat or it could aversively affect the pet; this means it is important that our voltage regulators do not produce too much heat. Further testing will be needed to determine what amount of excess heat is acceptable. The collar needs to be wireless when in operation, so the pet is not constrained by wires. The only exception to this is when the user is using USB to get the data off the receiver. The transmitter needs to have good battery life to prevent the deterrent zone from turning off. Also, the transmitter needs to be small enough, so that it will not get in the way of the user's daily life.

The physical setup of the receiver collar and transmitters needs to be easy for the user. Putting the collar on the dog and activating the receiver should be easy. Activating the transmitter and selecting the deterrent settings also needs to be easy for the user to accomplish with minimal instructions. To do this, the design needs to be simplified that the average consumer will not have any difficulties.

The software design is user friendly and easy to setup for someone with basic computer knowledge. Also, the user interface is simple and easy for the average user to be able to learn and use quickly. The filters are all straight forward and logical.

The system needs to function in an indoor environment, so being able to work with obstructions such as furniture is needed. Currently, obstacles do cause a problem at close ranges because of the weakness of the signal strength. With a larger range this will not be as much of an issue.

Our prototype will not actually have a deterrent method, due to possible safety regulations and constraints. Instead there will be two LEDs, a green one to indicate a safe zone and a red one to indicate an off-limit zone.

2.4 Literature and technical survey

Indoor Positioning Systems

http://www.radianse.com/download/news-bit-2004.pdf

The goal of the Pet Deterrent System is to track pets in specific areas and deter them from off-limit areas. Top Dog Technologies looked at commercial products used to track items in an indoors environment. Indoor Positioning Systems have been used in hospitals for almost 20 years to locate patients and equipment. There are many different techniques that have been put to the test over the years and this document explains the pros and cons of each. The topics covered are Passive RFID Systems, Infrared Systems, Radio Triangulation Systems, Radio Fingerprinting Systems and Active RFID Systems.

PetSafe Electronic Indoor Pet Deterrent Systems

http://www.gundogsupply.com/peelpetdesy.html

This product is similar to what the proposed design. A transmitter with a variable range can be placed indoors to deter a pet from that area. If the pet, with the appropriate collar, gets in range of the receiver the collar will shock the pet.

Long Range Passive RFID-tag for Sensor Networks

http://ieeexplore.ieee.org/iel5/10422/33099/01559038.pdf?arnumber=1559038

This is a research project done by scientists at Tohoku University in Japan that works to control distances emitted by RFID transmitters. The bulk of our project involves controlling the range of the emitted signal from the transmitter. Our need to emit an accurate, strong signal in a specified range is a must for our product to succeed. The researchers in this paper have identified the problem with short-range RFIDs and have come up with a way of adjusting better ranges thereof. They have modified the traditional RFID tag to have a microstrip antenna and a voltage regulator to amplify the signal and to better control it. Our design calls for a specific control of the range by adjusting the input power. The microstrip antenna is an option on some of the transceivers we are looking at, in particular the TI TRF7960 unit. The benefits of this design are mentioned and the control, combined with voltage amplification, are noted and will be considered when building this project.

Pet Deterrent Project at Texas A&M University (Spring 2007)

Last semester a group of students in the Computer Science Department here at Texas A&M University worked on and developed a pet deterrent project similar to ours. Their work involved having a transmitter emit a signal over a controllable range and having the pet collar pick up the signal and deter whenever he was in close proximity. On the collar they made a PIC to store the information and had USB connectivity so that the user could monitor his or her pet's location. They ran into a problem when they ordered a long-range transmitter and were unable to successfully create the correct sized zones that would be needed in a common household. Top Dog Technologies hopes to work off of their idea and develop in a more system-approach manner that would allow more options to the user during setup.

Contech ScatMat

Contech provides a pet deterrent system called a ScatMat. This product is a mat that gives the pet a mild shock when he stands on it. This is meant to keep the animal out of certain areas of the house. It has three different levels of intensity, so the user can customize it for their pet. The downside to this design is that it can only keep a small area off-limits, 40''x20''. In addition to being a small area, the mat has to physically cover the entire area that the user wants to deter the pet from. If the user wanted to keep the pet off of a counter, the mat would have to cover the entire counter. Also, this would not be an effective tool to keep pets off of furniture. Another downside to this particular design is that the deterrent is only provided when the pet is standing on the mat. If the pet learns that he can walk over the mat quickly, then the pet might still get into the off-limit areas. Our design accomplishes the same thing as the product, just better. Ours can cover larger areas, does not have to physically cover the entire area, and cannot be easily bypassed by the pet.

Our proposed design is unique from the designs and products above because it stores data on what zones, both off-limit and tracking, the pet has entered. This data can be viewed using software on the computer. There are no products that are used to deter pets along with keeping a record of where the pet has been.

2.5 Evaluation of alternative solutions

One alternate solution to our pet deterrent system involves a system of Active RFID transceivers around a room setup to home in on a single transceiver located on the pet. The three transceivers setup on the walls would geometrically pick up the pet's signal and using the signal strength as a means of distance determine his location in the room. This would serve as a constant monitor for the pet. In order to deter him from certain objects, the user would setup an invisible perimeter by storing locations in the room (i.e. encircling the couch). If the pet came within proximity of these areas he would be deterred by a shock on his collar. The plus side of this design is that it is low cost and low bulk. There are only 3 transceivers and a system on the dog. The user could define their own off-limit areas and update them easily. The only downside to this project is the fact that all the distance measuring is done by the RSSI signal strength pin. This method of measuring distance, no matter how many transceivers you have setup, will always produce faulty measurements and never maintain any sort of consistency.

Building off of the previous design we could implement a similar system without having to worry about RSSI and signal strength. Several of the latest indoor positioning systems developed by researchers involve using multiple technologies together so as to act off one another's flaws. For instance, RFID on its own is unreliable because of the RSSI pin. GPS is excellent for positioning things with great accuracy, but only works outdoors. Ultrasonic technology can carry a precise signal, but only in a single room. By combing these technologies to work off of each other you have the potential to implement a somewhat accurate system that you could count on for consistency. This idea was introduced by a group of researchers at a conference in Korea in 2006. With the addition of a filter for the RFID, the system boasted the ability to search around obstacles, such as furniture, and noise that may alter the signal. The accuracy of this system was the big selling point in that it boasted a 2cm error maximum for a small room. Based on the research done this is by far some of the best results we have seen. In fact, this project could have been our first choice except for the complexity and price of the system. Having GPS, Ultrasonic, and RFID's working off of each other is not simple and building this project would take more than a single semester.

UWB is a radio frequency called ultra-wide band. UWB is not affected that much by multipath fading (distortion of a signal due to reflections), which is one of the main problems of normal narrow-band RF signals. This is because UWB uses short pulses that are not as prone to signal reflections overlapping the original pulse. Since they are not affected by the multipath distortion, the calculation for flight time/distance can be done by time of arrival instead of signal strength, and the signal can go through objects. Setting this up would be similar to the triangulation method as mentioned before, except that you would only need three receivers for the whole house. However, the downside to UWB is that it is very new. Because it is new, there are practically no receivers and transmitters for sell, and the ones that are for sell are very expensive.

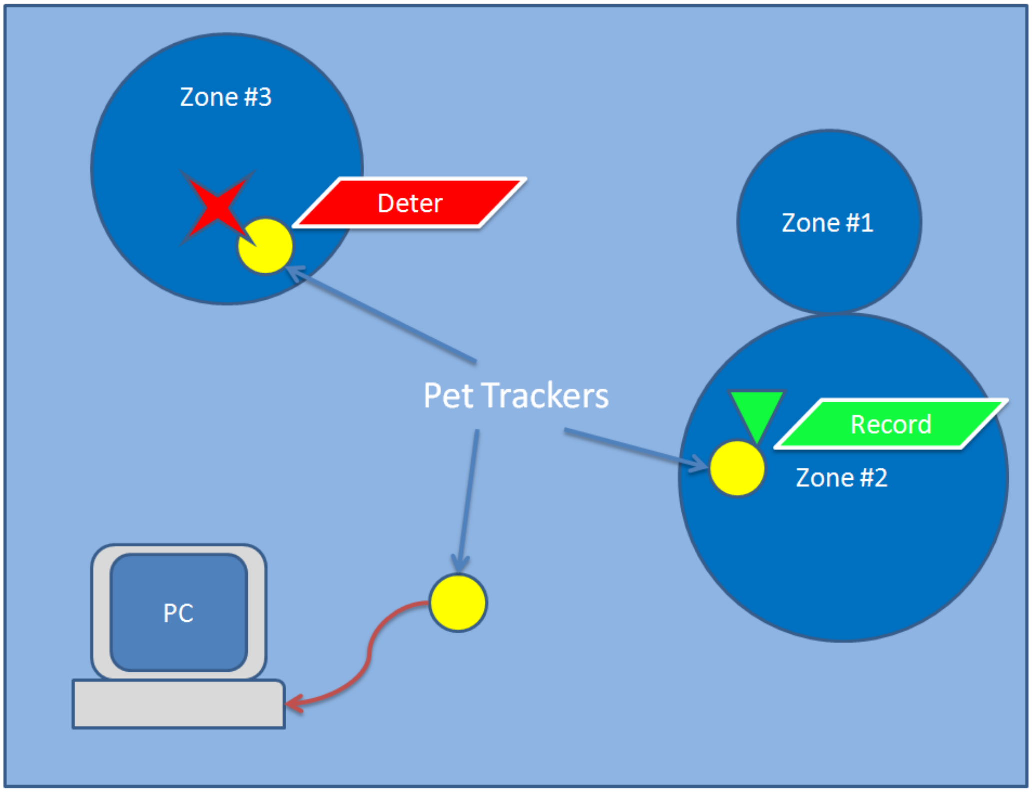
RFID stands for radio-frequency identification. The passive technology has no power source for the RFID tags. Instead they are powered by the electrical current induced from a signal by the reader. This gives the passive tags the advantage of using no power source that has to be replaced when it runs out. The downside to RFID is that it has a very short range, making it unusable for our project.

Unlike the passive RFID, active RFID can have a short or long range based on the transmitter. Even though it needs power, battery life in some RFID tags can last a few years. For tracking purpose's RFID has two major drawbacks. It cannot go through walls or objects well due to multipathing. The signal tends to reflect off objects making it hard to determine the signal strength. Secondly, signal strength has to be measured to determine distance, and this measurement is very inaccurate. The best accuracy we found using active RFID for tracking was with the LANDMARC system. The LANDMARC system used extra tags placed around the room to serve as reference points. Multiple readers are then used to read signal strengths from the reference tags and the tags being tracked. By comparing the readings, it can be determined which reference tags, the tracked tags are closest too. Using this data, and approximate location can be determined. However, even using this system, error was still in the 1-2 meter range, which is too high for a deterrent system. Also, placing all of these reference tags would make our system more expensive and impractical. No one will buy a commercial product requiring them to use reference tags.

3 Final design

3.1 System description

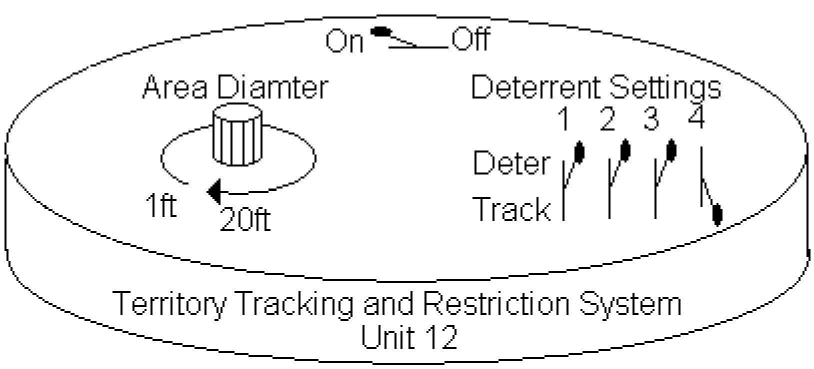
The system will be comprised of primarily three main components; the transmitters which create the zones of interest, the receivers which are on the pet collars, and the client program which is the user interface to monitor all the information. The initial step in the process is for the user to define the zones of interest. This is done by placing the transmitters in areas in which they want to either track or deter. They must then define a range for the radius and select which pet collars will be set to the deter mode and which collars are set to only track. This step is refined by testing the ranges and making sure the system is functional. Next the transmitter will continually broadcast a repeating signal stating its unique transmitter ID and a code for which pets to deter and which to only track. Once a receiver comes in range it will process the message and respond accordingly. If the pet is not supposed to be in that zone it will trigger the deterrent mechanism. In either case however the zone ID will be stored in the memory along with a time stamp. If the memory becomes filled (meaning the user has not cleared the data in a long time) old data will begin to be written over. All events will be 3 bytes long (2 bytes for the timestamp + 1 byte for the tag information). Our memory is 64kB which means the system could store more than 21,000 events or about one week of constant data. Finally, the user takes the pet collar and connects it to a home computer. All data is transferred to the PC and cleared automatically from the on board memory. The user can then interact with the client program to monitor the recorded data. All data is saved for later use and by using this information they can make adjustments to the zones or simply return the pet collar to the animal.



