

The Traveling Geocache!

by [williamanos](#) on January 31, 2011

Table of Contents

The Traveling Geocache!	1
Intro: The Traveling Geocache!	2
Step 1: Private Property Rights are Upheld by Police with Guns	4
Step 2: Materials + Tools	5
Step 3: Cutting the Box	6
Step 4: Breadboard Arduino	6
Step 5: Using a Servo	8
Step 6: Using a GPS	8
Step 7: Using an LCD	9
Step 8: Using a Relay for Auto-turn-off	9
File Downloads	10
Step 9: Perfboard Arduino	11
Step 10: Wiring it all up!	11
File Downloads	16
Step 11: Secret Back Door Wires	17
Step 12: Locking Mechanism	18
Step 13: Programming and Software	18
File Downloads	19
Step 14: Load USB Stick	19
File Downloads	21
Step 15: Fill your Box with Goodies!	22
Step 16: Release into the Wild!	22
Step 17: Thoughts, Concerns, Considerations	23
Related Instructables	23
Comments	24

Intro: The Traveling Geocache!

This project and future projects can be found at my website <http://revoltlab.com/> where you can also find my blog.

UPDATE LOG:

NOW SUPER SIMPLE TO REPROGRAM!

Easy FTDI USB interface hardware explained in step 10

Easy FTDI USB interface programming explained in step 13

NEW code uploaded works perfectly with speaker output! see step 13

For those not familiar with geocaching, it is a wonderful treasure hunt/ adventure game for grownups and kids! Traditionally one finds a set of coordinates and cryptic clues on a website that hosts geocache locations. Using a hand held GPS device, one simply tromps off into the wilderness (or busy city intersection) and uses the power of observation and clue solving to find a hidden box usually with a prize inside! This is loads of fun and free! <http://www.geocaching.com/>

The Traveling Geocache still requires the user to get to a specific set of GPS coordinates but the hand held device is replaced by the Traveling Geocache itself. It is a locked box with an LCD display and one button. Upon pressing the button an intro screen welcomes the user to the Traveling Geocache and then displays the distance in kilometers from the target location. If the distance is larger than the programmed margin of error radius, the LCD informs you that access is denied. The box will not unlock until it is brought to the correct location! This is a great way to give a present to a friend or loved one. You can put tickets inside and set the magic location to the museum, aquarium, stadium, train station (DO NOT set a location that requires plane travel! Unopenable homemade electronic boxes are not welcome at airport security!). Any prize will do. Because the box is intended for a specific person, the prize can be far more special than in a traditional geocache.

Apart from the prize, the box also contains a log book where recipients can sign their names and write messages. There is also a usb drive with information on all of the electronics and software. This enables the recipient to reprogram the box, change the destination, customize the LCD messages, replace the prize, and re-gift the box to another friend, loved one, or trustworthy stranger. You can even leave a url somewhere inside the box where users can enter their names and destination locations enabling every user to track the life of the box!

This was my first time using GPS and LCD modules as well as my first time working with relays. Don't worry about the learning curve if you can already make a light blink on the arduino.

Thanks to sparkfun and instructables for holding the microcontroller contest. It really motivated me to document and post my first ever instructable! If you like it, please rate and vote for this project!

Please leave comments! =)



Image Notes

1. A synthesis of human, machine, and our natural physical universe, the Traveling Geocache rejects alienation of lifeforms by machines. Rather than imbue a machine with an exploitative social relation (taylorism, that light on the fry-o-lator that tells you to hurry up), one can create a machine that changes social relations positively each time it is used!
2. background illustration from adafruit's blog post



Image Notes

1. Standard servo has been upgraded to micro servo. The micro servo has a smaller footprint and only costs 5 bucks at Deal Extreme (<http://www.dealextreme.com/p/towerpro-sg90-9g-mini-servo-with-accessories-12859>)
2. GPS
3. 16X2 LCD
4. Locking mechanism comprised of eye hooks and a nail attaches to the eye hook on the lid of the box.
5. eye hook on the lid of the box
6. Momentary switch
7. 5 volt relay
8. 4 AA battery pack (used the missing two batteries to take this picture!)
9. servo cable
10. secret back door override cables!
11. Arduino clone w/ Atmega328
12. A book I keep of my favorite songs. A moleskine fits perfectly in the box and makes a good log book for recording each user. You can use a smaller log book if you like.
13. USB drive with arduino software, libraries, schematics. Basically a compact users' manual!
14. Can't forget a pen!
15. Space for a geocache treasure!
16. a shirt
17. warm cozy sock
18. MINI USB! for reprogramming the destination, greeting screen, and contact information! (see steps 10 and 13!)
19. speaker for victory tune upon arriving at destination! (see pics in step 10)



Image Notes

1. momentary switch
2. 16X2 LCD this one is white text on a blue background.



Image Notes

1. I cut up the edges of the stickers sparkfun sends out in their shipments to make labels for these sockets
2. yellow wires are the digital pin connections for the LCD
3. Servo and GPS
4. Power and programming

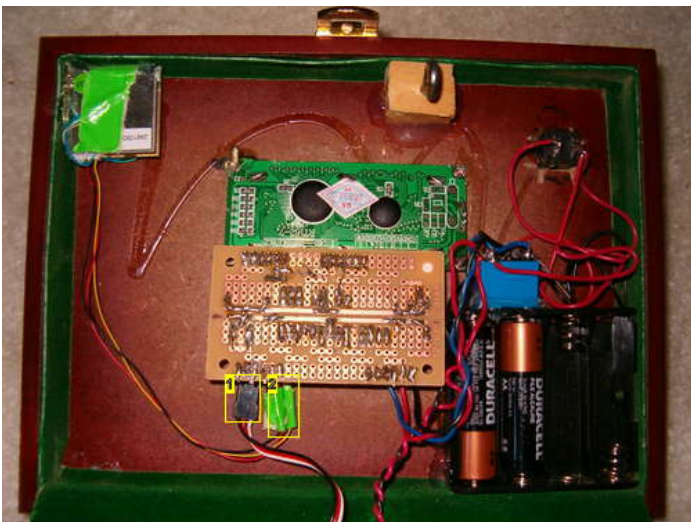
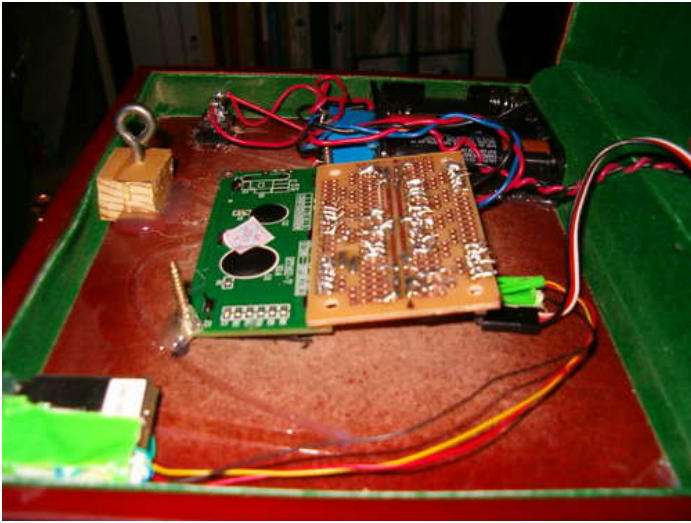
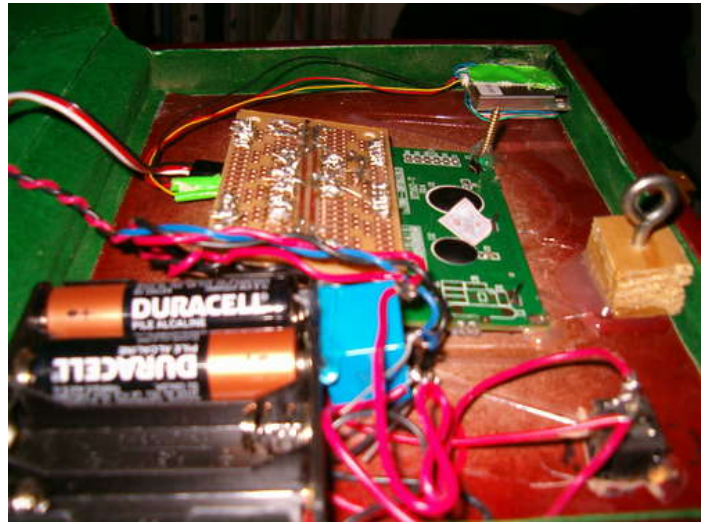


Image Notes

1. servo
2. GPS





Step 1: Private Property Rights are Upheld by Police with Guns

Despite our dreams of open source communism, capitalism is still a reality. With that in mind here is some info on intellectual property involved in this project.

