GPS Dog Shock Collar

Design Review
October 4, 2012

Team 23
Jacob Hardy & Joshua Passwater
T.A. Justine Fortier

Table of Contents

Introduction

<u>Concept</u> : Most pet owners appreciate the safety of knowing where their pet is at all times. A modern pet containment solution is a Shock Collar. A user sets up magnetic posts or buries a wire in their yard which serves as an invisible boundary against a pet's movement. When the pet crosses the threshold between two posts, a sensor in the collar is alerted and a training pulse causes displeasure to the dog. One problem with this design is that a stubborn dog could cross the threshold and be free of its perimeter. Another problem with this design is that only one perimeter can be set up for the pet, and it must be set up using a cumbersome outdoor fence system. Our idea is to replace the standard "fence" idea with GPS coordinates as set by the user. Not only is there no physical fence setup, but now multiple user locations can be specified for different occasions such as the user's home, office, or favorite park. Multiple perimeters can be saved on the collar and chosen for the proper occasion, and the boundaries apply beyond the limits of the physical fence setup.

<u>Behind the Concept</u> : Instead of the standard magnetic sensing on current dog collars, ours will use GPS coordinates to achieve the same affect. The GPS module will receive the current location from GPS satellites; transmitting the data via the NMEA-0183 serial protocol to a microcontroller. The microcontroller is in charge of interpreting all of the data given to it; determining whether the shock mechanism should be activated and making sure that the shock collar works only over certain intervals so as not to unjustly punish the pet. These coordinates will be written to RAM in the microcontroller using a couple possible methods:

- Instant: Triggering of a barrier with a certain radius as specified by a dial on the computer. A dial on the collar with selectable radii and an activation button will trigger an instant perimeter. This feature is beneficial when on vacation somewhere such as the beach and you want an instant perimeter for your pet. Good for small/medium portable perimeters.
- Corner Set Point: A user will put the collar into set mode upon which time he/she will walk around the perimeter to set the various corners of the perimeter. Pressing the corner activation button will trigger a new corner point for the perimeter. Good for medium/large perimeters.

In addition to GPS, there are a few other features that we planned on implementing, but due to the time restraints of this class have only chosen one for the time being: detection of nearby pets wearing the same type of collar. A small transceiver antenna in the collar will transmit each dog's unique I.D. to a small proximity radius. The same transceiver antenna in another pet's collar will pick up the I.D.'s of other pets in the near vicinity and determine whether contact is allowed between these two pets. The factor of whether two pets are allowed to interact or not comes from the user's selection to put certain pet's I.D. tags on a "No-Contact List" within the RAM on the pet's collar. The prohibited list will be programmable via the computer user interface. The option of preventing contact with all dogs wearing this collar will also be available. Some other features that could possibly be implanted into the collar include barking & jumping detection, and a possible GPS locator in case of a missing pet. The barking detection and GPS locator already exist as stand-alone products, but if all of these are implanted in one collar, then we could see potential in having an all-in-one training solution. Users could even select different models with different amounts of features at a lower cost to suit their needs.

The shocking mechanism is not a novel factor to the concept. In the sense of avoiding "reinventing the wheel", the shocking portion will not be implemented at this time. An analog output on the Microcontroller will be left open for future shock addition. Instead, two LED indicators and activation sounds will be implemented to provide warning to the pet and owner of an infraction before and during the shock. The collar will go into Warning Mode whenever an infraction is being committed such as being outside of a boundary or interacting with a prohibited pet. In Warning Mode, the LED and an associated buzzer will flash at roughly 1 Hz, much like when a garbage truck or school bus goes in reverse. This Warning Mode will serve as an indication to refrain from such activity before the shock occurs. Ivan Pavlov's Classical Conditioning proved that dogs can be trained to react to a stimulus when repeated multiple times preceding a constant result. Why shock when a simple buzzer sound will do? After a predefined amount of time (currently five seconds) without proper reaction to the Warning Mode, the pet will be shocked and a separate sound buzzer will go off simultaneously. The Pavlovian conditioning will take over and the pet will eventually learn to react before the shock. Another preventative measure to over-punishment is our method to prevent too many shocks. Instead of blindly shocking the dog for hanging around his barrier, the internal software will have delays between shocks and will eventually stop after too many consecutive shocks in case of an improperly set boundary. This will prove to be another advantage over the current shock collars on the market today which have no method of preventing repeated shocks.

Features :

- GPS-based perimeter detection for pet confinement
- Multiple user-programmable perimeters
- Multiple perimeter entry modes including Instant and Set-Point perimeters
- Prevention of unwanted contact with other pets as specified by the user
- Adjustable shock levels
- LED/sound shock indicators
- Battery powered
- Connects to computer via USB
- Computer user interface

Benefits :

- GPS Accuracy (the Venus model has 2.5m accuracy with up to a 20 Hz refresh rate)
- No physical fence setup which allows for larger and more diverse boundaries
- Multiple, configurable, and switchable perimeter locations
- Preventative measures beyond the theoretical boundary
- Usable by dogs, cats, exotic animals, and even flying animals.
- Inter-pet contact prevention
- Future component expandability & Improvements
  - Location known and retrieval of lost pet through added communication with possible host station
  - Jumping detection
  - Barking detection
  - Different punishment mechanism such as spray
  - User community with uploadable, shareable perimeters such as local parks
  - Rechargeable battery

Design

Component Diagram :

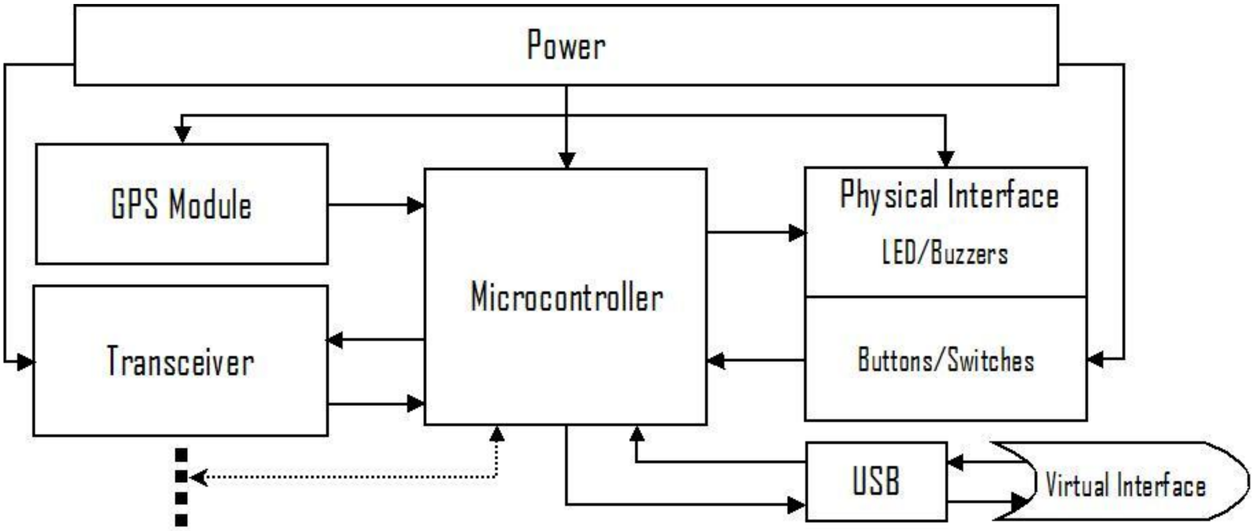


Figure 1

Component Descriptions :

- Power Source – Battery/Power Source for all other components in the project. This is run via 4 AAA batteries for 6V power. Immediate reason for use of AAA battery power is ease of use for consumers.
- GPS Module – Receives and transmits current coordinates to the Microcontroller portion for cross-reference against non-allowed zones. The Venus GPS-11058 transmits current coordinates via the NMEA-0183 protocol, has a 20Hz refresh rate, and is accurate within 2.5 meters.
- Transceiver – Sparkfun nRF24L01+: Transmits the unique ID of the pet collar and identifies others in the near proximity.
- MicroController – An Arduino-based ATmega-328P microcontroller: Takes input from the various detectors and references the physical interface buttons, switches, and knobs to determine if a violation is in progress while transmitting the unique pet I.D. through the Transceiver. It controls the alert/shocking system, stores GPS coordinates in its onboard RAM, and communicates to the Virtual Interface via the USB hub to receive programmed perimeters and prohibited pet I.D.'s. Will be programmed mostly in C with some additional subcode provided by the Arduino community. Refer to Software Flow Chart in Figure 3.