3.2 Complete module-wise specifications

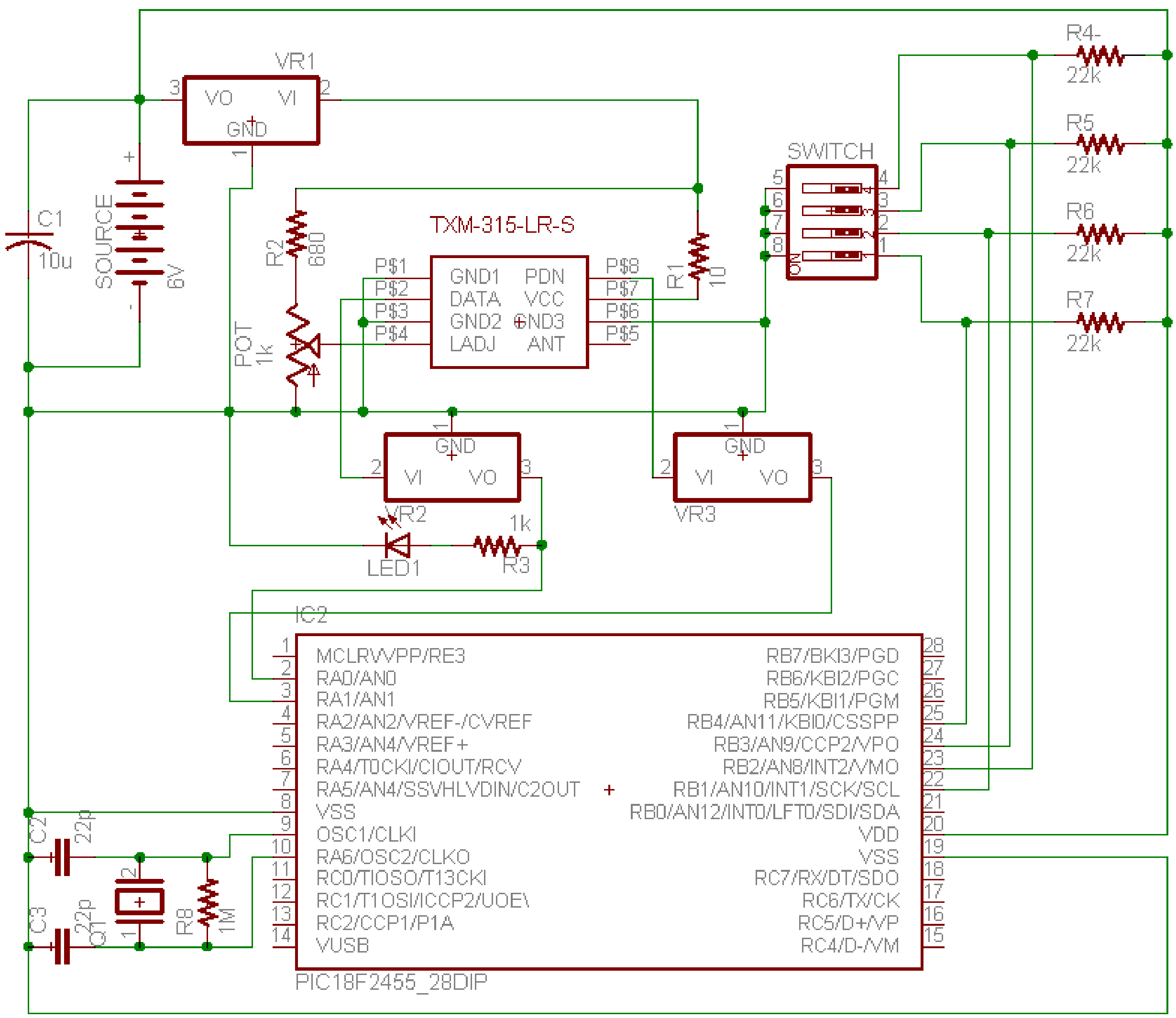
Transmitter

The transmitter has been designed to meet the objectives that specify that the product should be easy for the user to set up and adjust. Top Dog Technologies decided that the system should be usable with or without the software suite. A diagram of how the end project should look is found below. Each transmitter is given a unit number that represents the unique ID in decimal. Each collar is given a number 1 through 4 and the user may specify for each pet if the transmitter should track or deter by altering the appropriate switch in the deterrent settings section. The radius is adjustable by a knob, and should allow the user to create a circular zone with a diameter between one and twenty feet.



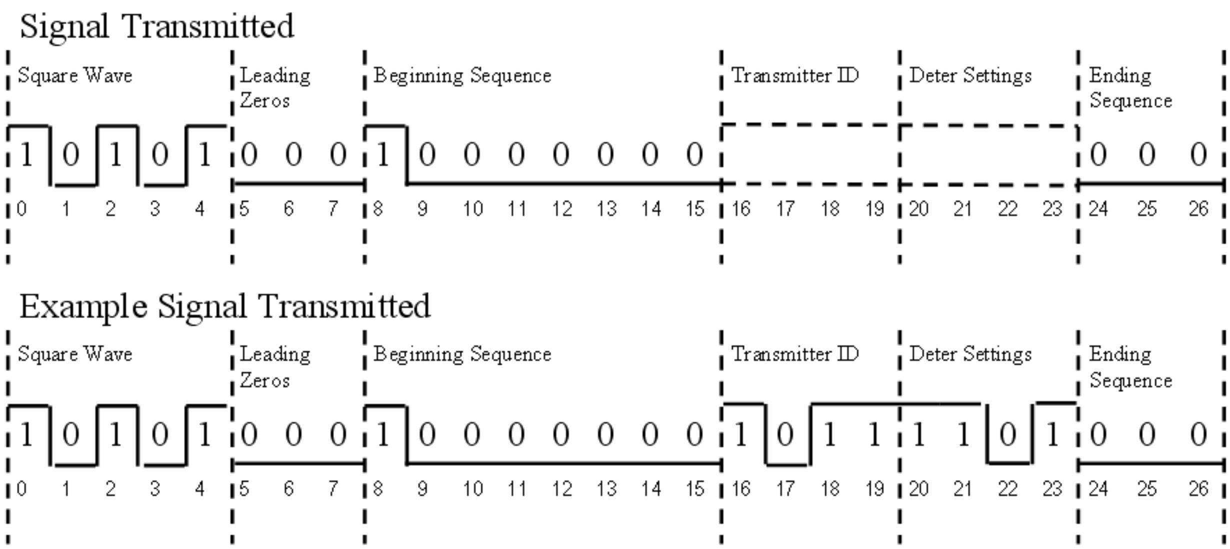
The schematics below are how we intend to make the transmitter fully functional. We have chosen to use the TXM-315-LR by Linx as our transmitter. The PIC created by Microchip, PIC18f2455, generates and outputs the signal so that the transmitter can send it. The switches determine the deterrent settings and the ID is programmed onto the PIC.

The schematic has undergone a few changes since the CDR report. Initially the ID was input to the PIC. This would require 16 different types of transmitters to be manufactured, the ID is now coded onto the PIC so that only one kind of transmitter is manufactured. We are using different kinds of switches, and they required pull up resistors. Another adjustment made was to sync the receiver and transmitter correctly they needed to run off the same oscillator speed. The receiver used an external oscillator to make USB connections possible, so the external oscillator, along with its resistor and capacitors, was needed on the transmitter as well. An LED was added to show when the transmitter is transmitting. A capacitor was added to keep the power source from introducing noise into the system. Two voltage regulators were added to keep the output lines from the PIC under 3.3V.



The transmitter initially sends a square wave so that the baud rate is within the range the receiver needs to pick up the signal. The square wave is followed by three zeros so that the rising edge of the beginning sequence is picked up. The beginning sequence of 10000000 is sent so that the receiver can verify that the signal is important. The beginning sequence consists of mainly zeros because noise is typically recorded as a one. The beginning sequence is followed by four bits containing the transmitter ID. The unique ID will identify this transmitter differently from the other transmitters set up in the house. Four bits for the ID will allow a user to place up to 16 transmitters. The next four bits sent in the signal determine the deterrent settings for each collar. This will allow the user up to four pets on the system. When the receiver picks up the starting sequence, it then waits for the ID to record it in its memory module. The PIC on the receiver knows which bit from the four bit deterrent setting that relates to it. If that specific bit is 0 then the transmitter is simply tracking the pet. If that specific bit is 1 then the transmitter is deterring that pet and the appropriate actions will be taken.

The signal pattern has been adjusted since the CDR report. The square wave was added to the beginning so that signal is sent at the correct baud rate to be picked up. The leading zeros were added so that the rising edge of the beginning sequence is easily detected. The beginning sequence contains many zeros so that noise is unlikely to be considered the beginning sequence. The ending sequence was shortened because it is not actually needed.

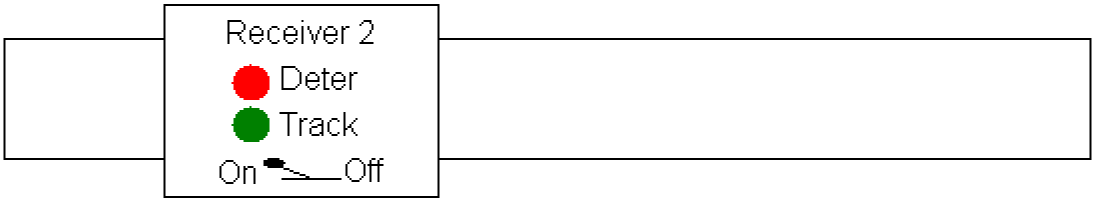


The user should be able to place a transmitter in his or her home, adjust the range with the knob and flip the switches to determine which pets are being deterred and which are being tracked. The transmitter then sends the appropriate signal, including the transmitter ID and deterrent settings, and when a receiver enters the zone the appropriate actions should be taken. The transmitter constantly sends a signal and then sleeps to save power. Here is a complete list of parts needed to create the transmitter that Top Dog Technologies has designed.

Part	**Part Number**	**Vendor**	**Distributor**	**Price**	**Number per Transmitter**
Transmitter	TXM-315-LR	Linx	Digi-key	$7.46	1
PIC	PIC18F2455	Microchip	Microchip	Free Sample	1
3.3 Voltage Regulator	UCC383TDK TTT-3	TI	TI	Free Sample	3
Switches	----------------	----------------	----------------	Free Sample	4
Potentiometer	----------------	Radio Shack	Radio Shack	$2.99	1
Resistor	----------------	----------------	----------------	Free Sample	8
Capacitor	----------------	----------------	----------------	Free Sample	3
LED	----------------	Radio Shack	Radio Shack	$.50	1
Resonator	----------------	Digi-key	Digi-key	$.33	1

Receiver

The receiver module is what receives signals from the placed transmitters and deters the pet from the off limit locations, as well as store information about the incident. Ideally, the module will consist of only a few ICs with the addition of a small battery. This should be a rather small unit upon completion and will easily fit on a pet's collar with ease and comfort, meeting our set objectives. The receiver will receive and process any information it gets from the designated transmitters. In order to process the data and eventually store it we will need a microcontroller, memory, and an interface. The method of deterrent for our application will be notification by an LED.

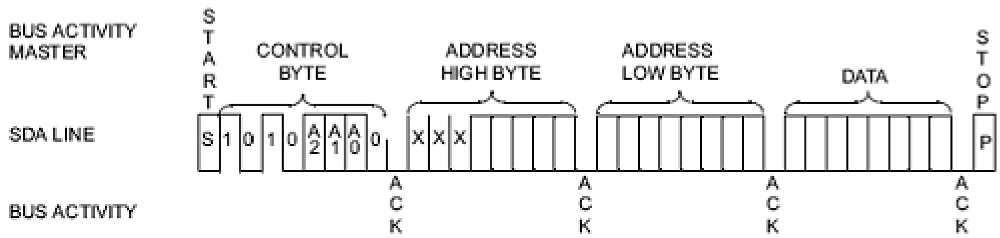


The receiver used is the RXM-315-LR-S from Linx Technologies which is the counterpart to the transmitter that is used in our setup. We will receive information across a 315 MHz signal. The -112 dBM sensitivity on this module is excellent for picking up the weak signals transmitted over short ranges on the transmitter. In fact, we have tested this to work within a range of around 3 inches to show consistency. The module itself is rather simple since it only has sixteen total pins, nine of them being No Connections. Upon hookup, the only pins used are the Vcc, GND, and the DATA pin. Whenever the receiver picks up a signal emitted from the transmitter it will send it across the DATA pin to be processed in the PIC.

The microcontroller in our case is a PIC18F2455. It handles all of the information processing and is the core behind the receiver module. Once the receiver gets a signal from the transmitter it sends it from its DATA pin to the PIC's Rx pin, which receives data and stores it to a register. Currently, our bit setup is to send an 8-bit open identifier, then 4 bits needed for the transmitter ID, then 4 bits needed for the hot-encoded pet deterrent settings, and 3 bits for a close identifier. The PIC will process the data accordingly and judge if a deterrent is needed. If the bit for the particular pet is high, then it means it is an off-limit zone. If this happens, a signal is sent out of the RB5 pin. When a signal is received the LED will blink, or in the case of a real module, shock to deter. The RB4 pin will flash if a pet enters a zone he is allowed to. This is meant solely for testing and observation and will not be used on the final product.