This project was inspired by Mikal Hart's "Reverse Geocache Puzzle Box" He has been wonderful about providing information on his design. **Please note: "Reverse Geocache" and "Reverse Geocache Puzzle Box" are trademarked intellectual property of Mikal Hart.** If you think that is uncool, he DOES have all the wiring diagrams and software available for free as long as you are not profiting off of them. He did after all write several arduino libraries for the program.

http://www.youtube.com/watch?v=Lu7lysgaZf8&feature=player_embedded

The original reverse geocache:
<http://arduiniiana.org/projects/the-reverse-geo-cache-puzzle/>

Build instructions for the original Reverse Geocache:
<http://arduiniiana.org/projects/the-reverse-geo-cache-puzzle/building/>

Other Acknowledgements:

Thanks to lady ada for all her super helpful tutorials!

Thanks to sparkfun for keeping up the Free Day tradition. I got my very first atmega328 with my 10 free day dollars (YES, I actually made it onto their site on free day!)

Thank you as well to RickP on the arduino forums for help with the relay switch design.
<http://www.arduino.cc/cgi-bin/yabb2/YaBB.pl?num=1293755082>

<http://www.instructables.com/id/The-Traveling-Geocache/>



Step 2: Materials + Tools

Sparkfun Electronics is a good place to get most of what you need <http://www.sparkfun.com/>

List of Tools:

Soldering Iron
Wire cutters + strippers
Hot Glue gun
epoxy
dremel

List of Materials:

Cool looking box

Solder

Breadboard <http://www.sparkfun.com/products/9567>

Arduino <http://www.sparkfun.com/products/9950> (you need to get one with a through hole chip. The surface mount SMD version of the arduino UNO will not work)

atmega328 chip <http://www.sparkfun.com/products/9061> or <http://www.sparkfun.com/products/9217>

28 pin dip socket (for atmega328) <http://www.sparkfun.com/products/7942>

perf board <http://www.sparkfun.com/products/8812>

LED <http://www.sparkfun.com/products/9592>

1N4004 or any 1N400X diode (you can get these at radioshack)

Momentary switch

330Ohm resistor <http://www.sparkfun.com/products/8377>

10KOhm resistor <http://www.sparkfun.com/products/8374>

10k Ohm trim pot <http://www.sparkfun.com/products/9806>

22pf Capacitors (2) <http://www.sparkfun.com/products/8571>

16MHz crystal <http://www.sparkfun.com/products/536>

22 gauge solid core wire (multiple colors help a lot) <http://www.sparkfun.com/products/8022>

standard servo <http://www.sparkfun.com/products/9065>

gps module <http://www.sparkfun.com/categories/4>

LCD (I used a 16X2) <http://www.sparkfun.com/products/790>

5 volt relay

4 AA battery holder

4 AA batteries

female headers <http://www.sparkfun.com/products/115>

male headers <http://www.sparkfun.com/products/116>

<http://www.instructables.com/id/The-Traveling-Geocache/>

scrap wood
large nail or metal bar
eye hooks (3)
usb stick
notebook
pen

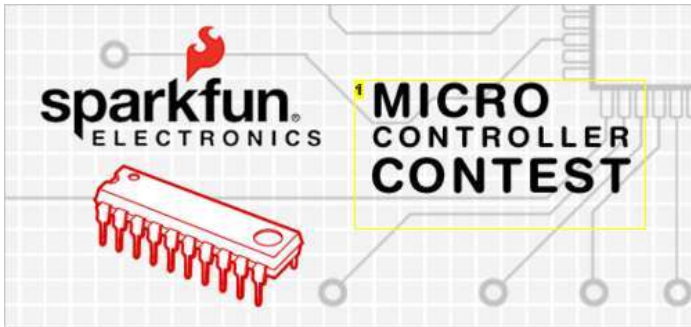


Image Notes

1. This is what motivated me to do this instructable! If you have a project you should totally enter this contest!

Step 3: Cutting the Box

This picture is of the first box I attempted to cut. This is the correct pattern but I cut the LCD hole too large! Always cut a smaller hole than you think you will need and work from there. I found it easiest to use a dremel but use whatever you like.

Now is also a good time to set four screws into the bottom of your box to act as legs. Later on, we will use two of these for a secret back door entrance into the box!



Image Notes

1. LCD opening
2. switch socket

Step 4: Breadboard Arduino

Here is all the info from the arduino website : <http://arduino.cc/en/Tutorial/ArduinoToBreadboard>

If you bought an Atmega328 with the bootloader already installed, simply follow the second half of this step. If you have a blank chip, please follow the entire step.

Burning the Bootloader

If you have a new ATmega328 (or ATmega168), you'll need to burn the bootloader onto it. You can do this using an Arduino board as an in-system program (ISP). If the microcontroller already has the bootloader on it (e.g. because you took it out of an Arduino board or ordered an already-bootloaded ATmega), you can skip this section.

To burn the bootloader, follow these steps:

1. Upload the ArduinoISP sketch onto your Arduino board. (You'll need to select the board and serial port from the Tools menu that correspond to your board.)
2. Wire up the Arduino board and microcontroller as shown in the diagram to the right.
3. Select "Arduino Duemilanove or Nano w/ ATmega328" from the Tools > Board menu. (Or "ATmega328 on a breadboard (8 MHz internal clock)" if using the minimal configuration described below.)
4. Run Tools > Burn Bootloader > w/ Arduino as ISP.

You should only need to burn the bootloader once. After you've done so, you can remove the jumper wires connected to pins 10, 11, 12, and 13 of the Arduino board."

<http://www.instructables.com/id/The-Traveling-Geocache/>

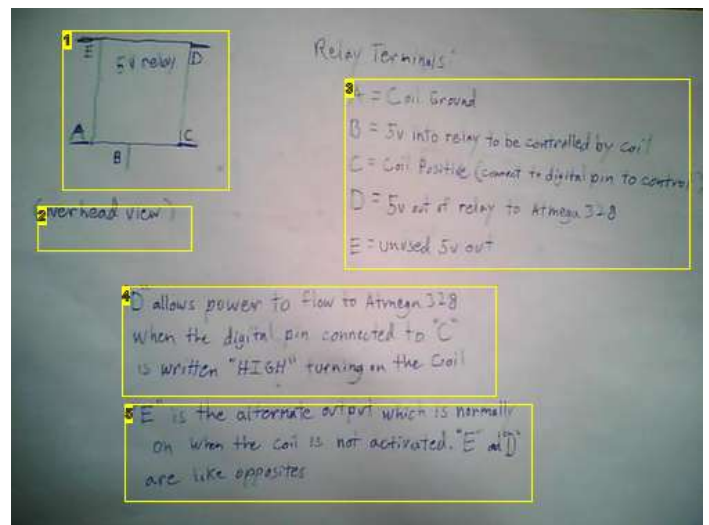
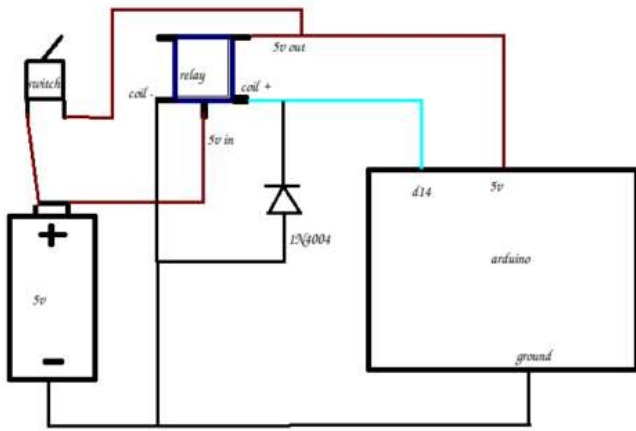


Image Notes

1. Because I could not find the data sheet for the relay online, I have arbitrarily assigned these letters to the terminals for the sake of explanation.
2. the relay is labeled as if you were looking down on it set in a breadboard.
3. A = Coil ground B = 5 volts into relay to be controlled by the coil C = Coil positive (connect to digital pin for controlling relay) D = 5 volt out of relay to power Atmega 328 E = unused 5v out
4. D allows power to flow to Atmega 328 when the digital pin connected to "C" is written HIGH turning on the coil.
5. "E" is the alternate output which is normally on when the coil is not activated. "E" and "D" are like opposites.