- Physical Interface: LED/Buzzers – As described in the introduction, the alert/shocking portion will be represented by two sets of LED's and sound buzzers. One buzzer will be in sync with the yellow LED and the other buzzer with the red LED. The yellow combo will appear in a repeating pattern for Warning Mode i.e. whenever the pet is concurrently infracting on either its perimeter or prohibited interactions and will only stop whenever returning to an area without infraction. The red combo will appear whenever the physical shocking should occur. The shocking mechanism will not be implemented at this time but a Microcontroller analog output will be left open for the future addition.
- Physical Interface: Buttons/Switches – The physical interface will feature a button, switch, and rotary knob configuration as approximated in Figure 2. This will be the interface for human interaction with the device when it is in use.
- USB – The USB interaction circuit with an outboard TTL Serial port for connecting to a USB port on a standard home computer. Data will transfer between the microcontroller and the virtual interface via the USB hub.
- Virtual Interface – A program installed on the pet owner's computer that communicates with the collar via the USB. Will be capable of input of pet I.D.'s and GPS coordinates. Hopefully, a future implementation would allow for a user community for sharing of opinions, common perimeters such as local parks, and their unique pet I.D.'s. It will be programmed in Microsoft's Visual C# 2010 Express.
- Extension Space – Room in microcontroller for further pet-training additions such as barking correction, jumping correction, and location of lost pets.

Project Requirements :

- Portable and compact, i.e. able to fit on a pet's collar.
- Consumer-friendly, easy to use, and marketable to a pet-owning community.
- Easily powered. AAA battery method preferred for now for ease of use.
- GPS accurate to a reasonable proximity, preferably 10 feet or less within perimeter.
- GPS functions in a vast majority of locations.
- No false infractions.
- Accurate transmission and detection of pet ID's through transceiver.
- Reasonable pet interaction distance for transceiver propagation must be achieved.
- Reprogrammable microcontroller capable of storing user's information.
- User interfaces, both Physical and Virtual, are intuitive to the average consumer.
- Controllable shocking mechanism that proves to not continuously shock, or shock in between intervals of 3 seconds. Needs to stop after a reasonable amount of time so as not to potentially shock pet for more than 5 iterations.
- Warning sound before shock with LED status indicators (yellow for out of proximity and red during the actual shock) and buzzers (different sound for each).
- Safe testing (probably on ourselves at first).
- Safe to all pets (dogs the main consideration at the moment).

Diagrams

Physical Controller & Interface :

Mounted as part of the physical device on the collar, the controller offers an array of buttons, switches, and knobs. Allowing for direct contact proves to be more useful for the average consumer, and the ability to set both instant-radius and corner perimeters requires a portable interface that travels with the GPS so it makes sense to attach it to the actual collar for ease of use and portability. The collar module will be the same size as the battery holder for aesthetics. In Figure 2, the switch sizes and layouts are approximated but they should easily be able to fit into the space provided with extra room for the logo.

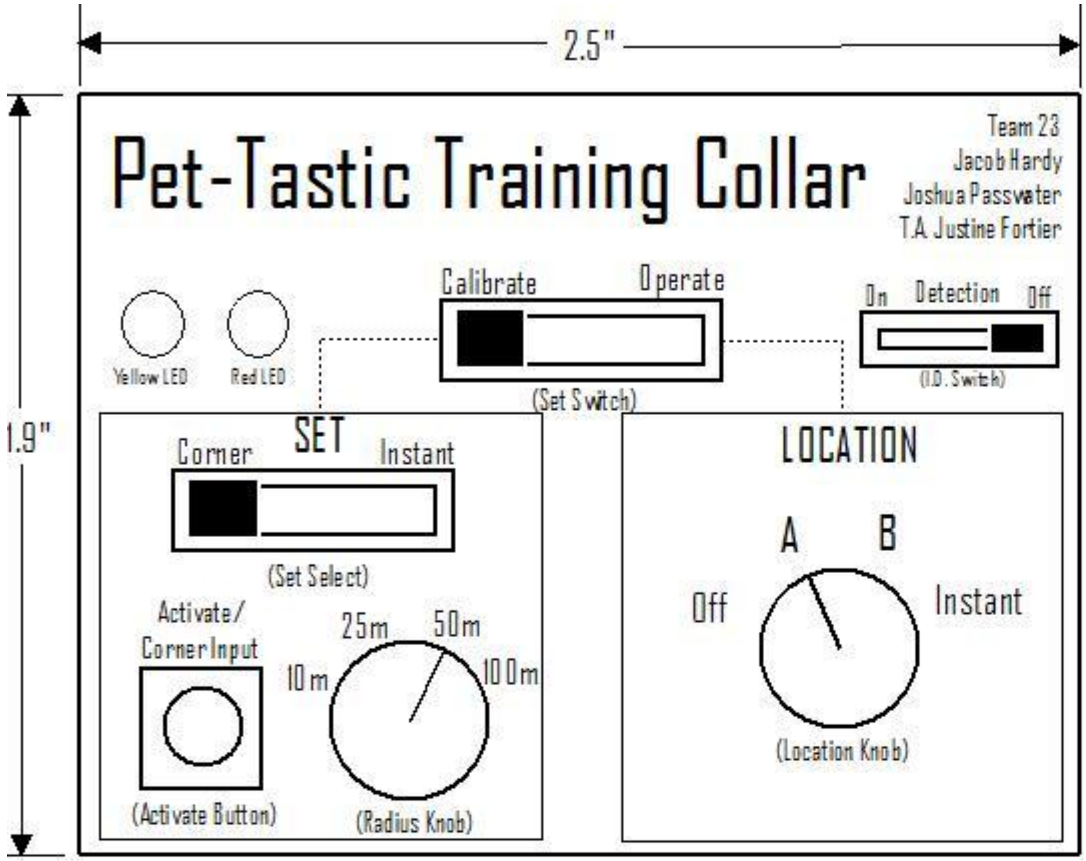


Figure 2

- Set Switch – Operational switch that allows user to define whether they will be setting up a perimeter or using an existing one. It switches between the Calibrate mode where the locations are set and the Operate mode which puts the collar into operation using the currently set location.

- The Location Knob is a user-selective rotary switch that chooses between four locations: Off, A, B, and Instant. Off specifies that there is no current perimeter so the dog is free to roam wherever. Off would prove useful for users only looking for the pet interaction prevention. A and B are perimeters put into the RAM using the Corner-Set input method. They are both over-writable, reconfigurable, and able to be stored even when the collar is turned off. The instant location refers to the last location set up using the Instant-Radius input method.
- The Set Select switch allows the user to specify whether they will be updating perimeter using the Corner-Set or Instant-Radius input method. The Location Knob must be set accordingly as it specifies what location in memory will be over-written. The Set Select switch is only relevant when the Set Switch is set to Calibrate.
- The Radius Radius selects the radius size for an instant perimeter when the Set Select is put into Instant.
- The Activate Button works for both Set Select Modes. In Instant Mode, it activates a radius-based perimeter around the current location at whatever radius is currently selected by the Radius Knob. The Instant Location is then over-written. In Corner Mode, the Activate button selects each corner location into memory, in order, as a user walks around the outside of whatever perimeter they choose. As the user traverses the perimeter, they will press the button at each corner they reach. After four corners, the internal software will determine the GPS coordinates for the perimeter and over-write either A or B depending on which is currently selected by the Location Knob. While the user traverses the perimeter, warning mode is enabled to let the user know that he/she is not yet finished entering the four corners.
- The I.D. switch allows the user to specify whether they would like to use the pet identification feature on the collar.

Software Flow Chart :

Figure 3 on page 11 shows the Internal Microcontroller software flow chart that will take input from the physical interface to control the collar. Figure 4 on page 12 enhances the perimeter input portion and Figure 5 on page 13 enhances the operational mode portion. The flow chart is analogous to the Physical Interface in terms of layout.