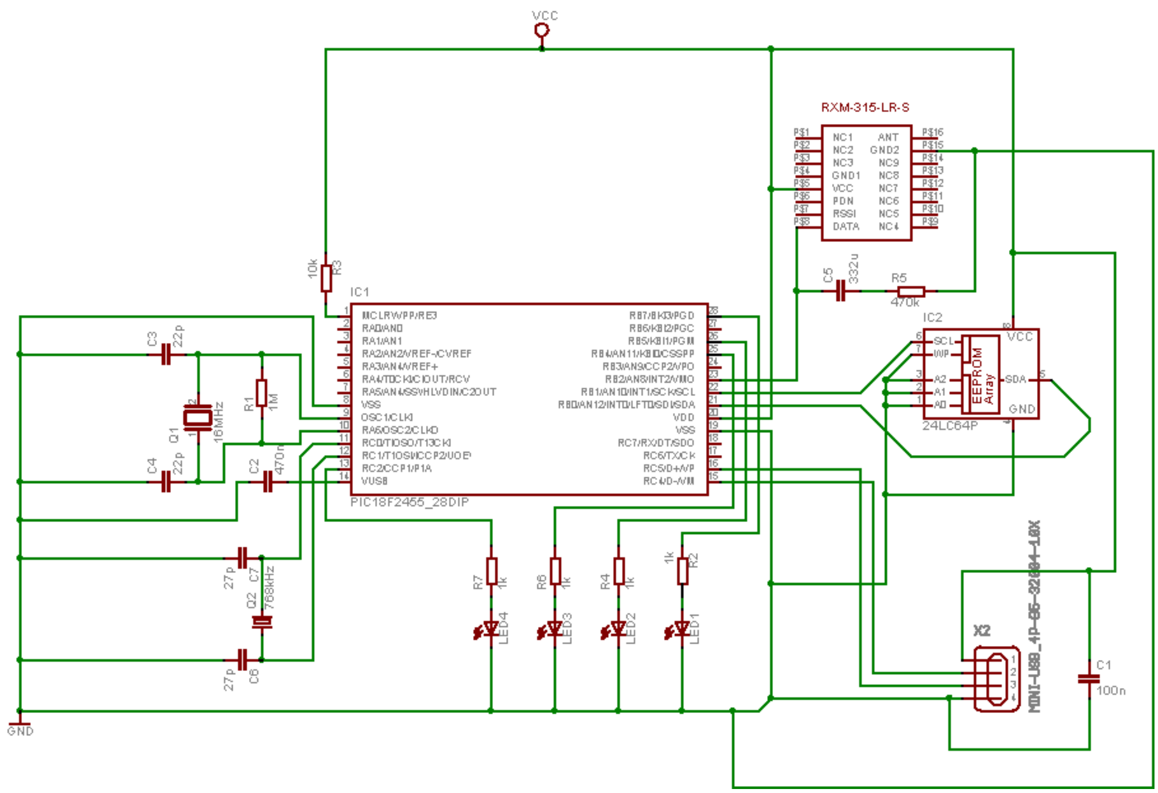
All instances of data that the PIC receives will be stored so that the user can follow the patterns of their pet(s). The information needed to be stored is 3 total bytes: the transmitter ID (which is four bits), a bit for deter/not deter, and two bits for the petID. The second and third bytes store a short timestamp so the user can know when and where their pet has been. The timestamp is a 16bit value in minutes. Since the PIC has only 256 bytes of memory to work with we needed an external memory module. For that we attained the Microchip 24LC64 memory module which has 64K memory that can be accessed serially. This means we only have to worry about two pins, SDA and SCL, which we can connect to the PIC respectively. Storing and retrieving the data will now work over the same line and the address pins are not needed.

The EEPROM is very simple. It has address lines that allow you to segment your storage, a write protect line, and the serial clock and serial data lines. Storage segmentation and write protection is not needed for our project. The SCL and SDA are used to do byte writes and byte reads. The control is set by the program and then the high and low addresses are kept in global variables and incremented or decremented after each operation. Lastly the data is sent and then an ACK is sent back to the PIC.



The user interface to the stored data will be by USB. This is by far the most complicated aspect of the entire module because it requires a large amount of code and a few extra components to allow it to run correctly. The USB connector itself will be a type-B connector that has 4 DIP pins on it. Those pins are Vcc, GND, D+, and D-. The D+ and D- are the data lines for USB. These pins are connected to their counter D+ and D- on the PIC. In order to send signals correctly, the proper frequency must be obtained. To do this we need an external oscillator. Based on the recommendations from the tech doc we need a 16 MHz ceramic oscillator and two 22pF capacitors to make it run properly. The two sides of the oscillator are fed into OSC1 and OSC2 on the PIC. From here they are fed into the PIC's internal PLL, which multiplies the frequency to make it run properly for USB purposes. The code we used for this was provider by PICCoder.co.uk and is implemented based upon the USB Generic class as provided by Microchip.

The final module for the receiver can be seen below.



Client Software

The User Interface for our project allows the customer to upload the data from the pet's collar to the computer. Here, the data is presented in multiple ways to show the user when and where the pet was in a zone, whether allowed or off-limit. The code is written in C# using the .NET framework. The power of Visual Studio allowed us to create a powerful GUI with little effort. It gives a very professional feel that will represent our product well on the market. The User Interface begins by calling a function that connects to the Receiver module (collar) through the USB interface and compiles the data onto the computer. The data is automatically saved to disk with a timestamp for a filename. All of the data that was on the PIC is now deleted and ready for a fresh start again. The user is now able to view and see different interpretations of his data on the computer. Choosing between Pet ID and Transmitter ID you can filter the material that is displayed on screen. You can view instances in a range of time or by zone. You have the ability to view bar graphs, pie charts, timelines, and tree charts to better observe the training of your pet in bad zones and his time spent in each zone. The purpose of the software is to demonstrate the process of training your pet from certain areas as well as an advanced tracking system.

3.3 Approach for design validation

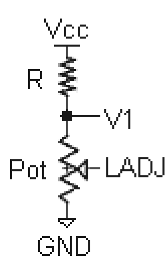
Top Dog Technologies followed the Range Test, the Deterrent Test, the Power Test and the software tests documented in the CDR. The power test was adjusted to use math to calculate how long the batteries would last. The software tests were combined into one Software Test for simplicity. These tests verified that the system does what it was designed to do.

4 Implementation notes

Transmitter

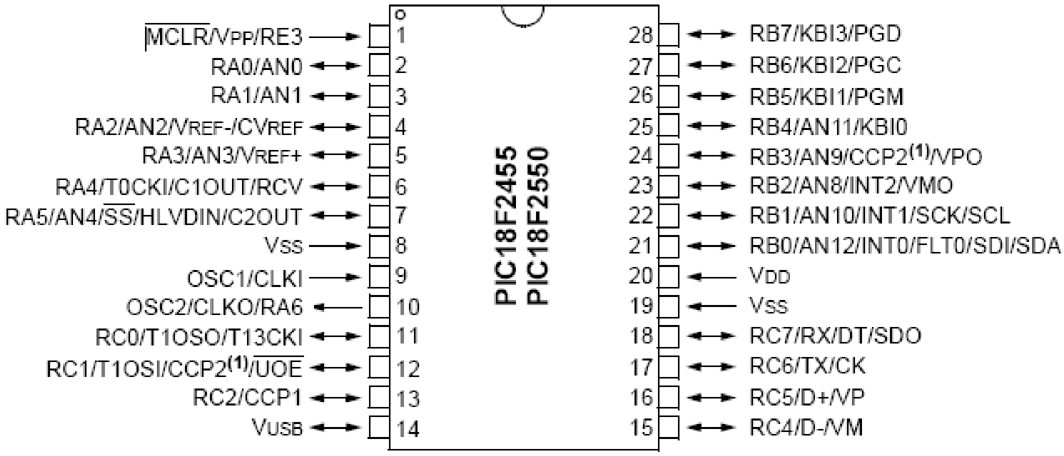
When building a transmitter the most important decision is what RF Transmitter IC should be used. There are many different options on the market, but not all will work for the Territory Tracking and Restriction System. The main considerations we took into account were the transmission range and range variation. Top Dog Technologies chose the Linx TXM-315-LR because the LADJ pin is used to change the transmission range based on input voltage. The chip met our range and range adjustment requirements. The transmitter's signal is sent at the frequency of 315MHz, this is a common frequency used in the USA and we should have used a different frequency. The TXM-315-LR runs off of 3.3V while the PIC we use runs off of 5V. This required several voltage regulators in our circuit to ensure the transmitter does not receiver too much voltage. A feature we also liked about the transmitter was that it could be sleeped based on a one or zero at the PDN pin. So when choosing a transmitter for a pet deterrent system, take into consideration the range, range adjustment, frequency, voltage and sleep potential.

In the project requirements the team decided that the transmitter should have an adjustable area from a three foot diameter to a twenty foot diameter. A voltage divider using a resistor and a potentiometer is found in the transmitter schematic to control this feature. We invested in a 1k multi-turn potentiometer to have better precision. During the range tests the transmission range was mapped to different voltages at the LADJ input pin on the transmitter. A ten foot radius was found to have 2V at the LADJ pin. Here are the calculations required to determine the resistor that regulates the potentiometer to only change the transmitter range from zero to ten feet. With zero feet determined when LADJ is grounded and when ten feet is determined when LADJ is equal to V1. The variables may be adjusted to achieve different results.



$$V1 = \left(\frac{Pot}{R+Pot}\right)Vcc$$

The sleep pin and data pin on the transmitter are connected to the PIC through a voltage regulator. Top Dog Technologies chose Microchip PIC18F2455 because it had the correct number of I/O pins, had USB capabilities and was used the previous semester. Most of the project's hardships came from programming the PIC due to strange configuration settings and the fact that none of us had done PIC programming before. When determining which PIC to use for a pet deterrent system take into account the number of I/O pins, USB capabilities and ease of programming.



The PIC for the transmitter takes in the deter settings from four switches, generates the signal, and sends the signal to the transmitter data pin while it wakes the transmitter by sending a one to the PDN pin. The configuration settings on the PIC are set to allow these pins to act in the following way.

Pin Name	Pin Number	I/O	Variable	Variable Description
AN0	Pin 2	Output	pin_sig	Signal sent to transmitter
AN1	Pin 3	Output	pin_sleep	Value that sleeps the transmitter
AN8	Pin 23	Input	pin_det0	Deter setting
AN9	Pin 24	Input	pin_det1	Deter setting
AN10	Pin 22	Input	pin_det2	Deter setting
AN11	Pin 25	Input	pin_det3	Deter setting

The program begins by initializing the configuration settings and then setting up the signal. A different function is then called to initialize the signal into a global array. By changing the values being stored into the array the signal is very easy to adjust. This portion of the code allows the programmer to alter any segment of the signal including the ID of the transmitter.

Once the pins have been initialized and the signal set up the PIC enters into an infinite loop that will last until the circuit is turned off. The program cycle wakes the transmitter, reads in the deterrent settings, sends the signal, sleeps the transmitter and delays before beginning the cycle over.

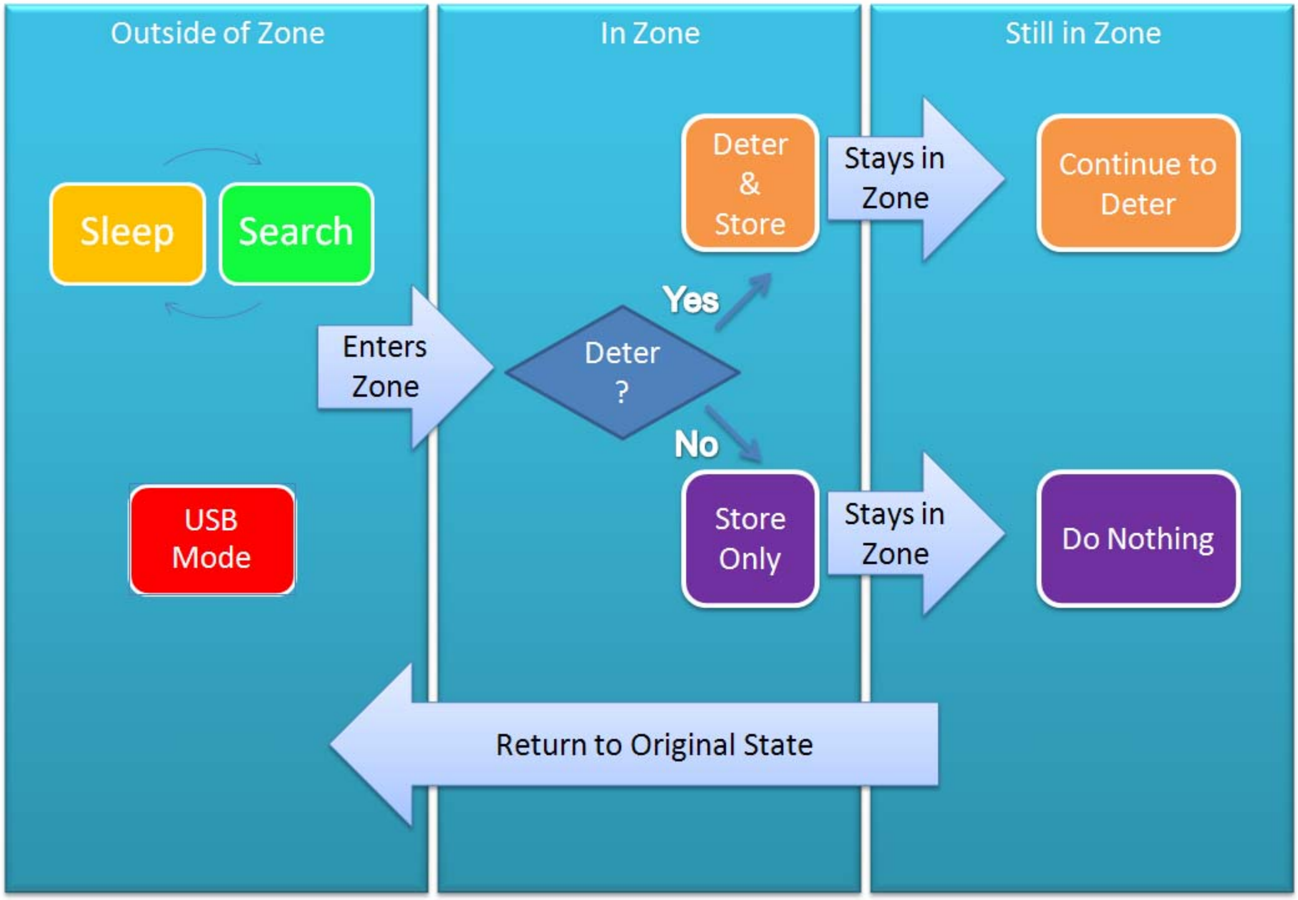
The important aspect of the transmitter code is how fast the signal is sent. The signal baud rate is determined by a delay that happens after each bit is sent through the signal output pin. When we first began sending the signal generated by the PIC, instead of a square wave generator, the receiver did not receive it. On the data sheet for the receiver, RXM-315-LR, there is a data rate specification of between 100 and 10,000 bits per second. This is information that must be checked to make sure the transmitter sends data at the correct baud rate. A simple way to check the baud rate of data is to send a square wave from the PIC and connect an oscilloscope to it for measurement. So that the delay amount on the transmitter corresponds to the delay amount on the receiver the oscillator the PICs run off of must be the same. Typically the internal default oscillator would work, however for the receiver to have USB capabilities it had to run off of a faster external oscillator. This added the external resonator, two capacitors and a resistor to the transmitter schematic.