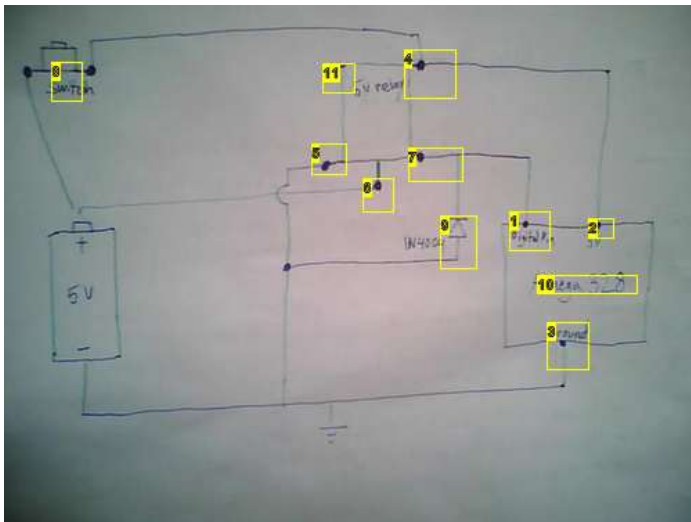


Image Notes

1. Digital Pin
2. 5 Volt to power the arduino
3. Ground
4. relay pin out latches power to the arduino. The switch supplies initial power while the arduino starts up and activates the coil, fixing it in an 'on' state until the program on the arduino decides to cut the power and shut itself down.
5. ground
6. Relay power in. The power connected here flows through the relay to the arduino if the coil is activated.
7. positive end of the coil. power for the coil comes from a digital pin on the arduino. When the pin is written HIGH the coil turns on and power flows from the battery through the relay and to the arduino allowing you to let go of the button.
8. switch or button
9. 1N4004 blocking diode to protect circuit when relay coil powers off
10. Arduino
11. Unused relay out. This is the opposite of the connection directly to the right. When the coil is off this contact is on. When the coil is on, this contact is off.

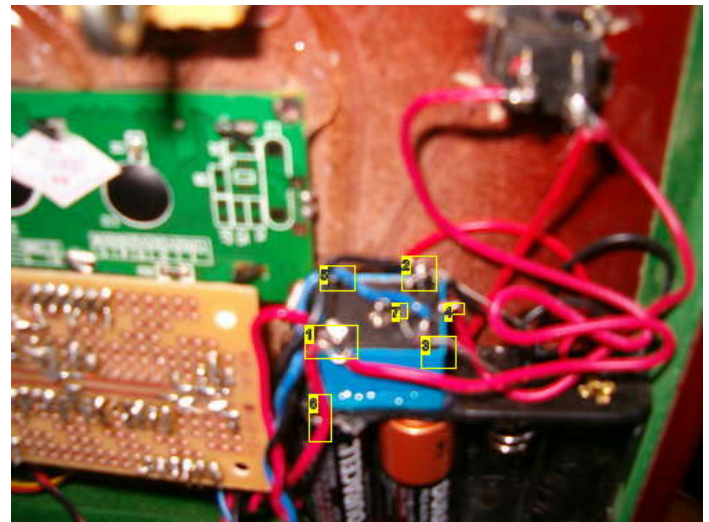


Image Notes

1. 5 volts to board
2. ground. Secret override ground also goes here. Situating the relay with the leads exposed is a great way to wire in new components!
3. coil 5v connected to digital pin
4. 5 volts into relay controlled by coil
5. unused relay out
6. secret back door power wire
7. 1N4004 diode (positive to terminal "A", negative to terminal "C")

File Downloads



relaycircuit.pdf (12 KB)

[NOTE: When saving, if you see .tmp as the file ext, rename it to 'relaycircuit.pdf']

Step 9: Perfboard Arduino

I found this instructable by jmsaavedra helpful in making the leap from breadboard to perf board: <http://www.instructables.com/id/Perfboard-Hackduino-Arduino-compatible-circuit/>

Basically your goal is to recreate the arduino breadboard circuit on this perf board making it smaller, stronger, and more permanent.

All you need to put on the board is the same circuit you had on the breadboard with the addition of an LED on pin 13. Everything else in the above example you can ignore.

Keep your connections farther from the board edge than jmsaavedra because we need those outer spaces for female header connections and the LCD. Try to keep it tight!

The power regulator used in the above circuit isn't really necessary for our project because the operational voltage of the atmega328 is up to 5.5 volts and we are only going to be slightly over that at the very beginning of the Traveling Geocache's life. That being said, if you are worried about damaging the circuit you will need a regulator with less than the 2 volt drop of the 7805 regulator used in jmsaavedra's hackduino. The 7805 would make the maximum voltage available to the circuit 4 volts. This is not enough to optimally power the LCD and the servo.

The perfboard is going to be the central hub of your box. The servo, GPS, LCD, batteries, relay, and switch will all be connecting to this board in one way or another. Before getting into any of those connections however make sure you can run the simple blink LED program demonstrated in jmsaavedra's instructable. This will let you know if your board is working to begin with before complicating any troubleshooting process with more wires.

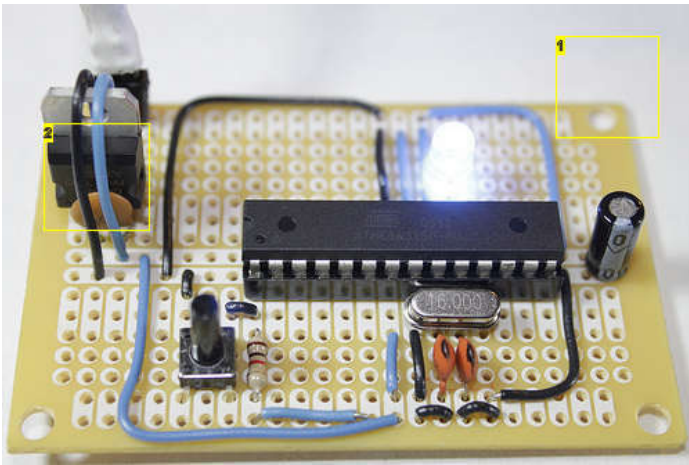


Image Notes

1. This photo is from jmsaavedra's instructable in the link
2. use a low voltage drop regulator if you want one

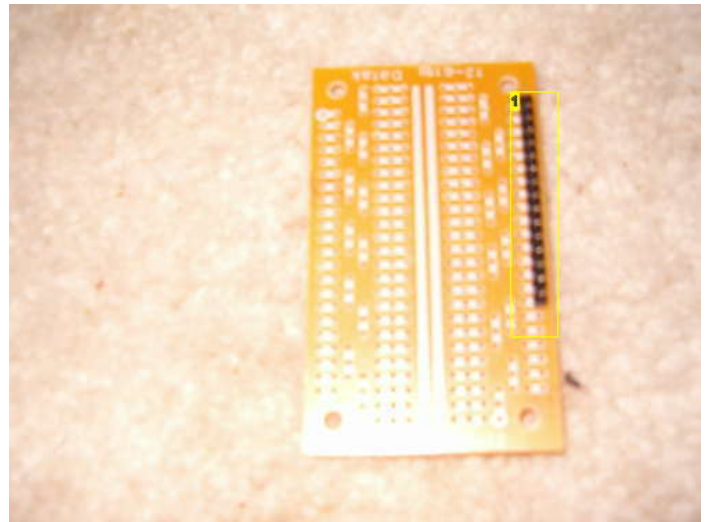


Image Notes

1. this header is placed where the LCD will connect to the board. Keep the bare bones arduino circuit away from the edges as much as possible to leave room for the LCD, gps, servo, power, ground, reset, and relay latch wires later.

Step 10: Wiring it all up!

Follow the schematics and photos. If the schematic images are too blurry, they are also available in pdf form. Make sure to read the image notes though!

The important thing to remember is that the

- power,
 - ground,
 - reset,
 - tx,
 - rx,
 - analogue 0,
 - digital 2,
 - gps power,
 - gps ground,
 - digital 9,
 - servo power,
 - and servo ground
- are all connected to the board via stackable female headers. SEE THE PHOTOS!

FTDI USB MODULE

I used a sparkfun 5v FTDI breakout board for easy usb interface. <http://www.sparkfun.com/products/9716>

Essentially this module replaces the arduino main board between the breadboard and the computer in step 4. The connections are very simple. 5 volts goes to power (connected on relay terminal), Ground goes to ground (also on relay terminal), rx goes to tx, tx goes to rx, and the DTR pin goes to reset on the perfboard arduino. **NOTE** You will need a small capacitor (i used a .47uF capacitor) between the DTR pin and the reset terminal on the perfboard.