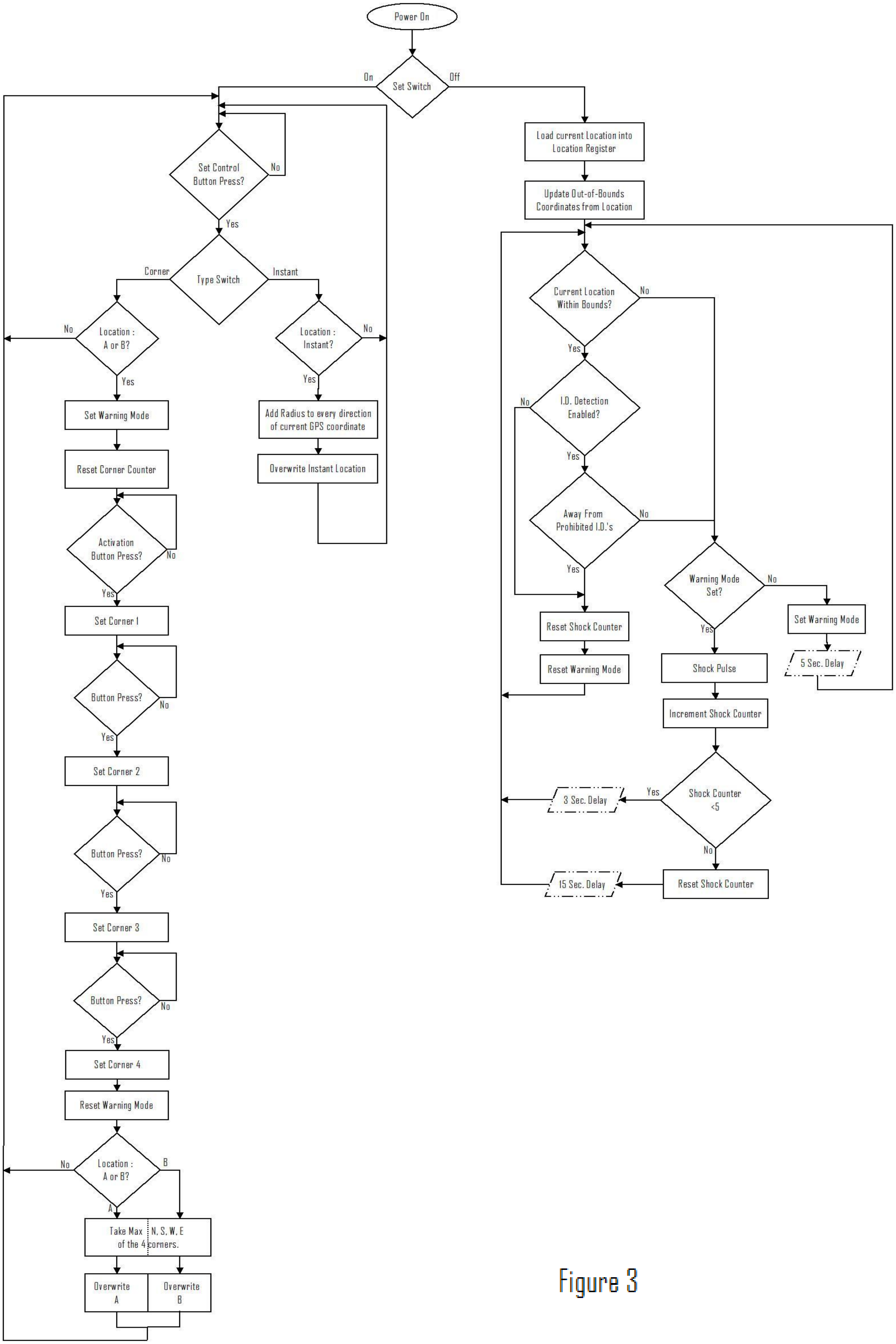

Power On
Set Switch
On
Off
Set Control Button Press?
No
Yes
Type Switch
Corner
Instant
Location : A or B?
No
Yes
Set Warning Mode
Reset Corner Counter
Activation Button Press?
No
Yes
Set Corner 1
Button Press?
No
Yes
Set Corner 2
Button Press?
No
Yes
Set Corner 3
Button Press?
No
Yes
Set Corner 4
Reset Warning Mode
Location : A or B?
No
A
B
Take Max of the 4
N, S, W, E corners.
Overwrite A
Overwrite B
Location : Instant?
No
Yes
Add Radius to every direction of current GPS coordinate
Overwrite Instant Location
Load current Location into Location Register
Update Out-of-Bounds Coordinates from Location
Current Location Within Bounds?
No
Yes
I.D. Detection Enabled?
No
Yes
Away From Prohibited I.D.'s
No
Yes
Reset Shock Counter
Reset Warning Mode
Warning Mode Set?
No
Yes
Set Warning Mode
5 Sec. Delay
Shock Pulse
Increment Shock Counter
Shock Counter <5
Yes
No
3 Sec. Delay
Reset Shock Counter
15 Sec. Delay