The signal transmitted is split into six portions. The first five bits create a square wave that is used to get the baud rate in a good range to be distinguished by the receiver. Top Dog Technologies added this section of the code when we noticed the beginning of the signal was not being received consistently. The receiver starts checking for a beginning sequence one a rising edge interrupt. Therefore the second portion of the signal consists of zeros so that when the beginning sequence starts the rising edge is not overlooked because the system was already in an interrupt. The third portion of the signal is the beginning sequence. The beginning sequence is used so that the receiver can verify that it is reading the signal from the transmitter. Originally the sequence consisted of mostly ones, however we began receiving many false positives because noise is often registered as a one. The sequence is now 10000000 which rarely produces a false positive. The fourth portion of the signal is the four bit transmitter ID. The ID indicates which transmitter zone the receiver is in. When the receiver is connected to the client program, the transmitter ID will allow the user to see which areas the pet traveled into. The fifth portion of the signal is the deterrent settings. The PIC determines the deterrent settings according to four input pins connected to switches. This allows the user to determine which pets are allowed in the zone. The ending sequence that consists of zeros is the sixth and final portion of the signal.

The transmitter is an important part of the pet deterrent system. The design specifications given above will help build or adjust the transmitter. Unfortunately RF technology is inconsistent. There is a lot of noise and the signal strength varies. If a person places their hand between the transmitter and receiver there is a good chance the signal will be obstructed and the receiver picks up bad data. Overall we feel the transmitter was completed and achieves most of the objectives we specified in our project proposal. However, we do not feel the transmitter can be consistent enough to be a consumer product. If it is placed underneath a couch, the receiver will probably not be able to read the signal consistently.

Receiver

The receiver is a complicated combination of technologies that revolve around the 18F2455 PIC we used. The receiver module will receive a signal on the RXM-315-LR-S receiver and then send the bit stream to the PIC to be processed. Once processed, data parts will be parsed and certain parameters set based upon this. Finally, a complete instance, including a timestamp, will be stored on the EEPROM memory module to be stored for later use. So, the receiver can be broken into basically four main parts: the data processing from the receiver, the code parsing and timestamps, the storage to memory, and the USB uploading.



In order for the PIC to even function correctly with all the things we need, a series of configuration bits must be set to get the PIC to not only function, but to oscillate accordingly with the USB. The bits for PIC operation are:

```
#pragma config WDT = OFF
#pragma config WDTPS = 256
#pragma config MCLRE = OFF
#pragma config LVP = OFF
```

The LVP pin in this case is very important because it turns off a feature known as Low-Voltage Programming. Unless you implicitly tell the compiler you do not want this feature, the PIC will not run, yet it will not give any errors in compilation, which makes it difficult to debug. For all of the USB material we have to set the right oscillator to get it to work correctly. The internal PIC clock is only 16MHz, which is not enough to power the 48 MHz required USB frequency. In order to get High-Speed USB working with our PIC, we must incorporate an external oscillator and run it through the PIC's PLL which will implement a 48 MHz signal. Following the data sheets we added a 16 MHz ceramic oscillator with two 22 pF capacitors going to ground. These are run out of the OSC1/OSC2 pinouts on the PIC. The configuration bits needed to run this setup are as follows:

```
#pragma config USBDIV = 2
#pragma config CPUDIV = OSC4_PLL6
#pragma config PLLDIV = 4
#pragma config FOSC = HSPLL_HS
#pragma config VREGEN = ON
```

Basically, these pins are telling the compiler that we need the 48 MHz USB frequency and that we have a 16MHz resonator. The CPUDIV bit sets the frequency of the PIC's normal operations to 16 MHz so that they are easier to work with for receiver processing. The VREGEN is an extremely important pin for USB as it sets the USB voltage regulation for the whole PIC. Without setting this bit you are disabling your USB functionality with very little debug help to find out why.

The data processing is done on the PIC and it looks for certain patterns to tell when to look for data. This is because we feed in a series of start bits that lets the PIC know when real data is being sent. A lot of research went in to knowing how to properly process the data. Since the receiver picks up a lot of noise, it is hard to distinguish a real data signal and a fake one. For this, we spent time researching something known as Manchester Encoding. This process parses data bit streams by edge detection. For instance, a 1 is read as a rising edge while a 0 is read as a falling edge. For this to work on the PIC, in addition to two interrupts running off of each other we would have to incorporate timers and measure distances between the data. Considering the time constraints we had on our project and the issues we were experiencing with the PICs we could not implement this fully. Instead, we took the idea of edge interrupts and played off of that. We setup edge detect interrupts to trigger whenever a rising edge was detected on the DATA in pin for the PIC. When this edge was triggered, the following function was called:

```
#pragma code low_vector = 0x18
void low_interrupt (void)
{
     _asm GOTO LB _endasm
}
#pragma code
```

When the low interrupt is triggered it goes to this function and, using assembly commands, it calls our function for processing data, LB(). The configuration bits to setup these are as follows:

```
//Interrupt Initialization
INTCON2bits.RBPU = 0;
INTCON2bits.INTEDG2 = 1; //set rising edge sensitivity
INTCON3bits.INT2IP = 0;
INTCON3bits.INT2IF = 0; //turn flags off
INTCON3bits.INT2IE = 1; //enable interrupts
IRCONbits.IPEN = 1;
```

```
INTCONbits.GIEL = 1;
INTCONbits.GIEH = 1;
IPR2bits.USBIP = 1;
```

When we get inside our LB function to process the data, we disable interrupts and begin checking for patterns. The main idea behind our algorithm is polling. We send a beginning sequence of 1000000 each bit being sent for about 300 clock ticks. This gives us enough time to process a data point. Once we find the beginning sequence, we then poll the next bits for about 310 ticks a piece and guage whether it's a 1 or 0 based upon the percentage of 1s and 0s we polled. Once we have our 8-bit data stream, we parse according to the pet ID and then determine whether to deter or track.

As mentioned in the transmitter section the first four bits of the data is the transmitter ID while the last four are hot-encoded data pins for pets 1-4, respectively. The PIC stores the 4 bits for the transmitter ID in a byte. Then, based upon which pet number the receiver is, it finds its particular hot-encoded bit. If the bit is 1, this means deter and for our demonstration we light a red LED. If the bit os 0 this is an allowed zone for the pet and all we do is track. This single bit is also packed in the same byte as the transmitter ID as well as the Pet ID. For storing purposes, we always store a deter occurrence, regardless of time. For tracking we only store once per minute.

Another portion of the receiver is also the largest code-wise, the USB uploading. Whenever the user wants to logon and view where his pet has been, he plugs in the USB cable and uploads his pet's data onto the computer. The USB code was provided by PICCoder.co.uk and it uses the USB Generic Modules from Microchip to produce a high-speed USB connection. This code required several modifications to work not only with our PIC, but in general. The code was written for the 4550 PIC, which happens to be in our same family, but still requires an overhaul for the 2455. For one, we had to modify the linker script as well as all the pin connections (the pinouts on the 4550 are different than that of the 2455). The one last thing we had to change was to get the the power detection code in the program to work, which turned out to have several bugs in it. The USB has a power-detection module built-in. Meaning it can run the PIC off of a 5V source or straight from the computer. It has a sensor coded in that automatically detects which power to use. This code was buggy and it had to be modified for functionality. Basically, after reading through several hundred lines of code, all that needed to be fixed was to comment out the following two lines.

```
//#define USE_SELF_POWER_SENSE_IO
//#define USE_USB_BUS_SENSE_IO
```

These modules are buggy and if you comment them out the USB sensor will work and your code will run.

Once the USB is setup and the driver is installed, you are ready to send data streams to the computer. The USB Generic class send packets of data as an array of bytes. Basically we used a dyanmic array of bytes structure called DATA_PACKET which was provided with the code. We modified it send array of 3 bytes everytime the call signal was sent to the PIC. Three bytes are needed to be sent for each recording: the first for all the packed data (transmitter ID, pet ID, good or bad zone) plus two more bytes for the timestamp.

```
HDByteReadI2C(0xA0, htemp, ltemp, &temp1, 0x01);
ltemp++; //increment the temp address
```

```
if(ltemp == 0x00)
      htemp++;
dataPacket._byte[1] = 1;
dataPacket._byte[2] = temp1;
HDByteReadI2C(0xA0, htemp, ltemp, &temp2, 0x01);
ltemp++; //increment the temp address
if(ltemp == 0x00)
      htemp++;
dataPacket._byte[3] = temp2;
HDByteReadI2C(0xA0, htemp, ltemp, &temp3, 0x01);
dataPacket._byte[4] = temp3;
temp1 = 0x00;
ltemp++; //increment the temp address
if(ltemp == 0x00)
      htemp++;
```

This section of code highlights how we send data to the PC across USB. The PIC pulls the 3 bytes off of the EEPROM and then packs them accordingly into the data packet. The second bit of the data packet is set to 1 in this example. As long as that is 1 the PC will continue to read off of the EEPROM. When that is 2 it will stop sending requests. Thus, the data read will cycle through all recorded instances on the EEPROM and then stop at the end.

When the receiver module is powered on and everything is initialized the EEPROM is also initialized with *InitPICEEPROM* function. This configures the serial clock and serial data lines, disables SMBus and Slew Rate control, enables MSSP Master mode and sets the baud rate for the clock that we use.

```
Void InitPICEEPROM(void)
{
      TRISBbits.TRISB1 = 1; //Configure SCL
      TRISBbits.TRISB0 = 1; //Configure SDA
      SSPSTAT = 0x80; //Disable SMBus & Slew Rate Control
      SSPCON1 = 0x28; //Enable MSSP Master
      SSPADD = 0x09; //Should be 0x09 for 400kHz
      SSPCON2 = 0x00; //Clear MSSP Conrol Bits
}
```

The EEPROM storage is done every time the receiver picks up a deter signal for its hard-coded pet collar ID. If the signal is not set to deter the pet the zone instance is only recorded once a minute. This sets our tracking resolution to one minute and our deter signal resolution is set by a delay in the receiver software. The basic idea is to signal the deterrent for a set amount of time and then delay for another period of time allowing the pet to leave the zone before we deter it again.

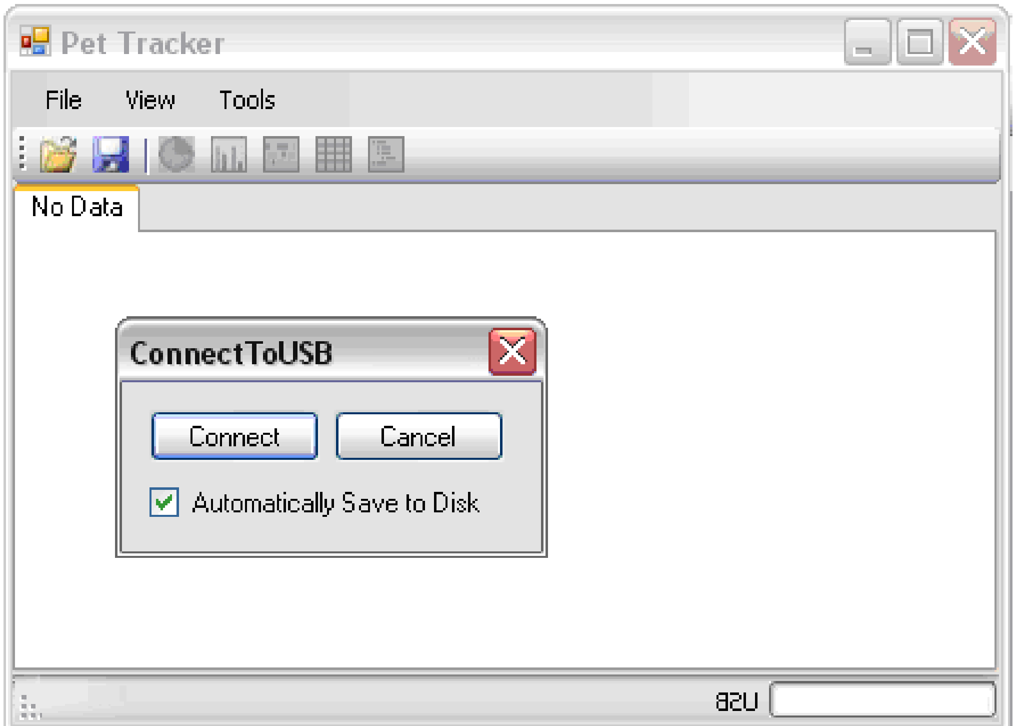
The basic flow for the storage logic is as follows. First, the signal is processed. An initial rising edge is detected and after the starting high bit is sent a sequence of 7 lows are sent. This was chosen because most of the time noise is read as a high. After the initial sequence the receiver then begins processing the actual data section of the signal. The transmitter ID and pet deterrent settings are stored in local variables. Next, the deter bit for the collar ID is checked. If it is set to deter the deterrent is signaled, the byte is stored to the EEPROM using the current address, and the address is implemented. There is a delay after the store to make sure the PIC has time to properly store the information on the EEPROM. The current address is incremented by incrementing a lower address and if there was an overflow incrementing the high address. The address is stored in this way because the EEPROM we have uses a two byte address. After the data is stored the PIC then stores the current timestamp to the EEPROM. This takes two calls to the write function because the time stamp is two bytes. Again after each write the address is incremented and there is a delay. Finally, the PIC waits to allow the deterrent to finish and then waits again before listening allowing the pet to leave the zone.

```
if((dataArray[COLLAR_ID + 3] == 1) && (SentPacket != 0))
{     // deter the pet
      // signal the red LED
      LATBbits.LATB4 = 1;

      // Pack the Data
      SentPacket = SentPacket << 1;
      SentPacket = SentPacket + 1;
      SentPacket = SentPacket << 2;
      SentPacket = SentPacket + COLLAR_ID;

      // Write the packet to the current address
      HDByteWriteI2C(0xA0, haddr, laddr, SentPacket);
      //Record Instance of Zone Intrusion

      // Delay to make sure it writes
      delay();delay();delay();

      laddr++; //increment the address
      if(laddr == 0x00)
      haddr++;
```

```
        // Store the current time
        HDByteWriteI2C(0xA0, haddr, laddr, H_byte);
        delay();delay();delay();
        laddr++; //increment the address
        if(laddr == 0x00)
        haddr++;
        HDByteWriteI2C(0xA0, haddr, laddr, L_byte);
        laddr++; //increment the address
        if(laddr == 0x00)
        haddr++;
        //Delay the deter signal
        // Turn off the Deter and delay again
        // allowing the pet to leave the zone
    }
```