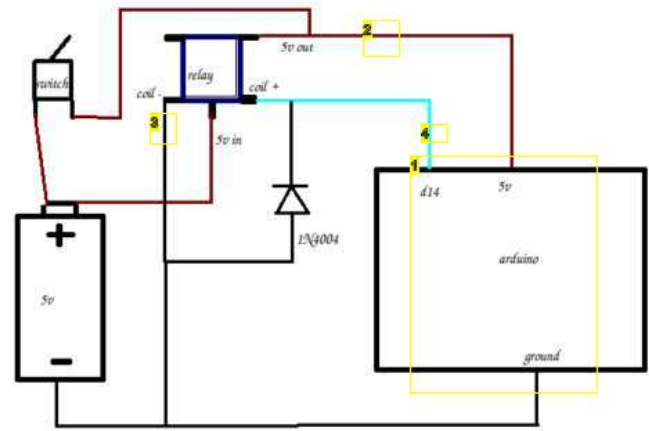
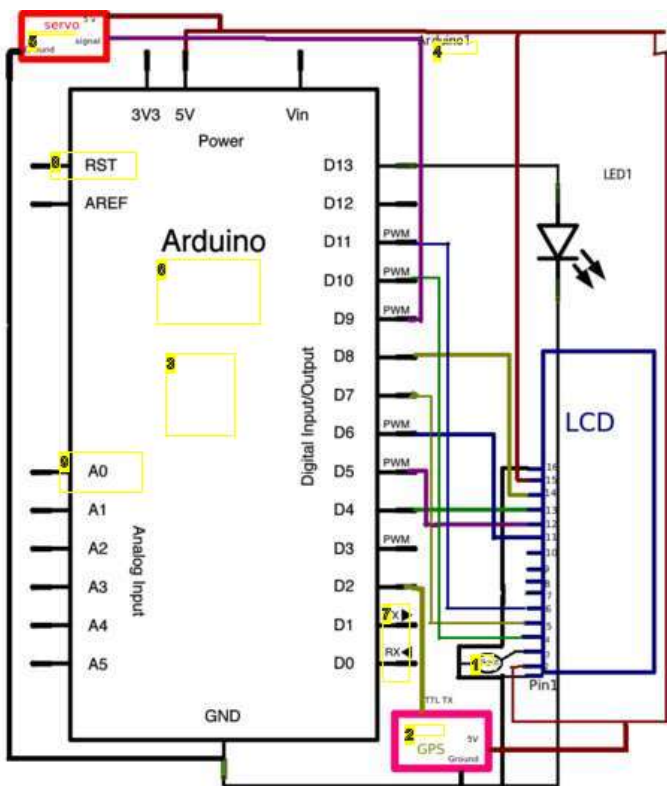


Image Notes

1. Take note that all three wires going to the arduino in this electrical diagram will attach to the board via the stackable female headers shown in the photos.
2. this is where you will eventually connect the power for secret access.
3. This is where you will eventually connect the ground for secret access.
4. d14 or digital pin fourteen is labeled as "analogue 0" in the arduino diagram and on regular arduinos.

Image Notes

1. 10K Ohm potentiometer for adjusting LCD contrast
2. GPS REMEMBER: The GPS attaches to the arduino board via the stackable headers in the photos. This is a purely electrical diagram.
3. Check the PDF for a closer look!
4. Ignore this text. It means nothing.
5. Servo REMEMBER: The servo attaches to the arduino board via the stackable headers in the photos. This is a purely electrical diagram.
6. Note that this circuit is just for the modules. It does not include the break out wires from RX, TX, RESET, 5v, Ground, or Relay digital pin. It also does not contain the 10k resistor, LED resistor, 16MHz crystal, or two 22pf capacitors that should already be on your board.
7. Don't forget to connect these to the female header pins! (as shown in the photos)
8. Don't forget to connect this to the female header pins! (as shown in the photos)
9. Don't forget to connect this to a female header pin! (as shown in the photos)

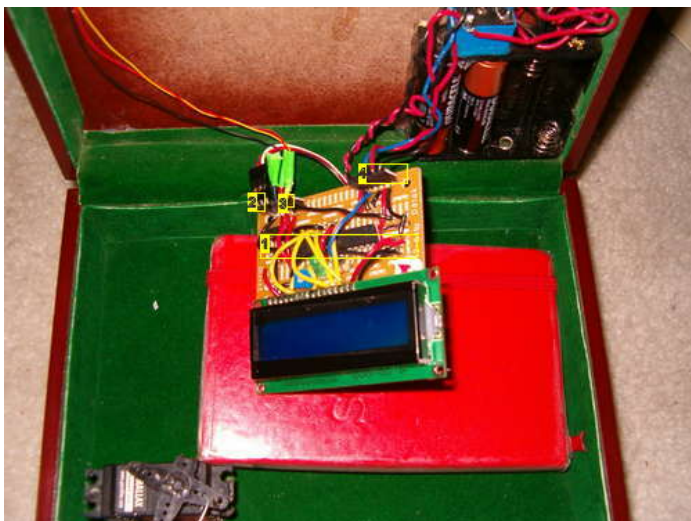


Image Notes

1. The beautiful beautiful board!
2. servo connection with a three prong male header bent at 90 degrees to fit

- into half of the 6 pin stackable header.
- 3. gps wires stuck into the other half of the 6 pin header
- 4. header includes tx, rx, power, ground, reset, and digital pin 14 for controlling the electromagnetic coil in the relay.



Image Notes

- 1. to ground
- 2. to power
- 3. to reset
- 4. to digital pin 14 (also known as analogue 0)
- 5. to rx
- 6. to tx
- 7. to digital pin two (gps data)
- 8. to 5 volts (GPS power)
- 9. to ground (GPS ground)
- 10. to digital pin 9 (servo signal)
- 11. to 5 volts (servo power)
- 12. to ground (servo ground)



Image Notes

- 1. I cut up the edges of the stickers sparkfun sends out in their shipments to make labels for these sockets
- 2. yellow wires are the digital pin connections for the LCD
- 3. Servo and GPS
- 4. Power and programming

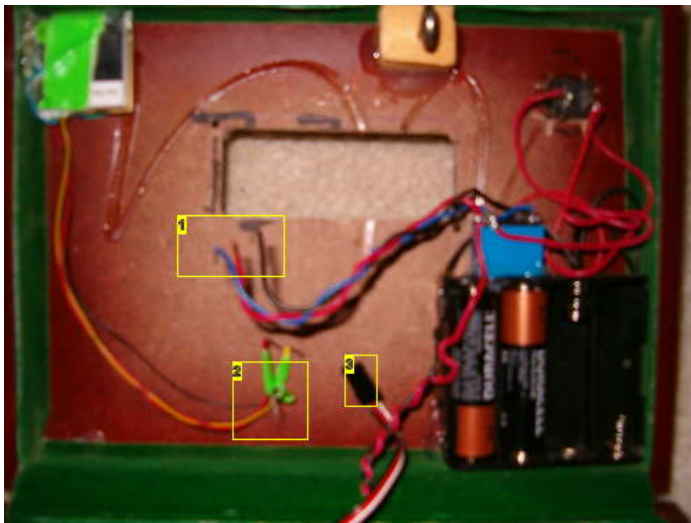


Image Notes

- 1. three wires coming from the relay to the board control power
- 2. 3 wires from the gps
- 3. servo wires

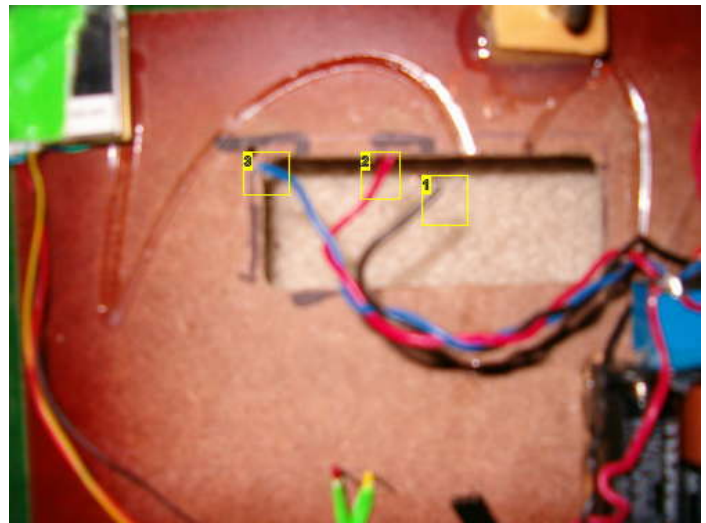


Image Notes

- 1. ground from relay to board
- 2. power from relay to board
- 3. coil wire from relay to board (digital pin 14 aka analogue 0)