Figure 3

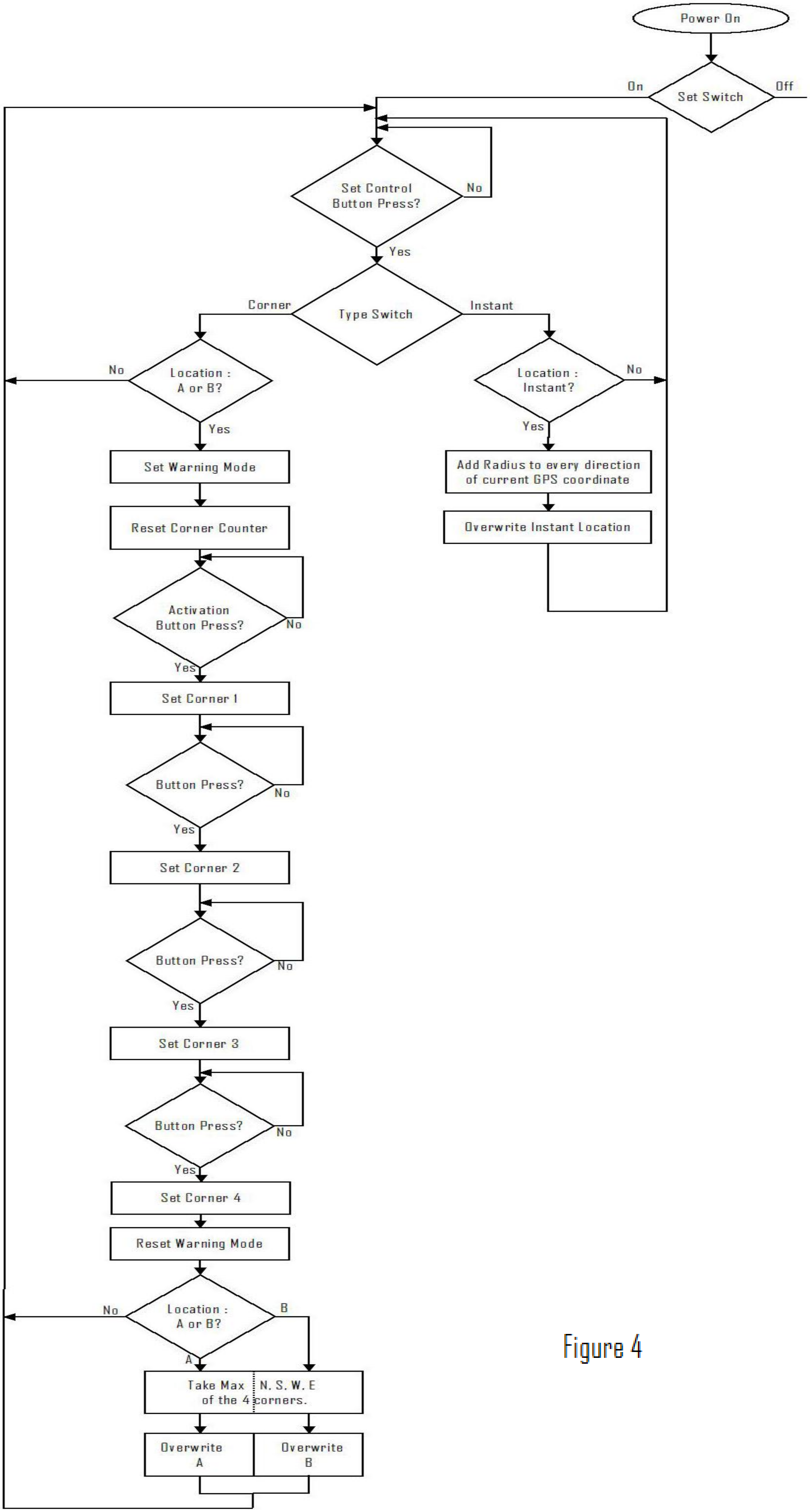

Power On
Set Switch
On
Off
Set Control Button Press?
No
Yes
Type Switch
Corner
Instant
Location : A or B?
No
Yes
Location : Instant?
No
Yes
Set Warning Mode
Add Radius to every direction of current GPS coordinate
Reset Corner Counter
Overwrite Instant Location
Activation Button Press?
No
Yes
Set Corner 1
Button Press?
No
Yes
Set Corner 2
Button Press?
No
Yes
Set Corner 3
Button Press?
No
Yes
Set Corner 4
Reset Warning Mode
Location : A or B?
No
B
A
Take Max of the 4
N. S. W. E corners.
Overwrite A
Overwrite B

Figure 4

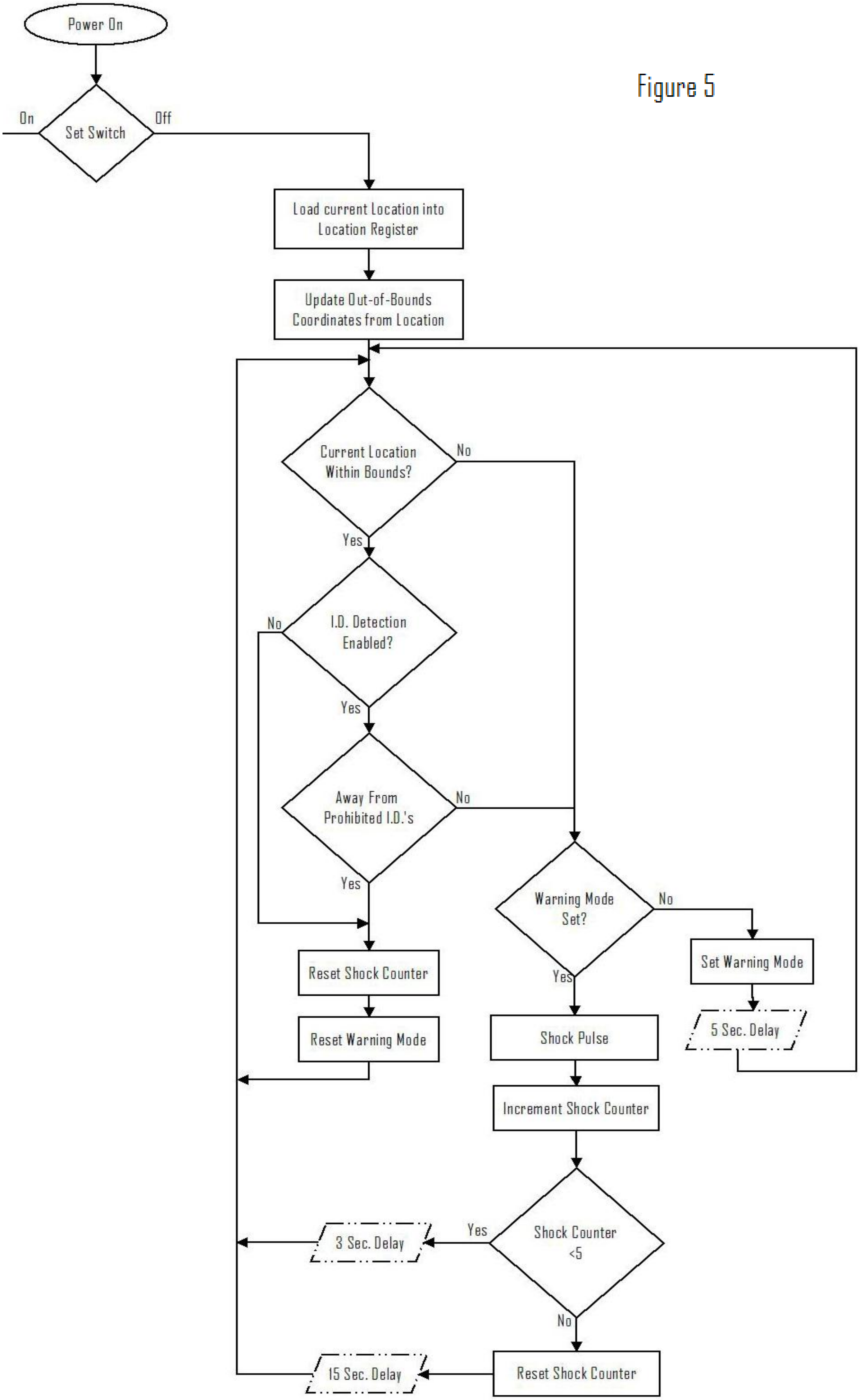
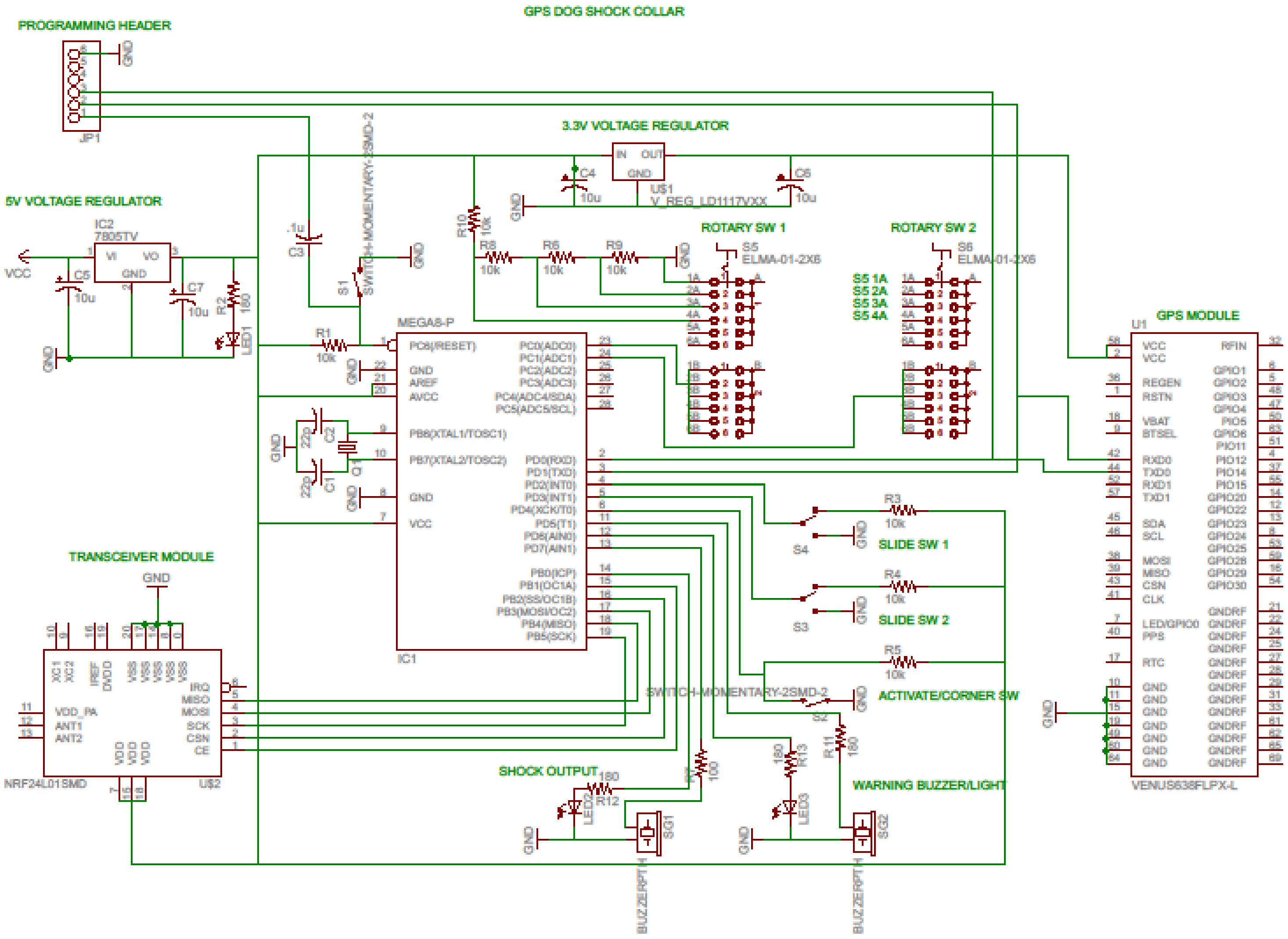