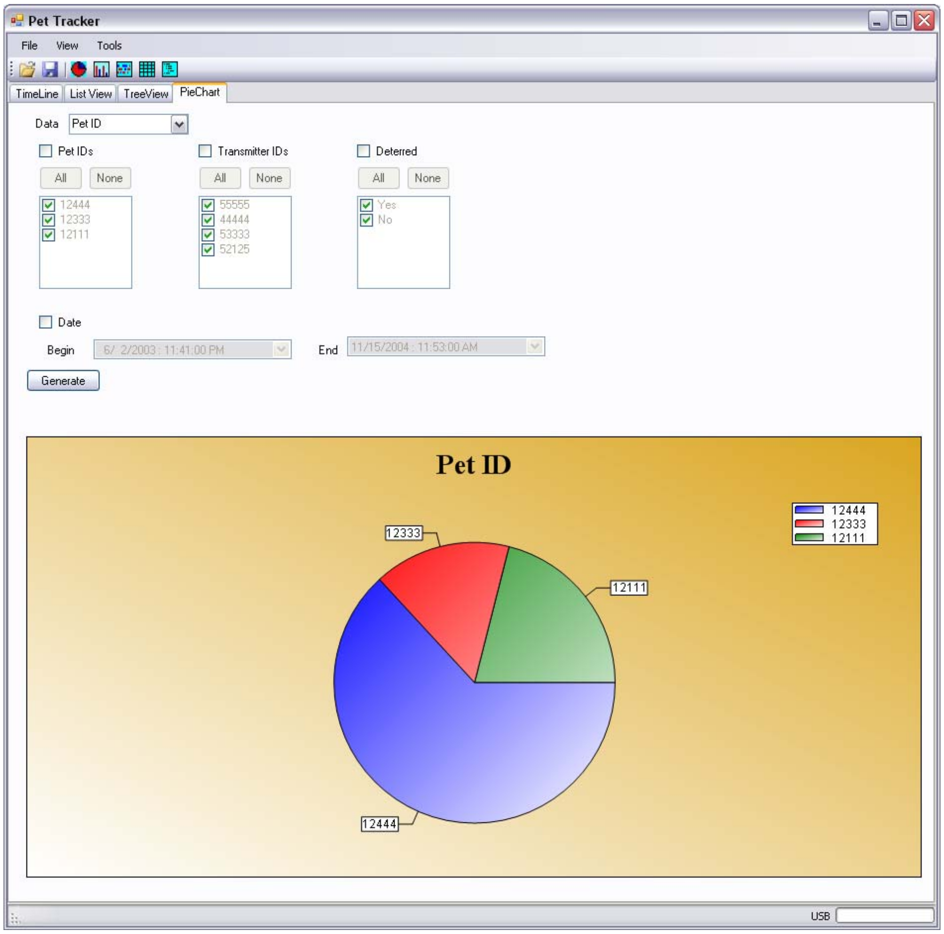
Client Software

The software is composed of a GUI (graphical user interface) written in C# that allows the user to upload and view data from the receiver. The software will interface with the reader through USB. Utilizing a C# file that includes a DLL which adds low level functionality for USB communication the software is able to download all of the data containing the time and zones the pet entered since the data was last read from the receiver module. The software will be able to receive the pets ID from the data along with the timestamps to associate the zone breach events with a specific pet. The timestamp for each event that is stored is determined in a clever way. Instead of syncing times between all of receivers and transmitters with the users PC, we utilize a timestamp that is started at 0 once the receivers are initialized and continue incrementing this value. Once the information is downloaded the client software looks at the current time and works backwards to determine the real time that each event took place at. Then the real times, as well as transmitter IDs and the deterrent setting , are kept for later use in a file.



Example of the USB download

Once the data is automatically saved to disk it is loaded into the software for display. The user can then view the data in multiple different ways and apply several different filters. The first and most rudimentary view is a list view. It is simple a spreadsheet view of the data organized chronologically and separated into columns with all the relevant fields. Next, there is a tree view which allows the user to expand each file loaded and continue traversing the tree from there. Then the software has three graphical depictions of the data. This is where the filters can be used. The first graphical view is a bar graph, the second is a pie chart and the third, and probably most useful, is a timeline view. Each of the views are displayed below the filter management section and have a legend and labels describing the information being displayed. This effectively accomplishes our team's goal of having easy and straightforward information communicated to the pet owner. There are four useful filters that the client software has implemented. The first allows a user to filter data based on the pet IDs. This way a user can choose only the dogs or a specific pet. The next filter is a filter based on transmitter IDs. This means a user can look exclusively at all zones in the living room area or even home in on a specific zone of interest. A third filter is selecting just deter instances or non-deter instances. Filtering the data this way allows the user to look an only the tracking aspect of the data or only the pet training part. The final filter is a filter on the date. This allows a certain range of time to be shown.



A sample pie chart view of the data

One or more of the filters can be applied at a time and after each change in the filter management section the user must click the generate button to update the view.

The main structure of the program:

Main Class – Initializes the program and starts the GUI

GUI Classes – Contains the graphical user interface implementations and event handling

Pet Class – Contains the data for one pet, including: pet id, timestamps, misc. info

Data Class – Contains the data storage and retrieval functionalities

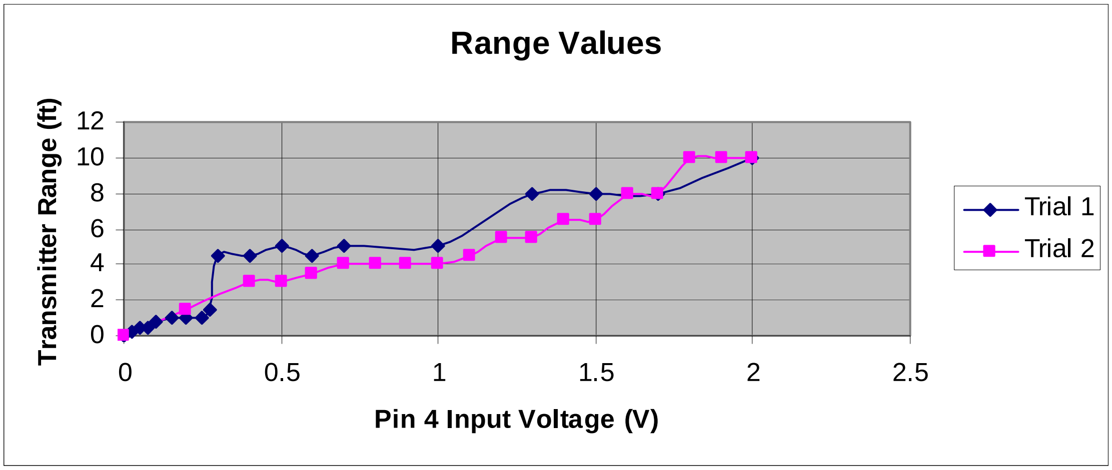
5 Experimental results

Range Test

To begin with Top Dog Technologies is going to demonstrate the Territory Tracking and Restriction System with one transmitter unit and one receiver unit. The receiver unit on the collar is set to portray pet 1. The transmitter is now turned on and the range adjusted with the knob to a ten foot radius. The deterrent setting for pet 1 is set to deter. The collar should be moved to enter the ten foot radius and the LED on the collar should light up, indicating that the deterrent device is activated. This test must be repeated for different transmitter ranges.

The range tests were fairly successful. We charted the ranges that correlate to different voltages at the LADJ pin. The signal was not cut and dry, there tended to be a one to two foot fuzzy region beyond the designated area where the receiver occasionally picked up the signal. When data was recorded from the fuzzy range it had a higher probability of being garbled.

Trial 1 Pin 4 Voltage (V)	**Trial 1 Boundary (ft)**	**Trial 2 Pin 4 Voltage (V)**	**Trial 2 Boundary (ft)**
0	0	0	0
0.025	0.25	0.2	1.5
0.05	0.5	0.4	3
0.075	0.5	0.5	3
0.1	0.75	0.6	3.5
0.15	1	0.7	4
0.2	1	0.8	4
0.25	1	0.9	4
0.275	1.5	1	4
0.3	4.5	1.1	4.5
0.4	4.5	1.2	5.5
0.5	5	1.3	5.5
0.6	4.5	1.4	6.5
0.7	5	1.5	6.5
1	5	1.6	8
1.3	8	1.7	8
1.5	8	1.8	10
1.7	8	1.9	10
2	10	2	10



These results indicate a linear upwards trend from 0 to 2 Volts that changes the range from 0 to 10 feet. Trial 1 (blue) and trial 2 (pink) were taken on two different days. The variation between the two trials is a little unsettling because the system should have a consistent range from day to day so that the pet is not confused on what the off-limit zones are.

Deterrent Test

This test will begin with the receiver unit on the collar portraying pet 1 and the transmitter deterrent setting should deter pet 1 but track the other pets. The collar must enter the zone and the red LED should light up. The transmitter deterrent setting should now be adjusted to track pet 1 and deter the other pets. The collar must enter the zone and the green LED should light up. This test must be repeated with the receiver unit on the collar portraying pet 2, 3 and 4.

Receiver Unit	**Deter Mode Rate of Success**	**Track Mode Rate of Success**
1	94%	95%
2	89%	98%
3	86%	93%
4	95%	96%

Overall the results of this test were successful. Occasionally when a hand or item separates the transmitter and receiver the data sent is mainly zeros and the green LED will be displayed even if the transmitter is on deter mode. Tracking is signified by a 0 bit so that it is less likely a pet will accidentally be deterred from a safe area. The rate of successes is lower when the distance between the transmitter and receiver is increased.

Power Test

The life of the power source in the transmitter and receiver can be calculated by measuring the amperage the units need to run. The calculations assume the consumer is using average AA batteries which are rated for 800mAh. When 800mAh is divided by the mA needed to run the system the result is how long the batteries will last.

The receiver needs 25.06mA on average to run. This means the life of the receiver is 31.92 hours. The transmitter needs 18.12mA on average to run, making the transmitter life 44.15 hours. We attempted to sleep the transmitter for longer durations to improve its lifetime. As found in the results, sleeping the transmitter for a longer duration slightly improved the lifetime of the batteries until the time period levels out at 44.86 hours.

Percentage the Transmitter is on	Amperage the Transmitter needs	Lifetime of the Transmitter
12% (Common case)	18.12mA	44.15 hours
10%	17.95mA	44.56 hours
8%	17.83mA	44.86 hours
6%	17.83mA	44.86 hours
4%	17.84mA	44.84 hours
2%	17.84mA	44.84 hours

There are a handful of ways the Territory Tracking and Restriction System's battery life can be improved. There are more powerful batteries on the market that have a higher mAh rating. Currently the transmitter is the only IC component that sleeps, the receiver and PICs can be given a sleep schedule to cut down on power. Another way to greatly improve the power consumption issue is to use better voltage regulators. If these power saving techniques do not achieve the desire result there are other options for the project. A docking station could be designed to charge the receiver portion of the collar every night. The transmitters could run off AC electricity and be plugged into the wall. The power results of the current system are unacceptable according to our project's objectives. We believe that the problem could be overcome if we had more time to implement some of these changes.