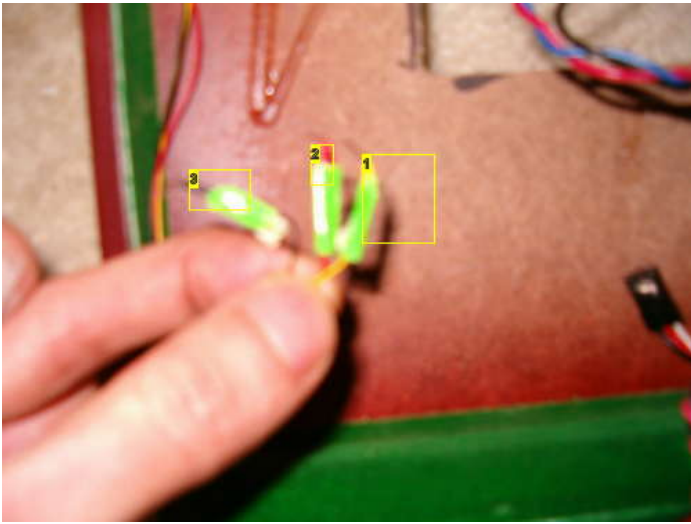
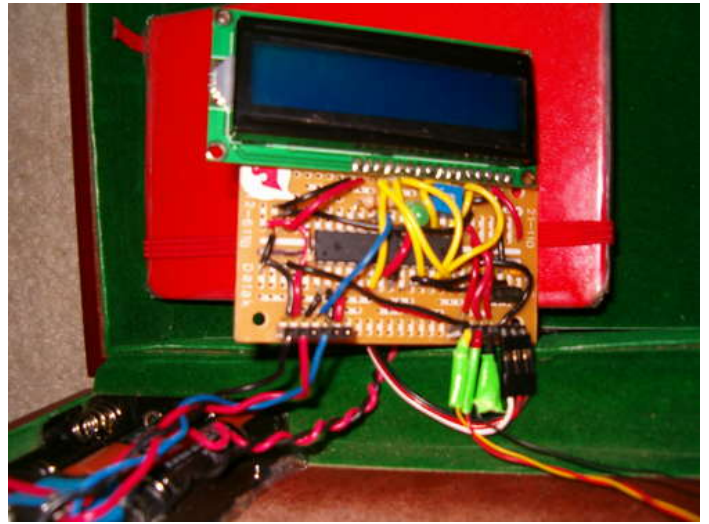


Image Notes

1. gps data goes to digital pin 2
2. gps power
3. gps ground



Im

1. :
2. :
3. :
4. :
5. :
- for
6. :

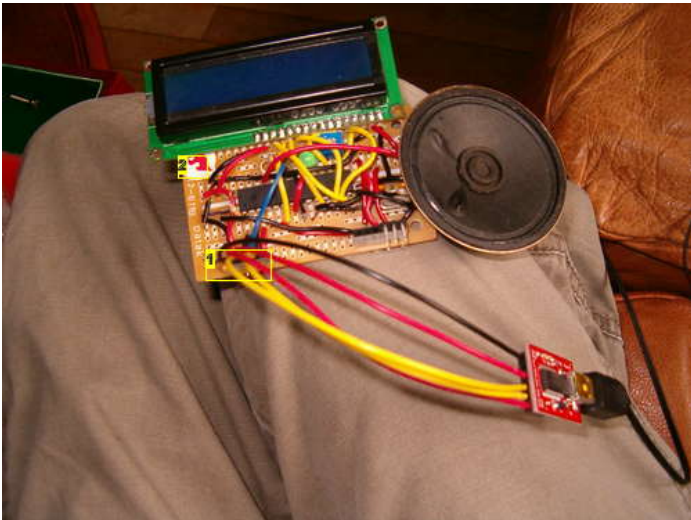
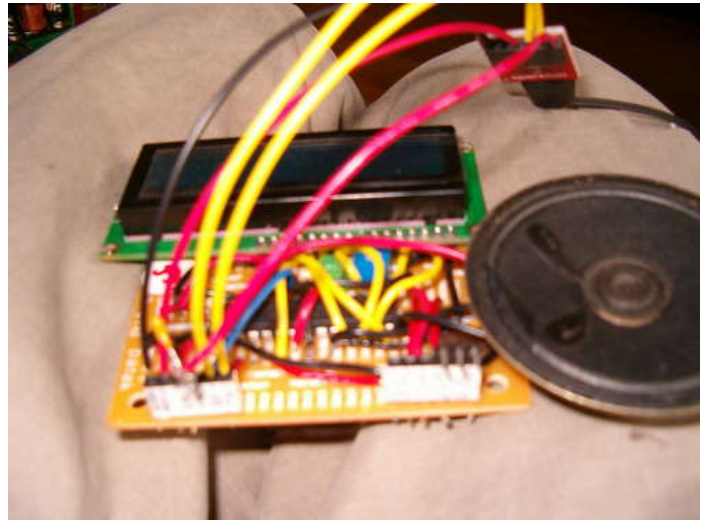


Image Notes

1. for testing purposes power and ground were connected on the perf board. Upon installation these power and ground wires will be attached at the relay terminals.
2. spark of fun



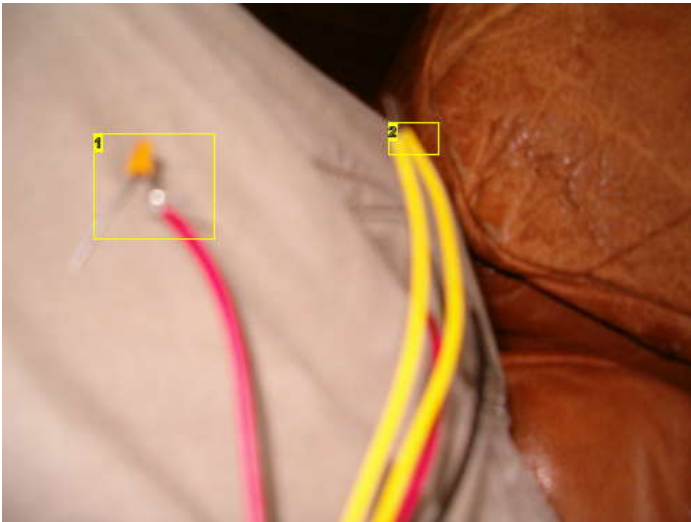


Image Notes

1. .47uF capacitor coming from DTR pin leading to RESET on the perfboard.
2. tx and rx

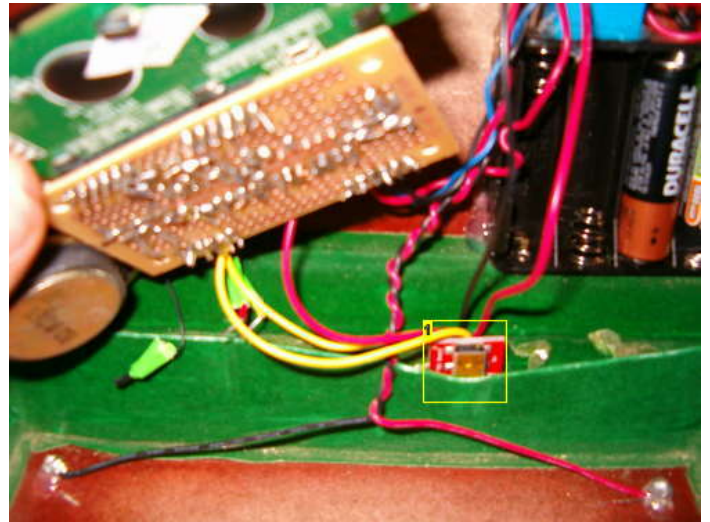


Image Notes

1. Wedged the module in the back here. I found a place where the wires sat comfortably and then drowned the sides of the board in hot glue and pressed the felt backing against the wall securing it in a tight position.