Power On
On
Set Switch
Off
Load current Location into Location Register
Update Out-of-Bounds Coordinates from Location
Current Location Within Bounds?
No
Yes
No
I.D. Detection Enabled?
Yes
Away From Prohibited I.D.'s
No
Yes
Warning Mode Set?
No
Yes
Set Warning Mode
5 Sec. Delay
Reset Shock Counter
Reset Warning Mode
Shock Pulse
Increment Shock Counter
3 Sec. Delay
Yes
Shock Counter <5
No
15 Sec. Delay
Reset Shock Counter

Figure 5

Schematics



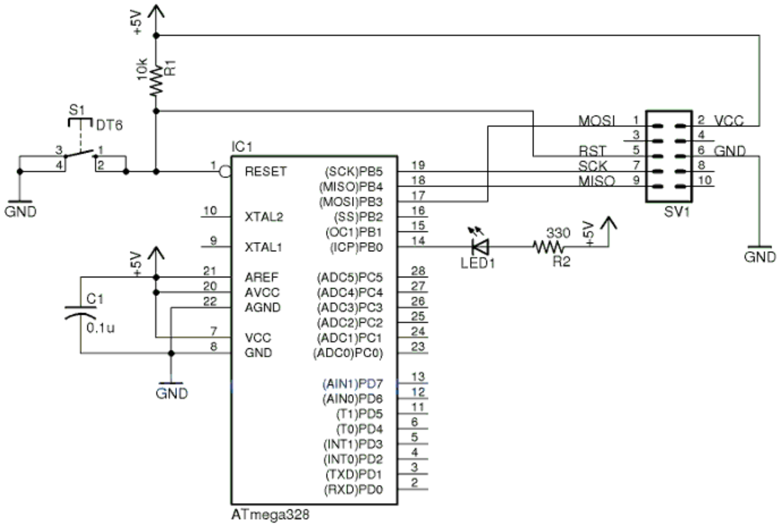
10/3/2012 3:21:51 PM C:\Users\Josh\Documents\eagle\ece445\ECE445.sch (Sheet: 1/1)

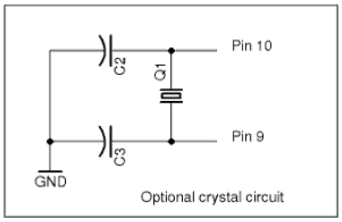
Figure 6

Figure 6 is a simple EAGLE implementation of the physical component outlay/PCB we plan on creating for the actual device imbedded in the pet's collar. Each component is reasonably small for portability. The only concern might arise when certain interconnections impede a certain placements of components, like the switches, that will alter the physical look of the physical interface.

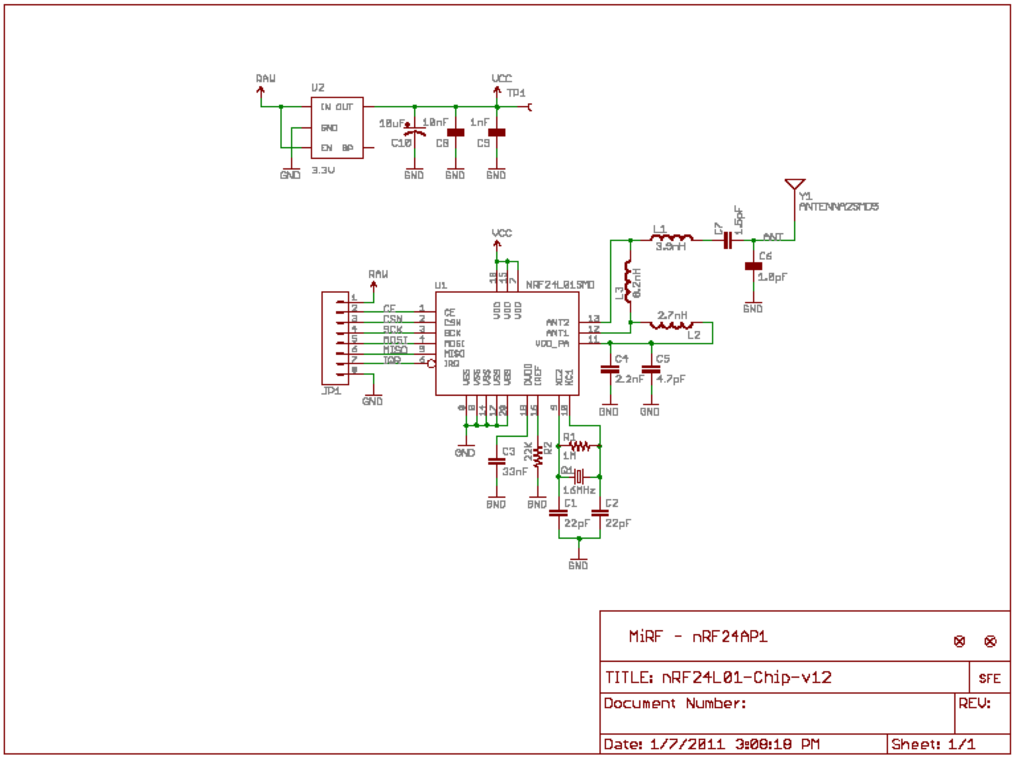
Part Schematics :