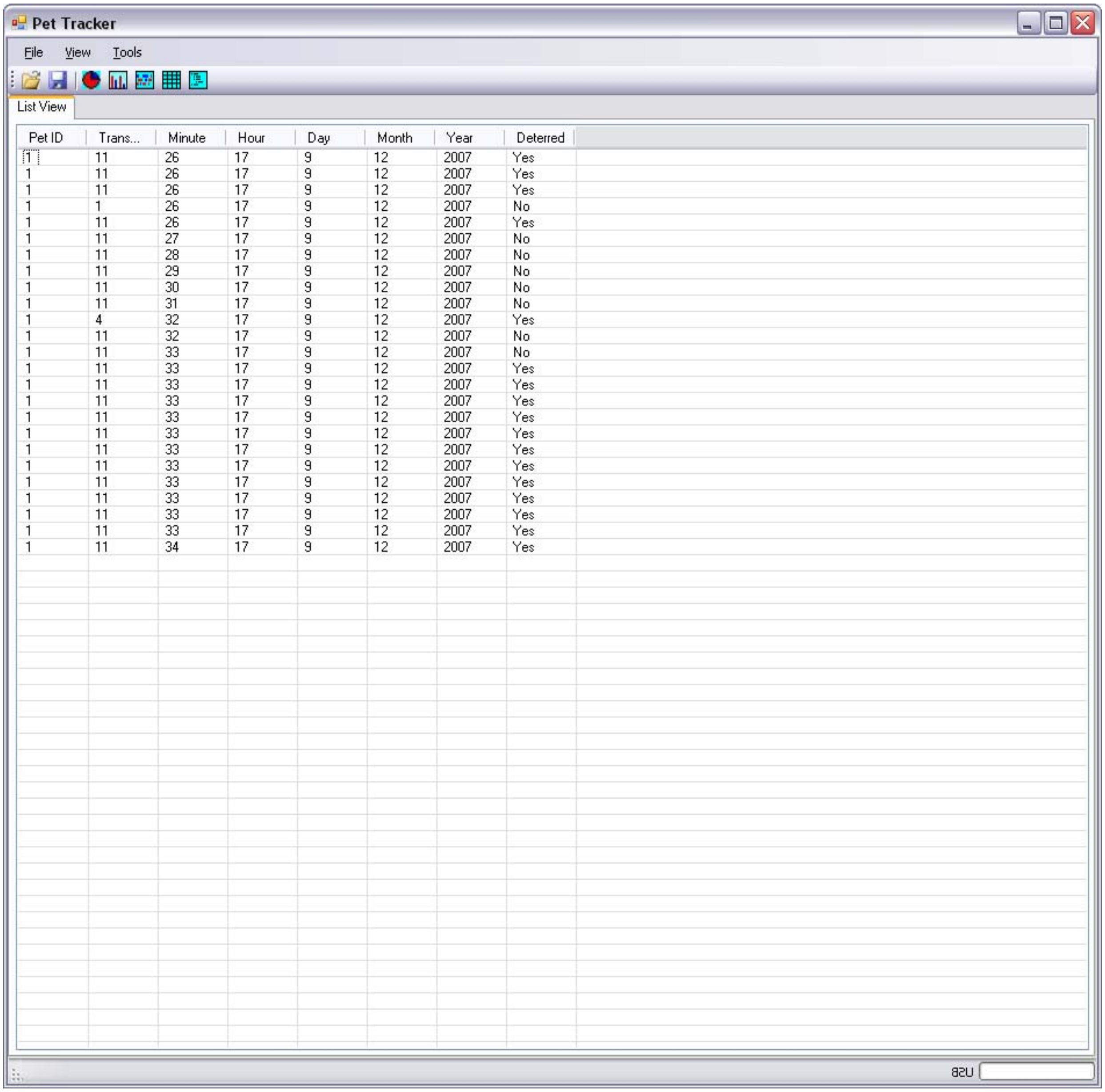
Software Test

After the hardware has been validated the software must be tested. To test the software the team members must keep a handwritten record of when the collar enters a zone and what the zone is set to. To begin this test the receiver unit on the collar should be set to portray pet 1. The transmitter should have the ID of 0001 and be set to deter pet 1. The collar should move into the transmitted zone multiple times while the transmitter is being adjusted. The ID on the transmitter should occasionally be changed. The transmitter deterrent setting should also switch between tracking and deterring for the pet. The collar should then be connected to the computer and the data represented with the software must be checked for accuracy. This test should be done with the receiver unit portraying different pets.

These tests were performed and the system consistently represented accurate data. We are looking into why the software crashed on one occasion when the USB was connected to the machine. Here is a record of one of our tests along with a screenshot of the results.

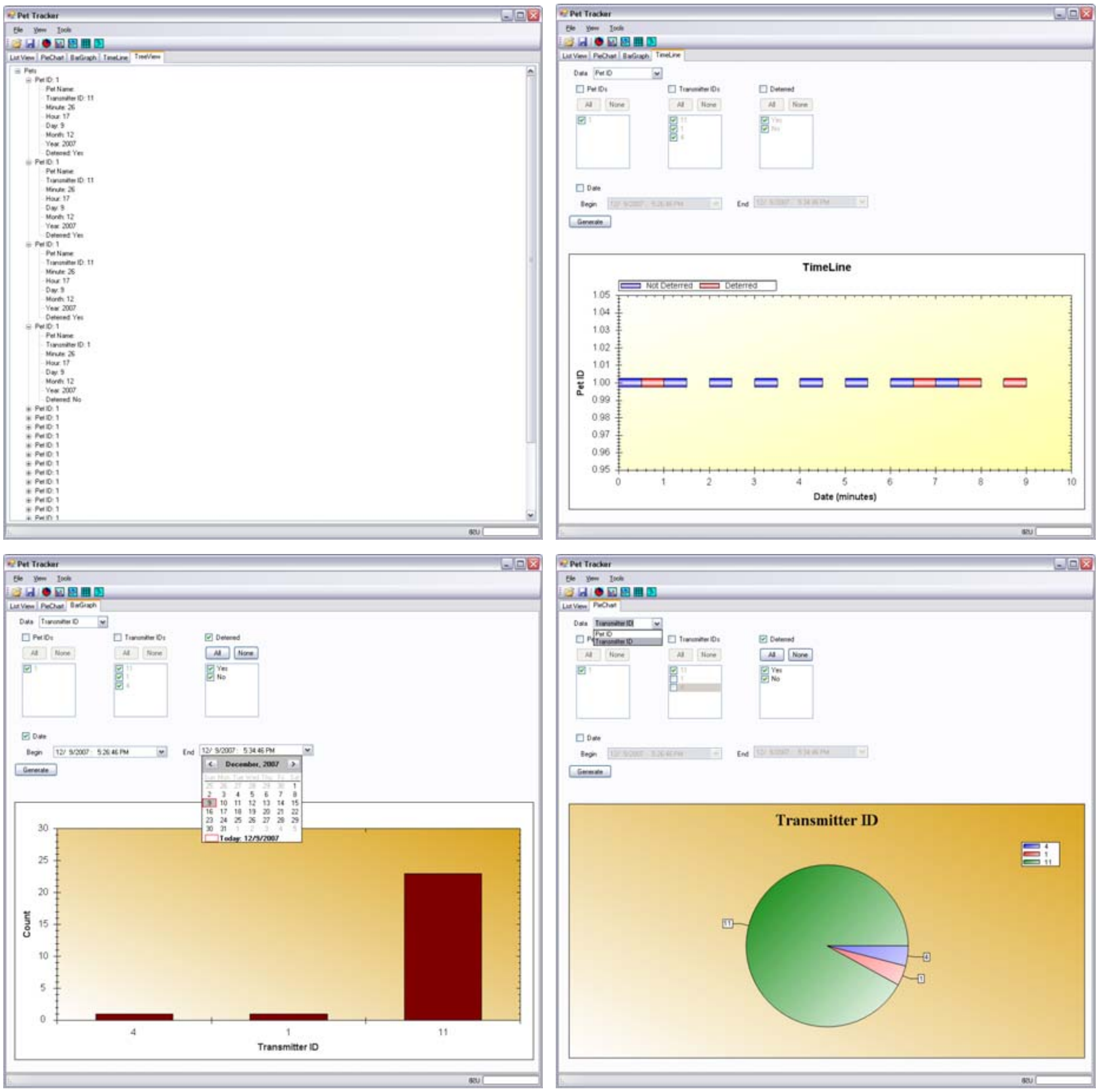
Pet ID	Transmitter	Deter or Track
1	11	Deter
1	11	Deter
1	11	Deter

1	**11**	**Deter**
1	11	Deter
1	11	Track
1	11	Track
1	11	Track
1	11	Track
1	11	Track
1	**11**	**Track**
1	11	Track
1	11	Track
1	11	Deter
1	11	Deter
1	11	Deter
1	11	Deter
1	11	Deter
1	11	Deter
1	11	Deter
1	11	Deter
1	11	Deter
1	11	Deter
1	11	Deter
1	11	Deter



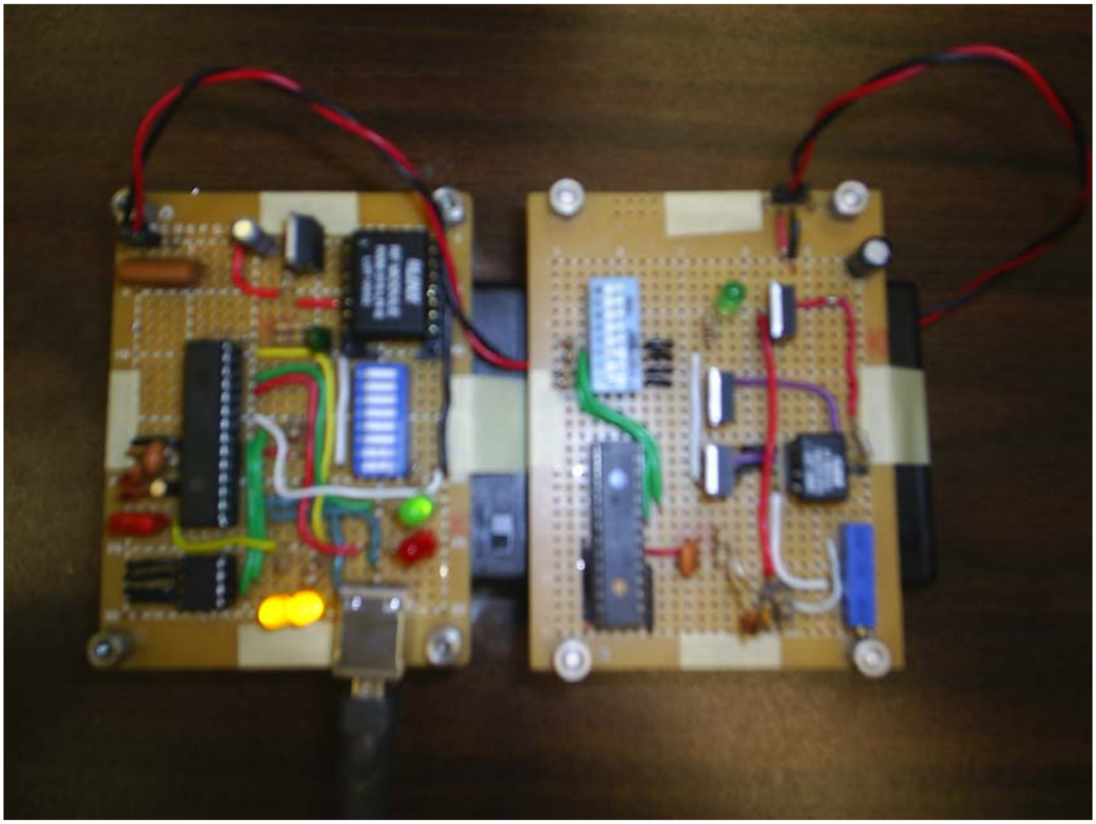
Pet ID	Trans...	Minute	Hour	Day	Month	Year	Deterred
1	11	26	17	9	12	2007	Yes
1	11	26	17	9	12	2007	Yes
1	11	26	17	9	12	2007	Yes
1	1	26	17	9	12	2007	No
1	11	26	17	9	12	2007	Yes
1	11	27	17	9	12	2007	No
1	11	28	17	9	12	2007	No
1	11	29	17	9	12	2007	No
1	11	30	17	9	12	2007	No
1	11	31	17	9	12	2007	No
1	4	32	17	9	12	2007	Yes
1	11	32	17	9	12	2007	No
1	11	33	17	9	12	2007	No
1	11	33	17	9	12	2007	Yes
1	11	33	17	9	12	2007	Yes
1	11	33	17	9	12	2007	Yes
1	11	33	17	9	12	2007	Yes
1	11	33	17	9	12	2007	Yes
1	11	33	17	9	12	2007	Yes
1	11	33	17	9	12	2007	Yes
1	11	33	17	9	12	2007	Yes
1	11	33	17	9	12	2007	Yes
1	11	33	17	9	12	2007	Yes
1	11	33	17	9	12	2007	Yes
1	11	34	17	9	12	2007	Yes

The data shown on the client software has two inaccurate transmitter IDs and two inaccurate deterrent settings. Since we visually watched the receiver LEDs during the test we can verify that the data is received incorrectly before being sent to the client program. The client program accurately displays the data in list format, tree format, timeline format, bar graph format and pie chart format.