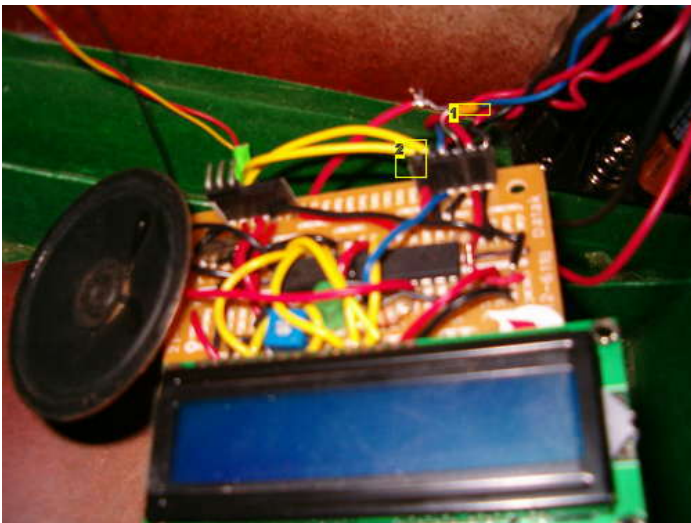


Image Notes

1. .47uF capacitor
2. remember rx to tx and tx to rx

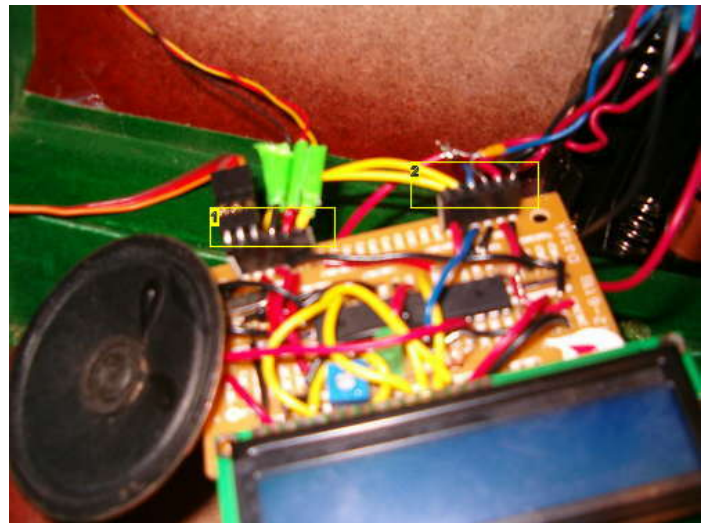


Image Notes

1. GPS and Servo
 2. rx, tx, DTR from FTDI board AND power, ground, relay coil wire from relay
- NOTE: the power and ground from the FTDI board is attached at the relay!

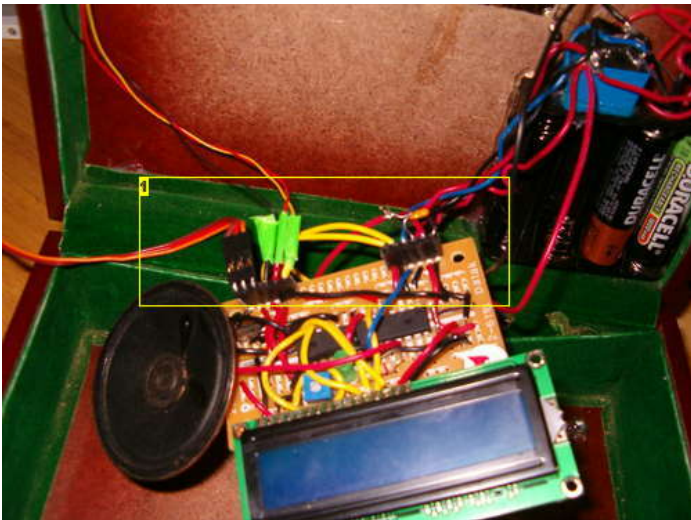


Image Notes

<http://www.instructables.com/id/The-Traveling-Geocache/>

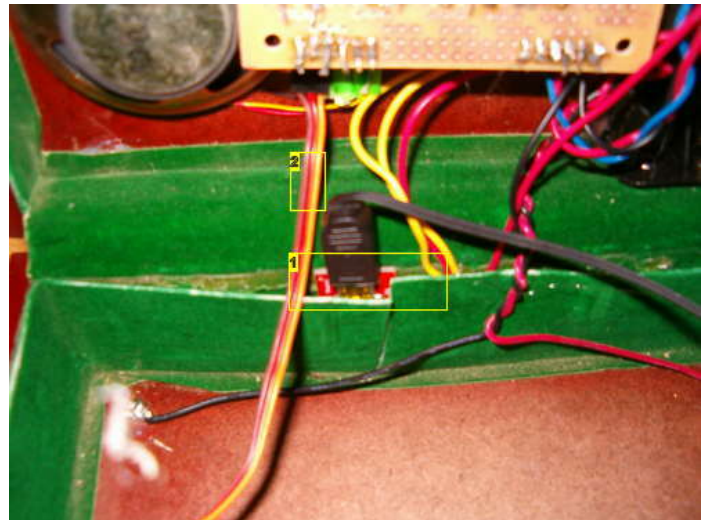


Image Notes

1. fully connected After testing the box is working, apply a thin layer of hot glue over the header pins to secure wires in place.

1. mini usb plugged in for programming! No need to ever take your board and screen out again!

2. don't worry about the wires changing colors in the photos. I switched the original servo with a smaller one.

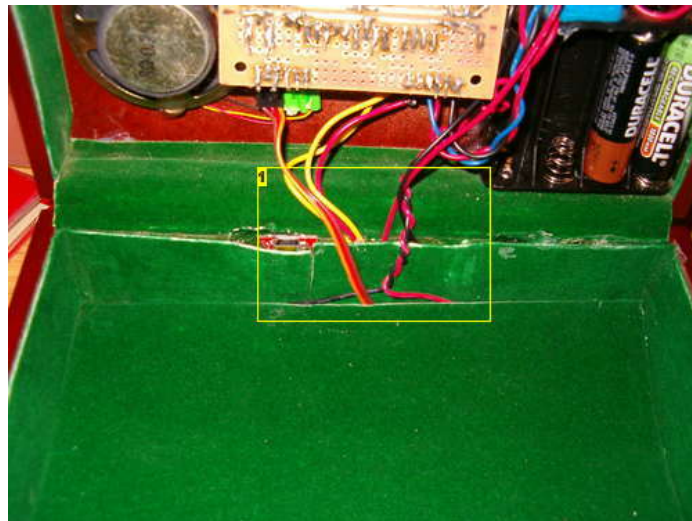
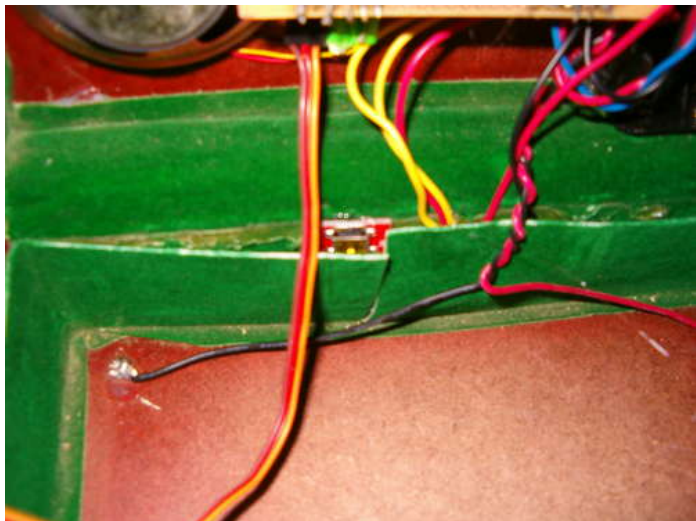


Image Notes

1. I am considering a cloth mask to cover these wires and all the electronics. I would want to do it with a zipper so the lid could be accessible.

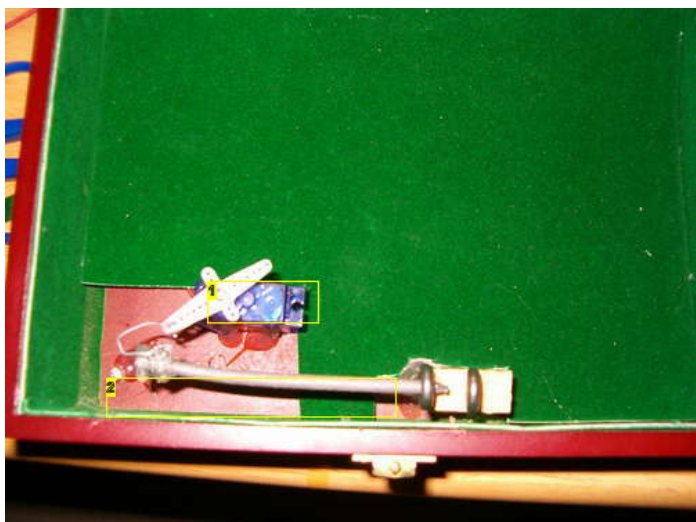


Image Notes

1. Five dollar micro servo from deal extreme!
2. space under the nail can be used for storing the usb to mini usb cable!

File Downloads



modueshookup.pdf (67 KB)

[NOTE: When saving, if you see .tmp as the file ext, rename it to 'modueshookup.pdf']



relaycircuit.pdf (12 KB)

[NOTE: When saving, if you see .tmp as the file ext, rename it to 'relaycircuit.pdf']

Step 11: Secret Back Door Wires

This super secret wiring will allow you to open the box even when it is not at the correct location. The arduino program has a function in it which is activated only when the relay switch power is being bypassed. When the program normally powers off (and says powering off) the secret wire terminals will keep the whole system running. After a few minutes the servo opens the box for 10 seconds before sealing the box back up.