Microcontroller

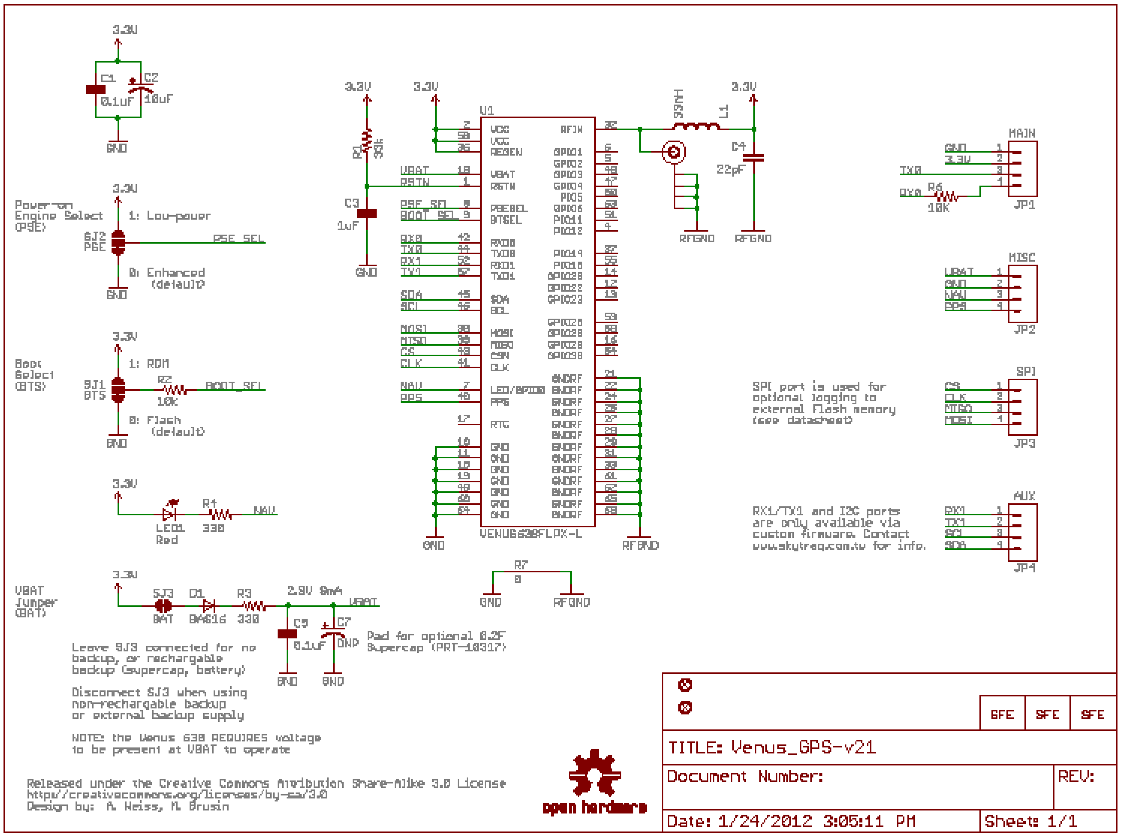




Transceiver



GPS


3.3V
C1
0.1uF
C2
10uF
GND
Power-on Engine Select (PSE)
1: Low-power
0: Enhanced (default)
SJ2 PSE
PSE_SEL
Boot Select (BTS)
1: ROM
0: Flash (default)
SJ1 BTS
R2 10k
BOOT_SEL
LED1 Red
R4 330
NAV
VBAT Jumper (BAT)
SJ3 BAT
D1 BAS16
R3 330
2.8V 9mA
VBAT
C5 0.1uF
C7 DNP
Pad for optional 0.2F Supercap (PRT-10317)
Leave SJ3 connected for no backup, or rechargable backup (supercap, battery)
Disconnect SJ3 when using non-rechargable backup or external backup supply
NOTE: the Venus 638 REQUIRES voltage to be present at VBAT to operate
R1 39k
C3 1uF
U1
VCC
VCC
REGEN
VBAT
RSTN
PSESEL
BTSEL
RXD0
TXD0
RXD1
TXD1
SDA
SCL
MOSI
MISO
CSN
CLK
LED/GPIO0
PPS
RTC
GND
RFIN
GPIO1
GPIO2
GPIO3
GPIO4
PIO5
GPIO6
PIO11
PIO12
PIO14
PIO10
GPIO20
GPIO22
GPIO23
GNDRF
VENUS638FLPX-L
RFGND
R7 0
33nH L1
C4 22pF
MAIN
GND
3.3V
TX0
RX0
R6 10K
JP1
MISC
VBAT
GND
NAV
PPS
JP2
SPI
CS
CLK
MISO
MOSI
JP3
SPI port is used for optional logging to external Flash memory (see datasheet)
AUX
RX1
TX1
SCL
SDA
JP4
RX1/TX1 and I2C ports are only available via custom firmware. Contact www.skytraq.com.tw for info.
Released under the Creative Commons Attribution Share-Alike 3.0 License
https://creativecommons.org/licenses/by-sa/3.0
Design by: A. Weiss, M. Grusin
open hardware
SFE
SFE
SFE
TITLE: Venus_GPS-v21
Document Number:
REV:
Date: 1/24/2012 3:05:11 PM
Sheet: 1/1

Verification

Development Process

The product development will move through phases based on the importance of certain components within the overall design. Since the GPS application is our novel idea, we will first work to achieve that; first only with the Instant perimeter method. Preliminary code will be written (mostly in C) and loaded onto the microcontroller circuit we will build on the breadboard. Proper barrier detection will be indicated through the LEDs and buzzers. After GPS, we will move on to the antenna calibration. This antenna portion will require another test collar that will be stripped of GPS and built on the breadboard for monetary sake. This is justifiable by the fact that the select collar is there strictly for pet proximity testing. Also, the antenna portion will require a computer user interface design to adjust pet I.D.'s. The virtual interface will be programmed before initial transceiver testings. All of these features will be combined near the end to achieve the portable device.

Modular Requirements

::GPS::	
Requirement	Verification
1.1 Must receive coordinates from global positioning satellites.	1.1 We will first test the GPS on the breadboard observing the output on the oscilloscope. Using the NMEA-0183 serial protocol and a website such as Google Maps for reference, we will cross check that the coordinates received from the Venus and observed on the Oscilloscope coincide with the results from Google Maps.
1.2 Must update coordinates with movement.	1.2 As our collar will not be mobile at this point in time, we will need a very long extension cable. For this verification we will have had to have set up the communication between the microcontroller and the GPS unit. A small subroutine will be written for the microcontroller that strictly stores the current GPS coordinates received in to the EEPROM as they change from location to location. Using the long extension cable we will walk down the hallways of Everitt as the microcontroller should be storing the updated coordinates.
1.3 Must be accurate enough for reasonable boundary detection.	1.3 By now the rig should be mobile and attachable to a pet's collar, Using the same code subroutine as in 1.2 we will test the GPS's ability to update coordinates and its accuracy in between coordinate updates. While the GPS is rated at 2.5m accuracy, we will measure the exact accuracy for

	determining a proper refresh rate as outlined in the Calculations section. We will need to determine the significant figures for exact accuracy of the GPS coordinates.
::Transceiver::	
Requirement	Verification
2.1 Must communicate with the same model of transceiver.	2.1 We will set up both transceivers on the breadboard each properly grounded and powered at 5V with a signal generator attached to the input of one and an oscilloscope attached to the output of the other. Using a 2.785 GHz (our selected frequency of operation so as not to coincide with other bandwidths) square wave at the input and observing the output at the oscilloscope connected to the out of the second transceiver.
2.2 Must transmit only enough power to be received within a small radius similar to the interaction of pets.	2.2 This will take rigorous trial and error of various transmit and supply powers to each of the transceivers so that they barely sense each other at roughly 1'6" apart. After ensuring 2.1, we will set up the transceivers roughly a foot and a half apart and reduce the output of the first transceiver until it is not detected by the second transceiver. This will dictate the power output threshold needed to achieve short range communication.
::Microcontroller::	
Requirement	Verification
3.1 Must be reprogrammable onboard.	3.1 A code subroutine will be written that can increment the contents of a specific location

	in EEPROM. This can simply be tested by making a push button command subroutine that will increment a counter in EEPROM. The results will be observed by connecting the microcontroller to the computer via USB and checking the contents of EEPROM.
3.2 Must accept and receive all proper I/O from other components. - 3.2(a) GPS	3.2(a) Using the same subroutine programmed for 1.2, this will be the portion where we see if the actual coordinates are written in to the EEPROM. We will walk around with our semi-mobile rig and examine the contents to see if they update with new coordinates.
- 3.2(b) Transceiver	3.2(b) Using a similar technique as in 2.1 except now with the addition of the microcontroller. Replacing the oscilloscope from 2.1 with the microcontroller and observing the input via the USB connection ensures proper output from the transceiver. Replacing the signal generator with a pre-specified square wave as programmed onto the microcontroller with insure proper input into the transceiver.
- 3.2(c) Buttons & Switches	3.2(c) First proceed with 4.1 to make sure that all of the switches are in physical working order. Using the same code subroutine as in 3.1, implement each button and switch individually as a means of incrementing a counter in the EEPROM. Try each button with its respectively assigned I/O port on the microcontroller and change

- 3.2(d) Power	the subroutine accordingly. 3.2(d) This should be easily testable as the microcontroller specifications list specific power supply values for microcontroller operation. Simply set up accordingly and see if the microcontroller turns on using the preprogrammed LED confirmation included with the Arduino programming when the microcontroller is booted for the first time.
::Physical Interface::	
Requirement	Verification
4.1 Physical switches must properly relay decision choices to microcontroller in a reliable and predictable fashion. 4.2 Must be configurable in such a way as to be aesthetically reasonable to a customer and configurable on a PCB.	4.1 Simple breadboard checking of proper connections between DC sources at the input of the switches and multimeters at the output. All of the switches function by making a physical connection in place of a short and are therefore easily tested with the breadboard. 4.2 This will require EAGLE testing to see how compact we can fit the switches without interfering with necessary space for the GPS, transceiver, and microcontroller portions. Aesthetics will suffer for the sake of functionality.
::Virtual Interface::	
Requirement	Verification
5.1 Must properly communicate decisions through the USB driver to the microcontroller.	5.1 Simple testing to see if input on virtual interface can store values in the EEPROM of the microcontroller. This will require verification from the virtual interface running in parallel with the Arduino programming interface. A change in the

5.2 Must be user-friendly.	virtual interface should reflect changes in the EEPROM as viewed from the Arduino interface. 5.2 This will require the help of outside criticism to aid us in deciding, by popular poll, whether some features are too complicated or simple for the pet-owner's needs. The virtual user interface will be designed and programmed, test-users will evaluate its functionality and ease-of-use, and we will reprogram as necessary.
::Power::	
Requirement	Verification
6.1 Must properly supply power to all components in need.	6.1 Virtual simulations point out that we should be fine in terms of power, but if not, then we will simply get a larger, more expensive power source. Testing will consist of replacing the breadboard-supplied power with our simple AAA battery holder and observing the results. As needed, various multimeter readings will help determine any source of over-usage of power. The power values supplied in the power allocation table are when the components are pushed to the extreme. As is the case with the transceivers, we will not be running our components at anywhere near maximum capacity (except maybe the LEDs for maximum visibility).