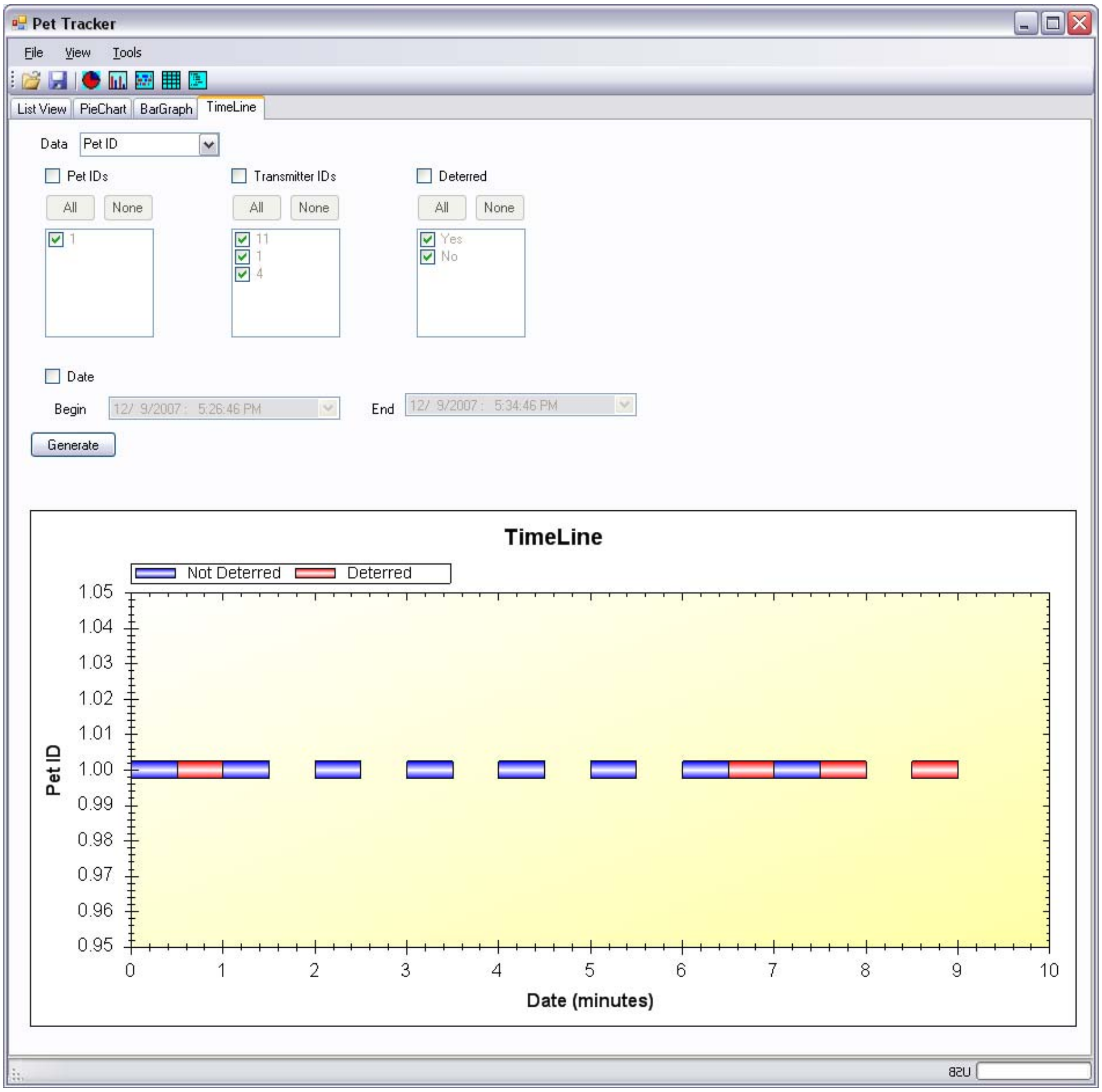


All of these tests will verify that our product works and meets the need to have a pet deterrent system that tracks pet movement throughout the house 24/7 by monitoring and documenting when a pet enters off-limit areas and deters the pet when needed. The tests have verified that the product runs correctly and meets all needs with the exception of power consumption.

The Final Product



This image shows the receiver on the left and the transmitter on the right. The transmitter and receiver are nicely laid out on experimental boards to be mobile examples of the products.



Here is a screen shot of the final client software system. The system shows the pet information in many useful ways.

6 User's Manuals

Hardware Installation/ Usage

The transmitter and the receiver modules are each pre-built and ready for consumer user provided battery installation. Each module needs 4 AA batteries to operate correctly. To install, place 4 AA batteries inside the attached battery holder on each module. Be sure to close the battery pack by screwing it shut when you are done.

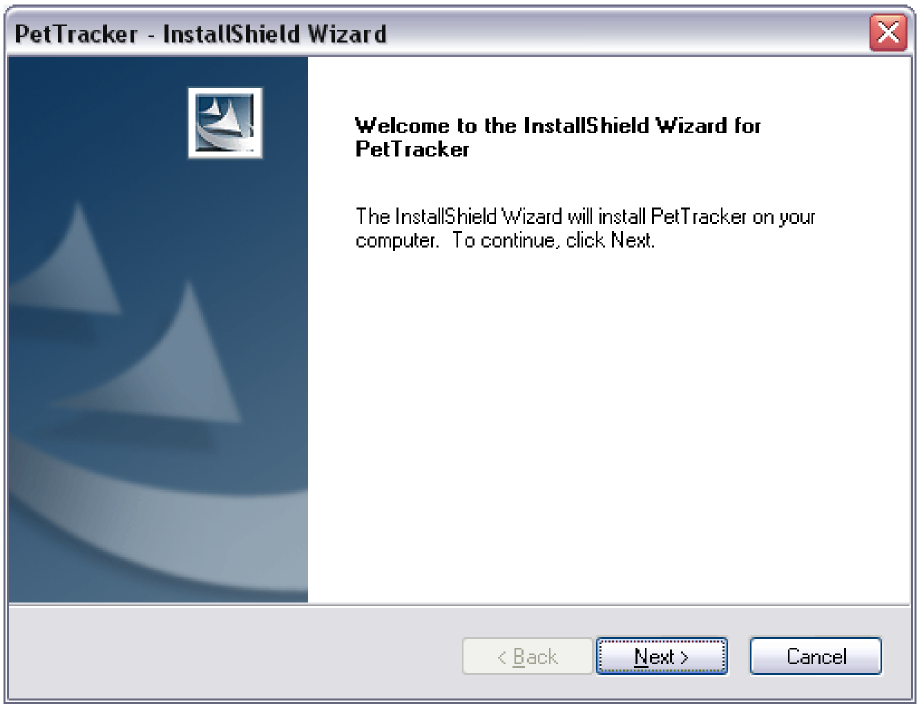
To turn on each module, flip the switch on the battery pack to the ON position. You will notice a blinking LED on the transmitter and flashing lights on the receiver. To begin tracking and deterring on the pet collar, be sure to flip the switch from data transfer mode to Track/ Deter mode.

When you want to connect to the PC to send data set the switch on the receiver to USB mode and connect to the computer. You should notice the flashing yellow light stay solid yellow, meaning a stable connection via USB to the PC. When you are done uploading unplug the cable and set the receiver back to track mode.

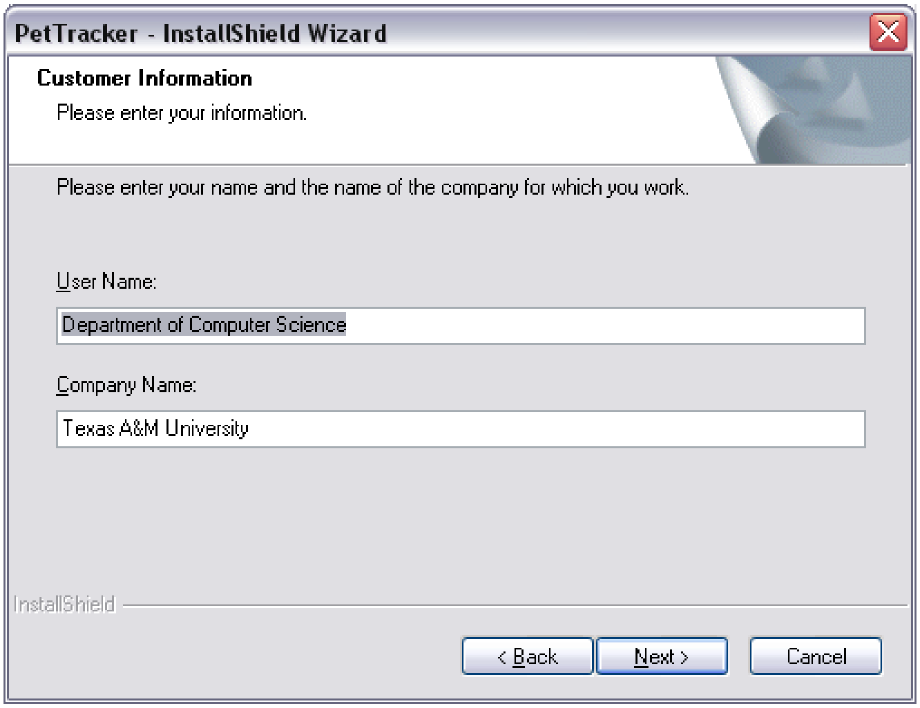
The transmitter has two modes to set: track or deter. Simply flipping the switch between the two will set the transmitter's function.

Software Installation

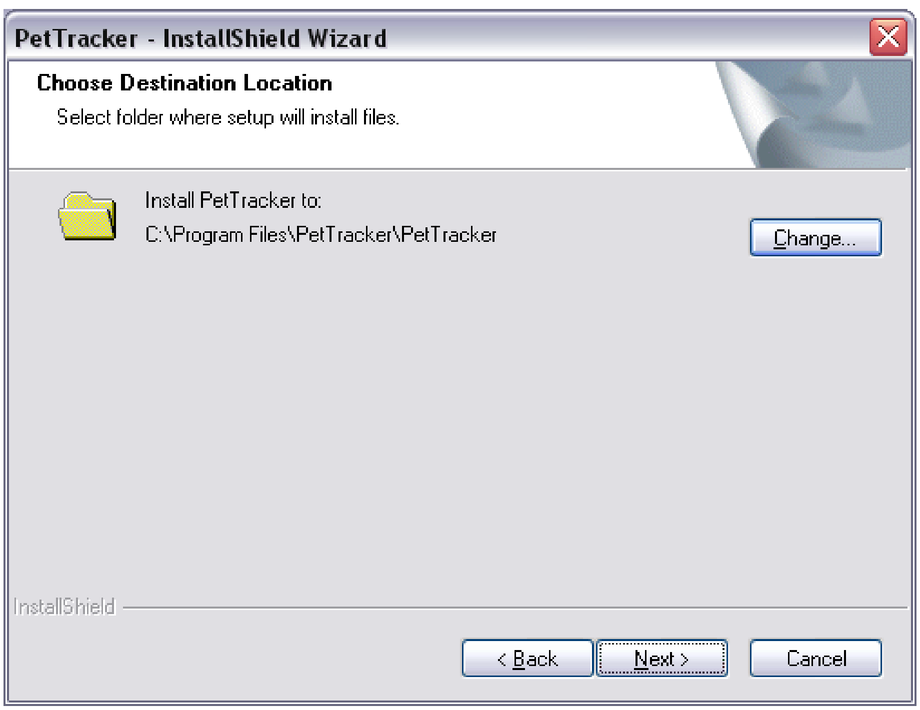
To install the Top Dog *Pet Tracker* software, click on the INSTALL.EXE file and follow the following instructions:



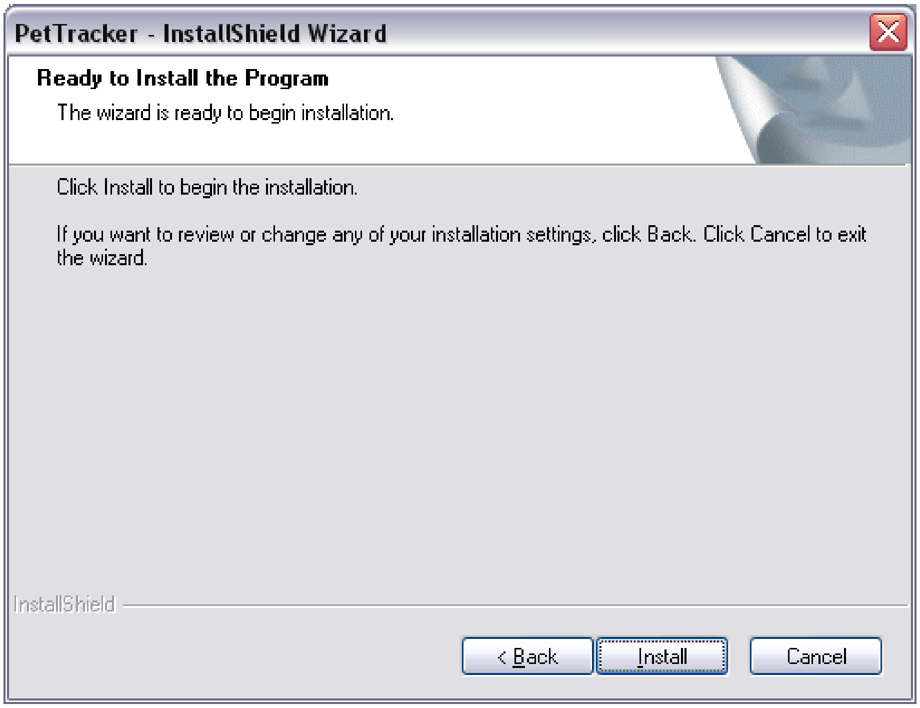
Click Next.



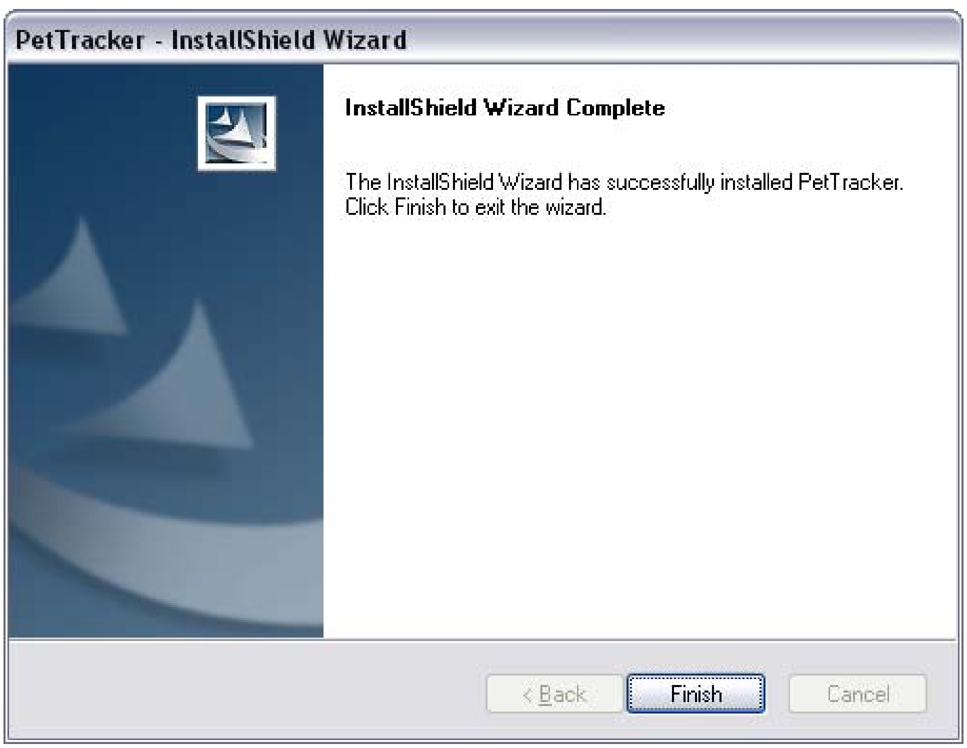
Fill out your information and click Next.