If you haven't already, now is a good time to add four metal legs to the box. I used some screws that were lying around my basement. You can use any two of these for the secret wires, but the back two were closest to the circuit so I chose those ones. Simply solder a wire to each screw. Assign one to be red and one black so you know what is going where. Take the red wire and solder it to the relay pin which carries 5 volts from the relay to the arduino. Take the black wire and solder it to the ground connection on the relay. Cover the screws in hot glue and you have made your back door circuit! You can test it by applying 4 AA batteries to the screw on the bottom of the box. Be sure you remember which is ground and which is power! If it is wired correctly the circuit should turn on and stay on after the "powering down" message is displayed

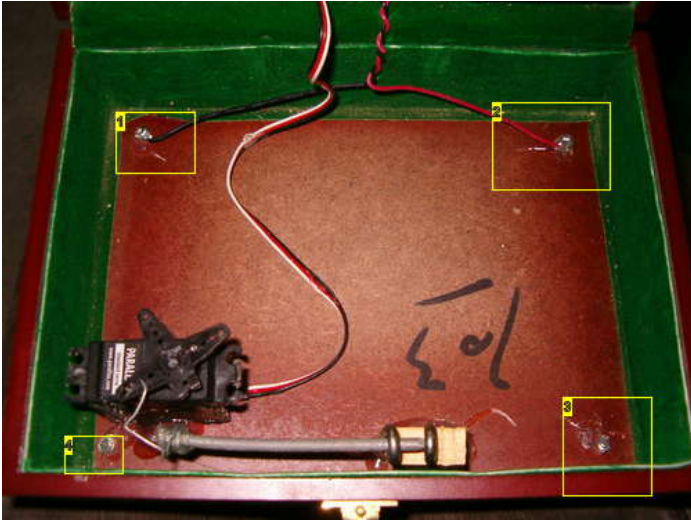


Image Notes

1. Ground for secret override power supply
2. 5 volts for secret override power supply
3. front legs so the back ones aren't so suspicious
4. front legs so the back ones aren't so suspicious



Image Notes

1. secret override terminals disguised as legs!

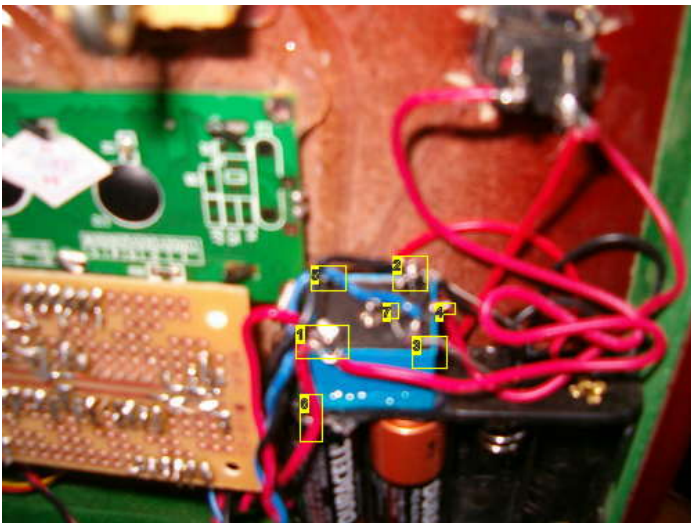


Image Notes

1. 5 volts to board
2. ground. Secret override ground also goes here. Situating the relay with the leads exposed is a great way to wire in new components!
3. coil 5v connected to digital pin
4. 5 volts into relay controlled by coil
5. unused relay out
6. secret back door power wire
7. 1N4004 diode (positive to terminal "A", negative to terminal "C")

Step 12: Locking Mechanism

The locking mechanism is pretty straightforward. A servo is attached to a nail with a paper clip. One end of the paper clip is bent and inserted into a hole on the servo horn. This allows the paper clip to pivot. The other end of the paper clip is hot glued to the head of the nail.

If your box has thick sides you can screw the hooks right into the box. Otherwise a scrap of wood epoxied to the box works great for setting the hooks!

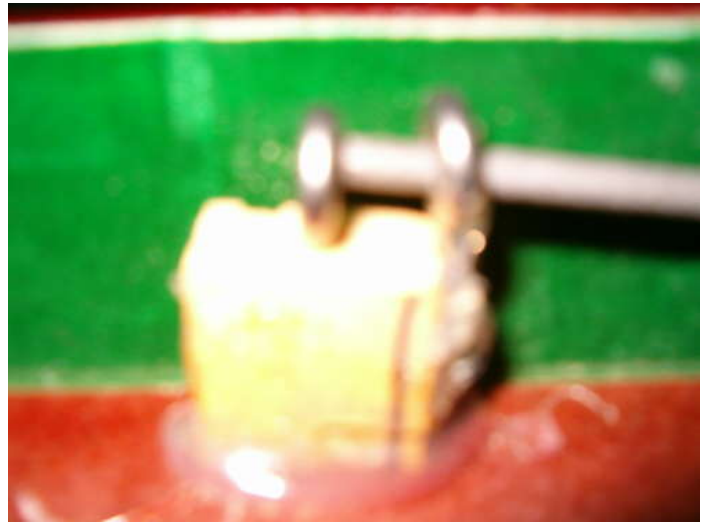
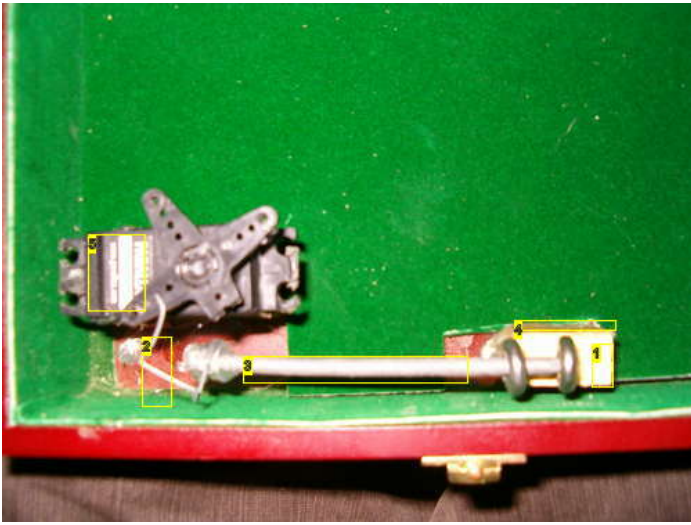


Image Notes

1. Lid hook rests here when closed.
2. paperclip
3. nail
4. random block of wood
5. SERVO!

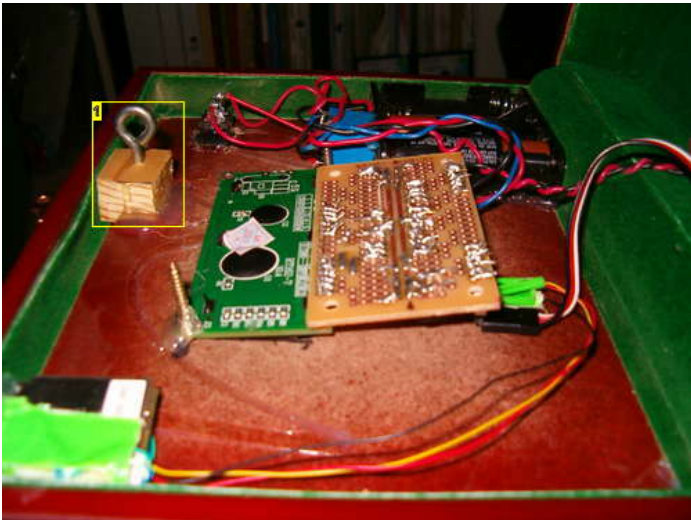


Image Notes

1. Eye hook on the lid is positioned directly next to the lower hook when lid is closed

Step 13: Programming and Software

Software

The software is based on Mikal Hart's "Reverse Geocache" code from Make Vol 25 <http://www.make-digital.com/make/vol25?pg=146#pg146>

Everything you will need to customize is commented in the code. Remember the LCD is 16 characters across so if you want to center text on the screen, add some spaces before the words.