Calculations

The PCB circuit will basically consist of our interconnected modules and few other additional components such as pull-up resistors, current limiting diodes, and filtering capacitors. Switching mechanisms such as the rotary knob will use Ohm's Law

$$V = RI$$

to produce variable amounts of current from the same voltage source with varying resistor values. One issue with our components is that some require different supply voltages than others. The batteries supply 6V which directly goes into the 5V regulator to bring it down. From there, we will use a resistor network capable of splitting up 5V into the various voltages needed for different components using the voltage divider rule.

$$V_{new} = \frac{R_1}{R_1 + R_2} V_o$$

Much of our system consists of digital information with varying amplitudes but this can be controlled in the internal software on the Microcontroller as it will be receiving and transmitting the data through the GPS module and the Transceiver. One worry however with those transmission methods comes from actual power transmitted and received. The VSWR refers to the amount of received power by

$$VSWR = \frac{V_{max}}{V_{min}} = \frac{1 + \rho}{1 - \rho} < 2$$

where ρ is the magnitude of the reflection coefficient and where the VSWR of the transceiver is rated at less than 2. By calculation, ρ will be less than $\frac{1}{2}$ so we will have at least half of our power transmitted. Physical transceiver power output will have to be tested once we get the physical transceiver in the mail.

One consideration to consider is that of what refresh rate will we use for the GPS. It is rated at 2.5m accuracy with a maximum refresh rate of 20 Hz. For the maximums to be of any use the pet would have to move at 2.5m every 1/20th of a second

$$\frac{2.5m}{\frac{1}{20}sec.} = 500\frac{m}{s} = 1118.47\frac{miles}{hour}$$

which is obviously unnecessary for any animal. The fastest land animal is the cheetah which runs at 61 mph. By our calculations, if someone was to use our collar for a pet cheetah, the maximum refresh rate would need to be

$$61\,\frac{miles}{hour} = 27.27\,\frac{m}{s} = 10.9\ Hz\ with\ a\ 2.5m\ accuracy$$

where 10 Hz will prove to be a very logistical improvement (power and computation-wise) over 20 Hz.

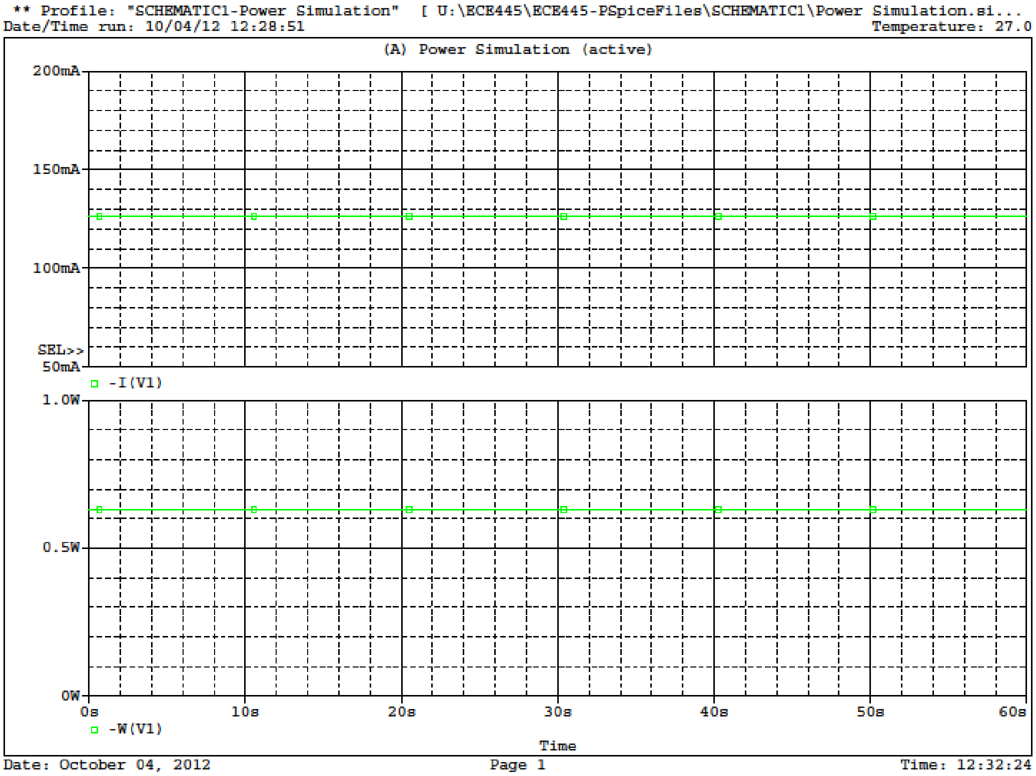
The transceiver's functioning built-in antenna will serve as both the transmitting and the receiving end of the communication process. We have decided to go with a process of alternating ½ second intervals of transmission and reception i.e. the transceiver will receive for half a second, transmit for half a second, receive for half a second, etc. The transceiver repeatedly transmits $B\ = 48$ bits (24 ID bits for six hexadecimal characters and 24 string code identification bits) at a data transfer rate of t_r over the duration of the ½ second interval. There is a risk of two collars in "sync". This would mean that the two collars near each other would be perfectly synced to transmit and receive at the same exact time; therefore, never registering each other's presence. The odds of this happening however are

$$\frac{2B}{t_r} * 100\%$$

so a higher data transfer rate would greater decrease the chance of collar "sync". An example t_r of 9600 bits/sec would produce a 1% chance of a miss. t_r can be greatly increased as needed. The operational frequency of the transmitters will be at 2.785 GHz.

Power Simulation

With every component running at maximum power usages, we will draw roughly 130 mA. With 1000-1200mAh supplied by AAA batteries, the absolute minimum battery usage is 7.69 hours. We project over fifty hours of usage with 4 batteries and the surprisingly minimal amount of power that the shocking portion uses. While this might not seem ideal for a product that will theoretically be on 24 hours a day, it works fine for the current task at hand of developing a functioning GPS collar. Future iterations would have a rechargeable battery.



```
** Profile: "SCHEMATIC1-Power Simulation"   [ U:\ECE445\ECE445-PSpiceFiles\SCHEMATIC1\Power Simulation.si...
Date/Time run: 10/04/12 12:28:51                                                        Temperature: 27.0
```

```
Date: October 04, 2012                    Page 1                    Time: 12:32:24
```

Preliminary Part Statistics :

::Unit::	::Relevant Statistics::
Venus 638FLPx GPS	20 Hz refresh rate 9600 bits/second 2.5m accuracy Separate Tx and Rx I/O Optional Internal Flash 1.15" x 0.7"
GPS Antenna	VSWR < 2 26 dB amplifier gain & 3 dB antenna gain Roughly 1" x 1" antenna with a 6" cable
nRF24L01+ Transceiver	Built-in antenna 2400-2525 MHz (125 selectable channels) Separate MISO and MOSI I/O Data MISO/MOSI select, chip enable, clock, and interrupt On board regulator (3.3-7V) 32 Byte separate FIFO for Tx and Rx 0.8" x 0.7"
ATmega328P Microcontroller	Speed : Up to 20 MHz 6 Analog I/O 15 Digital I/O
Magnetic Buzzer (2400 Hz)	0.46"Diameter x 0.35"Height
Piezo Buzzer (4000 Hz)	0.54"Diameter x 0.27"Height
AAA Battery Holder	Holds 4 Alkaline AAA Batteries 2.5"x 1.9"x 0.6"
Physical Interface	2.5"x 1.9"x TBD"