Choose an Install Path and click Next.



Click Install to install Pet Tracker software on your PC.

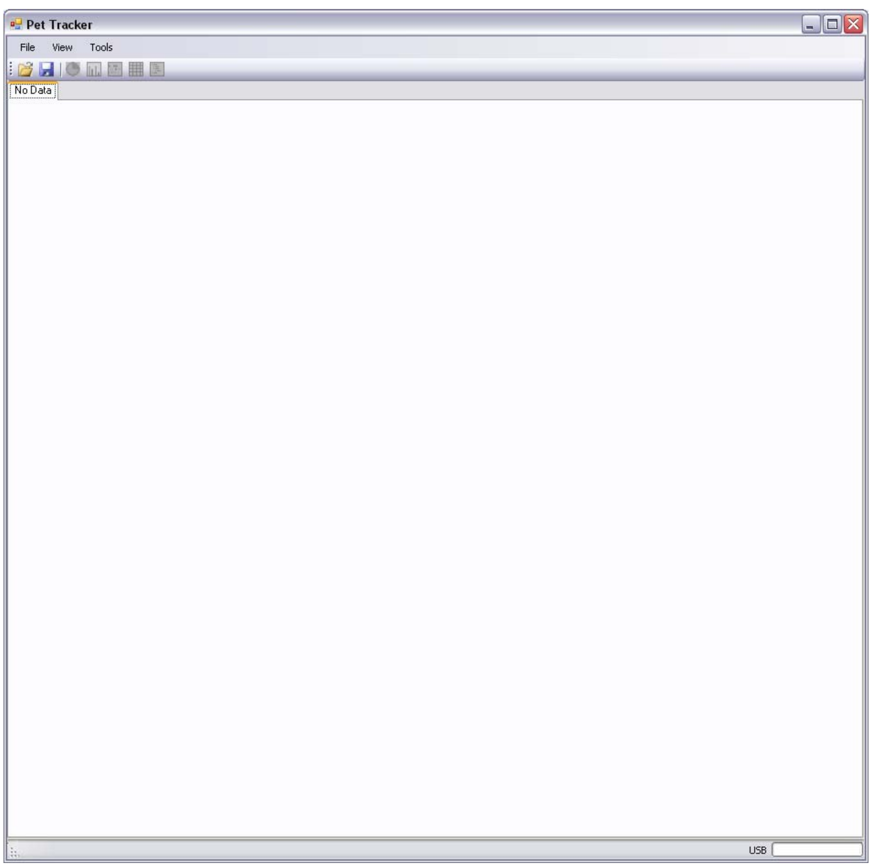


Click Finish and you are ready to use our software!

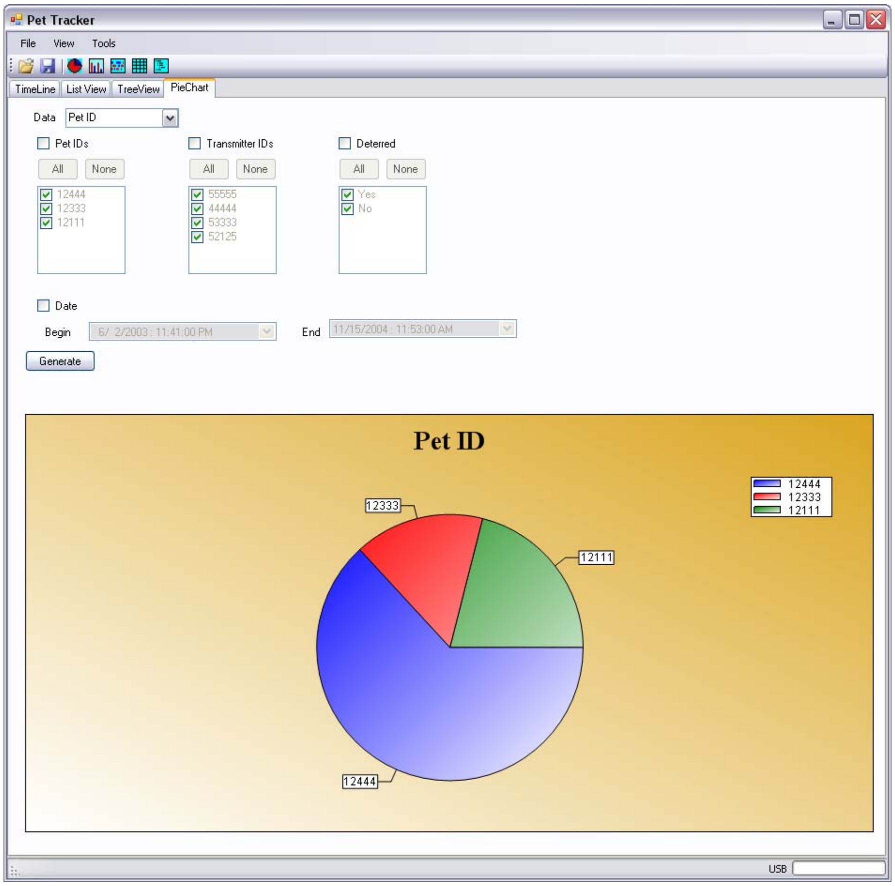
Software Use

To use our software, click on the Pet Deterrent icon and load up the suite.

Click the logo to enter.



Go to Tools-> Connect to USB to grab data from collar.



Click on the various tabs to observe all the different graphs and lists you can use to track and train your pet. Your data is stored automatically on uploads so if you ever want to load an instance again simply open the file for that particular date.

You can filter your observed data by date, pet, or transmitter as a means of organization.

7 Course debriefing

Team Management Style

Throughout this project our team management and our ability to work well with each other have allowed us to succeed. We divided the project equally amongst all the participants and kept up-to-date meetings and status reports so that everyone was on the same page. To keep our group on target with the preset schedule we had, we constantly set assignments to people so we could do as much as possible. Whenever a major issue came up, and that was common, some group members would stop their assignment and help whoever had an issue. The teamwork and the ability to help others really played out for us and took us to where we are now. In the end, it really was the ability to interact and communicate successfully that led to our progress.

Thoughts on Repeating the Project

If we had this project to do over again, the main thing we would do different is the receiver code and the picking of the receiver/ transmitter pair. Due to all the issues we had with the PICs and getting them to operate correctly we had little time to come up with a scheme to receive and process data. If we had gone into this project with a good background on PICs we would have easily spent a month on sending and receiving data and coming up with the best possible way to do it. As mentioned in the receiver code, we just use edge interrupts and poll data points for a specific number of clock ticks. Though this is functional and appears to work very well with our system, there are still many different ways to do it that are much better. When we were researching algorithms for picking out received data we stumbled upon many different algorithms, one of which was the mentioned Manchester Encoding, that were hard to understand and implement, but very reliable and maintainable. If we had the knowledge we have about PICs now at the beginning of the semester we would have been able to make a very reliable and sturdy system implementing the best possible algorithm for parsing and processing data.

Another thing we would have done differently would be to find a different receiver/ transmitter pair. Though our setup has a very small range, the RF communication we are using is very noisy and appears to be interfered with by obstructing objects. Since we needed to start testing and implement our design, we did not have enough time to fully research the parts we needed. I would have preferred looking more into what actual variable range transmitters the industry is using for our application. Certainly there is a technology out there that will produce a clean, variable range transmitter that is only slightly affected by obstructing objects. We just need to research more and find the best one for our application.

Safety and Ethical Concerns

As of right now there are no safety concerns with our product. Though it is intended for a shocker of some sort to replace the red LED we use in deter mode, it should be worn only by pets. We cannot really control what a consumer does with this product in their own home. Obviously someone could misuse the intent of this product and use it on their children. In that case a child would be physically harmed by our technology. Other than an obscure case like that, there is not any harm coming from our product, nor are there any pushing ethical concerns. We operate well within the legal FCC frequencies on our chip so there is no reason to interfere with something it might cause damage to.

Testing the Product

Testing is major issue that could use some adjustment for our project. Since all the technology we dealt with was new to us we obviously spent a lot of time testing. This was all part of the learning process, i.e. getting the PICs to run, testing the ranges of the transmitter, etc. When we finally had a functional product at the end, we were still running into various technical difficulties and never got into any kind of in-depth testing. There are a handful of shallow tests documented in the Experimental Results section of this report; however more thorough tests would have been completed if time was available.

If we had the testing to do over and, once again, had a better knowledge of our technology and were allowed more time we could have easily spent more time on repeated runs to measure data and find bugs. Besides more range testing, we would have liked to test out an actual situation that a customer might do. In the lab we did all our testing on a flat table top with nothing in between the transmitter and receiver. Obviously this is not a real case scenario. A user is probably going to want to conceal the transmitter device by placing it under a couch or behind some object that makes it out of view for someone to see. This kind of obstruction will lead to interference with the RF transmission and cause the receiver to not receive the signal or receive a garbled signal. This will only confuse the owner and cause him to lose trust in our product.

Only by extensive testing will we truly get to know our product and root out any deficiencies it may have. If we did not run into issues that caused us to lose time we would have spent more time on this pressing issue and possibly have made a better product.

8 Budgets

For this project we were given a building budget of $500. This amount was for parts, service, and other goods that were purchased with the intent of aiding in the success of this project. We compiled our goods by either purchasing them through online vendors or, like most of our IC's, we ordered samples that cost the team nothing.

Below is a complete list of all the products we ordered:

Product	**Cost**	**Vendor**
Pickit2 Programmer	$97.38	Microchip Direct
RF Transmitter (2) LR-315	$14.92	Digi-Key
RF Receiver (2) LR-315	$27.12	Digi-Key
Antenna (5) 315 MHz	$4.40	Digi-Key
16MHz Ceramic Res.	$1.47	Digi-Key
USB Type B Recept.	$4.02	Digi-Key
PIC18LF2455 Chips	$34.55	Microchip Direct
Strip Sockets Sing. Row	$1.95	Mid-State Supply
4 Packs AA Batteries	$10.83	Radio Shack
Battery Holder w/ Switch-2	$4.30	Radio Shack
PicKit2 Programmer	$97.38	Mouser

As you could have noticed we did not include some of the IC's such as the other PIC's, EEPROM's, voltage regulators, timers, resisters, etc. All of these products were either available to us by free sample from the original company (Microchip or Texas Instruments) or were attained free from the EE department. As you can see we have listed the Pickit2 programmer twice on the list. Near the end of our project our programmer short circuited on us and we had half a week until the demo. We talked to the EE department about our issue and they would let us program on theirs, but only on their site for a brief moment, meaning we would have to go back and forth between the departments to program and debug. This was highly unacceptable and we were forced to order another program and overnight it. We went to www.mouser.com, since they have a shipping location in Round Rock, TX and ordered the part again. So, technically the additional $100 charged to our budget never should have happened. It is just part of the extenuating circumstances groups run into.

We used other code in the production of this lab. The bulk of the USB and EEPROM code was provided by either an online source or directly from Microchip. Both of these sourced give permission to use their code in your project provided you give due credit. Either way, the code is generic enough to implement the portions yourself, given more implementation time.

The nicest part about our project is the reasonable cost of final production. Most pet deterrents on the market cost around $100 and do not do nearly as many things as ours does. Our boards only use a few ICs total, which allows for an affordable solution for pet-owners.

The following charts outline the specifics of our final solution in mass quantities.

The receiver cost analysis is listed below.

Product	Cost	Vendor
PIC 18F2455	$4.25	Microchip Direct
24LC64 EEPROM	$0.46	Microchip Direct
Linx RXM-315 Receiver	$13.56	Digi-Key
Voltage Regulator	$0.40	Texas Instruments
Resistors, Capacitors, etc.	Negligible	Various
Hybrid Dog Collar w/ Pop off switch for receiver	-	-
USB Connector	$1.34	Digi-Key

So, excluding the hybrid the dog collar that needs to be made for our project, the total cost of parts for the receiver in mass quantities would be $20.01. The hybrid dog collar we would like would be similar to a normal collar, except that it would have a connector on it where we could remove the receiver portion and the user could take it to their computer to upload the data.

The transmitter portions can be seen listed below.

Product	Cost	Vendor
PIC 18F2455	$4.25	Microchip Direct
DIP Switches	$2.79	Radio Shack
Linx TXM-315 Transmitter	$7.46	Digi-Key
Voltage Regulator (3)	$0.40	Texas Instruments
Resistors, Capacitors, etc.	Negligible	Various
Potentiometer	$0.85	Digi-Key

The transmitter, excluding outer casing, will cost around $15.75 in core parts. When you combine this with the receiver you are looking at around $40 in assembly for one transmitter and one receiver. These products have simple schematics and could be mass-produced rather quickly and efficiently. This is pretty close to our presumed budget in which we estimated a low cost outcome that would be well short of the $500 limit.

9 Appendices

Issue of Multiple Transmitters Operating in the Same Range

In our intent of design we meant for each transmitter to operate within its region, but it is very possible for a customer to place two transmitters close enough to each other that a receiver could be in both zones at the same time. RF communication in itself is very noisy and picks up a lot of interference. It is not uncommon for a signal as weak as ours (which transmits as low as 3 inches) to be interfered with by any sort of object and cause the system to malfunction and give bad data. So, when two transmitters are sending data off of each other it is virtually impossible for the receiver to parse. The best solution for this is to implement random sleep times in between data transmissions. Since the data stream is received and parsed in a few milliseconds, a sleep time range of 1-5 seconds chosen at random will be enough to throw off the transmitted collisions and keep consistency.