Power Allocation :

Using 4 Alkaline AAA batteries rated at 1.5V and 1000-1200mAh each:

::Unit::	::Max Power Consumption::
Venus 638FLPx GPS	2.7-3.3V 68mA Input : 0 – 3.3V Output : 0 – 3.3V
GPS Antenna	3.3V ± 0.5V 12mA
nRF24L01+ Transceiver	3.3-7V 13.5mA Input : -0.3 – 5.25V Output : -0.3 – 3.6V
ATmega328P Microcontroller	Vcc 1.8-5.5V gives operational frequency of 4-20MHz 9mA max (5.2 typical) per I/O at high Hz
Switch Rotary Knob	30V 300 mA
Switch Slider	50V 500 mA
Switch Button	12V 50 mA
LEDs	2.8V 20 mA
Magnetic Buzzer (2400 Hz)	5 Vp-p for 85 dB 40 mA
Piezo Buzzer (4000 Hz)	5 Vp-p for 85 dB 5 mA
Everything should be operational at or around 3.3V with minimal current distribution worries for the provided battery supply. At any one time, a max current amount of 206.5mA will be flowing through the system supplied by the batteries, but this is assuming everything running at maximum capacity simultaneously. Power simulation in Figure	

Cost/Labor

Labor :

Name	Hourly Rate	Total Hours	Total = 2.5 * HR * TH
Jacob Hardy	$40.00	200	$20,000.00
Joshua Passwater	$40.00	200	$20,000.00
Total		400	$40,000.00

Total Hours = 20 hours/week * 10 weeks

Parts :

Part	Price/Part	Quantity	Cost
Dog Collar	$10.00	2	$20.00
Venus GPS : GPS-11058	$49.95	1	$49.95
GPS Antenna	$11.95	1	$11.95
Transceiver nRF24L01+	$19.95	2	$39.90
Barebones Arduino Kit	$14.95	2	$29.90
AAA Battery Holder	$1.45	2	$2.90
Button Single-Throw	$0.29	10	$2.90
Slide Switch	$0.39	4	$1.56
Rotary Switch 6 Position	$1.50	6	$7.50
LED – Yellow	$0.15	10	$1.50
LED – Red	$0.12	10	$1.20
Buzzer – Magnetic	$1.49	1	$1.49
Buzzer – Piezo	$1.25	1	$1.25
Shipping (So Far)	$13.64	N/A	$13.64
Small Plastic Enclosure	Machine Shop Free	2	$0.00 est.
PCB	$40.00	1	$40.00 est.
Resistors, Capacitors etc.	cheap	assorted	$10.00 est.
Total			$235.64 est.

Labor	$40,000.00
Parts	$235.64
Total	$40,235.64

Proposed Cost-to-Consumer Analysis

Using the parts we've purchased from commercial retailers, the grand total for one collar's parts is $118.01 without the PCB or chocking mechanism cost figured in. Assuming that each part roughly costs 20% of its MSRP to manufacture and adding in estimated PCB and shocking mechanism manufacture, and labor adjustment, we estimate that a collar would cost roughly $35.00 to make. Once again factoring in the 20% manufacture-to-market mark-up, we would price our collar at $175.00. The average dog shock collar on petsmart.com runs anywhere from $115 to $190. Considering the added features, ease-of-use, and product expandability, we see a viable commercial value in our product.

Proposed Schedule

Week	Responsibility	
9/16	Proposal Write-up & Overall Product Design	JH
	Parts Search, Documentation	JP
9/23	Parts Review & Order Parts	JH
	Microcontroller Planning & Ordering	JP
9/30	Design PCB & Document Parts Received	JH
	Order Microcontrollers & Design Overall I/O Schematic	JP
10/7	Learn EAGLE & Start PCB Design & Test Received Parts	JH
	Work on Microcontroller code & Test Received Parts	JP
10/14	Finish PCB design/Submit & Work on Microcontroller Code	JH
	Work on Computer Interface & Microcontroller Code	JP
10/21	Individual Progress Report & PCB Assembly	JH
	Individual Progress Report & Code work	JP
10/28	PCB Assembly/Reworking & Plastic Chassis Design for ECE Shop	JH
	RF Characterization/Optimization for Pet ID Detection	JP
11/4	Plastic Chassis Reworkings & Syncing RF, GPS, and Microcontroller & PCB Redo if Necessary & Mock-Up Demo	JH
	Syncing RF, GPS, and Microcontroller & Mock-Up Demo	JP

11/11	Final Product Assembly & Doumentation	JH
	Code Assembly and Documentation and Product Assembly	JP
11/18	Thanksgiving Break (Catch-Up Work/Troubleshoot if Necessary)	JH
		JP
11/25	Prepare Demo and Presentation : Speech/Pitch Write-Up	JH
	Prepare Demo and Presentation : Powerpoint	JP
12/2	Demo/Presentation & Prepare Final Paper	JH
		JP
12/9	Schematics/Presentation/Final Paper Finish & Checkout	JH
	Code/Presentation/Final Paper Finish & Checkout	JP

Ethical Considerations - IEEE Code of Ethics

1. *To accept responsibility in making decisions consistent with the safety, health, and welfare of the public, and to disclose promptly factors that might endanger the public or the environment.*

One ethical concern that arrives from the onset of our project is that it is essentially a "punishment device". Its sole intention is to cause harm to an animal, and while that sounds harsh, it is for the good of all parties involved. We ourselves will not be implementing the means of punishment, but current shocking devices have been approved by the CVM (Center for Veterinary Medicine, part of the FDA) as a humane alternative training measure. Certain shocking regulations are in place that ensure the safety of the pets involved. The device itself (with shocking or not) will be entirely safe for use with no infractions to the safety, health, or welfare of the general public or their associated pets.

2. *To avoid real or perceived conflicts of interest whenever possible, and to disclose them to affected parties when they do exist.*

There might be some conflicts of interest with certain animal rights activists that do not approve of this method of pet training. One of America's greatest rights is that to protest, by boycott or otherwise, and it is understood on our part if people choose not to purchase our product. Our product will be entirely safe and all safety issues, if any arise, will be addressed to the general public.

3. *To be honest and realistic in stating claims or estimates based on available data.*

All of our stated claims and estimates are based entirely out of factual component data As the product nears completion, estimates will be updated to exact statistical and experimental models. No false claims will be made.

4. *To reject bribery in all its forms.*

While no bribes have been offered as of yet, we swear to reject all briberies in all of their forms.

5. *To improve the understanding of technology; its appropriate application, and potential consequences.*

We are proud of the work we achieve as we strive for the best product we can create. Using other subsidiaries' products to achieve that goal requires a certain understanding of each borrowed component and we swear to fully justify the use of every piece of technology. Understanding each component is the first step in assembly and we will knowingly take the time to fully understand each.

6. *To maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations.*

We will make sure our technical knowledge not only applies, but is also up to the modern standard in all tasks undertaken. This applies to everything we do ourselves and for all other projects upon which our assistance is requested.

7. *To seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others.*

We will openly accept, acknowledge, and correct for all honest criticms. All contributions will be honored.

8. To treat fairly all persons regardless of such factors as race, religion, gender, disability, age, or national origin.
9. To avoid injuring others, their property, reputation, or employment by false or malicious action.
10. To assist colleagues and co-workers in their professional development and to support them in following this code of ethics.

We promise to follow these three codes without question as they are already embedded in our very personalities.

We also realize that there may be ethical dilemmas that extend outside of the reach of the IEEE Code of Ethics. Whenever they arise we will properly consult an appropriate authority, and with accordance to IEEE Ethics Code 7, we will accept honest criticism and advice leading to a solution.

Approval from the UIUC Institutional Review Board for animal testing is not yet confirmed, but should be shortly.

Selected Resources

http://en.wikipedia.org/wiki/Classical_conditioning
http://en.wikipedia.org/wiki/Shock_collar
http://www.jameco.com
http://www.sparkfun.com/
http://www.arduino.cc/
http://www.ieee.org/about/corporate/governance/p7-8.html