Things you may want to change with each use:

Destination (get latitude and longitude by finding the location on google maps!) = **line 46**

Name of recipient = **line 105**

Distance from target necessary to open box (called RADIUS in the code) = **line 48**

"Welcome to BLANK" message upon opening = **line 158**

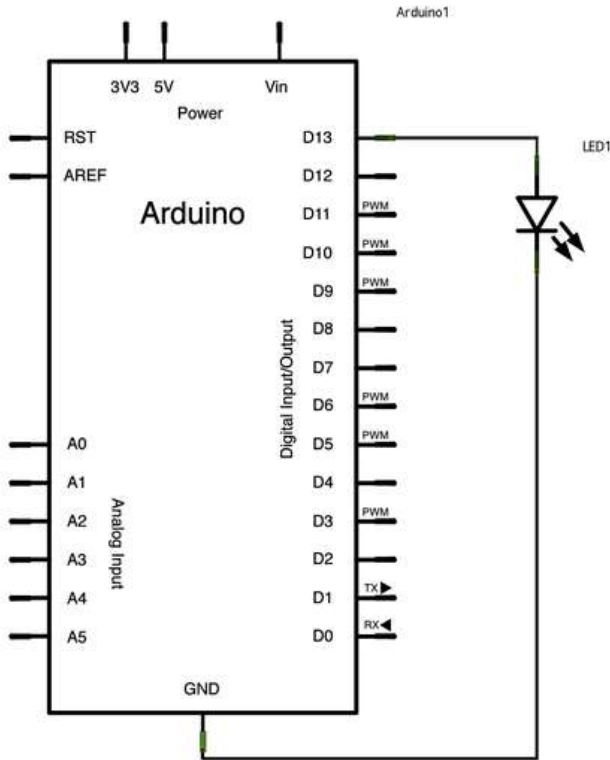
<http://www.instructables.com/id/The-Traveling-Geocache/>



Atmega168 Pin Mapping

Arduino function	Atmega168 Pin	Atmega168 Pin	Arduino function	
reset	(PCINT14/RESET) PC6	1	PC5 (ADC5/SCL/PCINT13)	analog input 5
digital pin 0 (RX)	(PCINT16/RXD) PD0	2	PC4 (ADC4/SDA/PCINT12)	analog input 4
digital pin 1 (TX)	(PCINT17/TXD) PD1	3	PC3 (ADC3/PCINT11)	analog input 3
digital pin 2	(PCINT18/INT0) PD2	4	PC2 (ADC2/PCINT10)	analog input 2
digital pin 3 (PWM)	(PCINT19/OC2B/INT1) PD3	5	PC1 (ADC1/PCINT9)	analog input 1
digital pin 4	(PCINT20/XCK/T0) PD4	6	PC0 (ADC0/PCINT8)	analog input 0
VCC	VCC	7	GND	GND
GND	GND	8	AREF	analog reference
crystal	(PCINT6/XTAL1/TOSC1) PB6	9	AVCC	VCC
crystal	(PCINT7/XTAL2/TOSC2) PB7	10	PB5 (SCK/PCINT5)	digital pin 13
digital pin 5 (PWM)	(PCINT21/OC0B/T1) PD5	11	PB4 (MISO/PCINT4)	digital pin 12
digital pin 6 (PWM)	(PCINT22/OC0A/AIN0) PD6	12	PB3 (MOSI/OC2A/PCINT3)	digital pin 11 (PWM)
digital pin 7	(PCINT23/AIN1) PD7	13	PB2 (SS/OC1B/PCINT2)	digital pin 10 (PWM)
digital pin 8	(PCINT0/CLKO/ICP1) PB0	14	PB1 (OC1A/PCINT1)	digital pin 9 (PWM)

Digital Pins 11, 12 & 13 are used by the ICSP header for MISO, MOSI, SCK connections (Atmega168 pins 17, 18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.



File Downloads



arduino-0022.tar.gz (3 MB)

[NOTE: When saving, if you see .tmp as the file ext, rename it to 'arduino-0022.tar.gz']



TravelingGeocache.pde (6 KB)

[NOTE: When saving, if you see .tmp as the file ext, rename it to 'TravelingGeocache.pde']



modueshookup.pdf (67 KB)

[NOTE: When saving, if you see .tmp as the file ext, rename it to 'modueshookup.pdf']



relaycircuit.pdf (12 KB)

[NOTE: When saving, if you see .tmp as the file ext, rename it to 'relaycircuit.pdf']



The-Traveling-Geocache.pdf (1 MB)

[NOTE: When saving, if you see .tmp as the file ext, rename it to 'The-Traveling-Geocache.pdf']

Step 15: Fill your Box with Goodies!

In addition to the log book, pen, and usb there is space for special geocache treasure! Keep in mind who you are giving the box to.

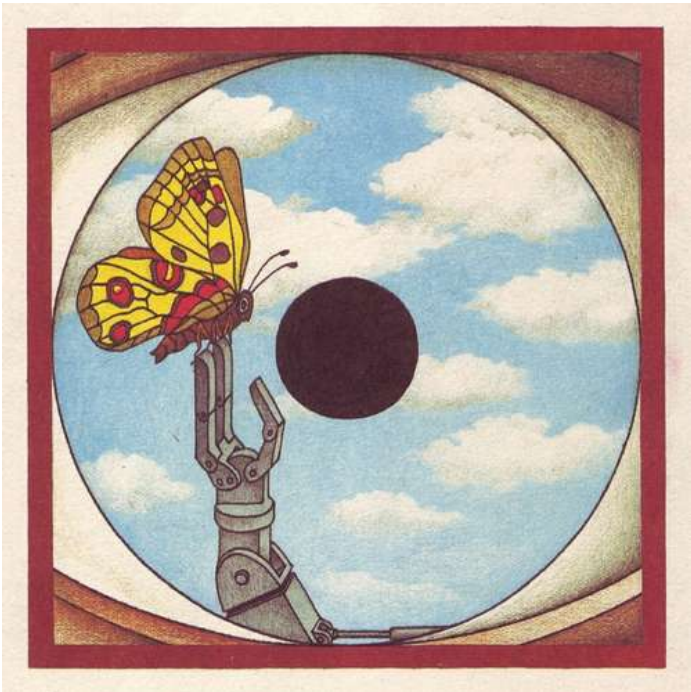
Wedding rings, keys to a new car, encrypted wikileaks files, cards, tickets, gold, consumables, arduinos, glow in the dark glass jellyfish, go crazy!

This part is all up to you.



Step 16: Release into the Wild!

Do a test run to make sure everything is functioning properly. Make sure you can install the arduino software with the usb and that the code runs fine before releasing your Traveling Geocache into the big beautiful world! Also, install NEW batteries with each gift!



Step 17: Thoughts, Concerns, Considerations

Final Thoughts:

The Traveling Geocache is not the cheapest of gifts. It is important to choose a good person to give it to. They should be trusted to actually find the location, not dismantle the box (steal components etc), and keep the box in circulation by giving it away. It is also a bonus if the recipient is interested in programming or electronics, but this is not necessary.

For me, the really exciting part of this project is being able to track where the box goes. To be sure this is possible you may want to write the url of your tracking site on the first page of your log book. As noted in the code, the url should also display on the LCD when the maximum attempt limit has been reached. Your website could be open to the public or require a password kept somewhere in the code or box. It could have space for media (pictures and video of where the box has been), user blog entries, and emergency instructions for opening the box after using up all 50 attempts!

I like the contrast of the log book and website. Something about using electronics to access a physical book makes opening the box more fun.

REMINDER: **NO PLANES!**

I hope you enjoyed this instructable! If you have any questions or comments be sure to put them in the comments section on the first page!

Happy Caching!

Will



Related Instructables



The Great Guide Of Geocaching.
by
james.mcglashan



Steampunk Geocaching by
r10n



How to Geocache by
PineapplebobTheC



How to make a geocache by
matthewja




Geocache Without A GPS
by Derelict




Geocache-A-Forest DIY Kit
by rrarunan


Comments


12 comments [Add Comment](#)

 **goblindust** says: Feb 6, 2011. 1:30 PM [REPLY](#)
Wow! What a great cache idea! Would love to have a look at the inter working some day so hope it heads my way. When and where will it be released? Happy caching!


 **williamanos** says: Feb 6, 2011. 2:59 PM [REPLY](#)
I am still waiting on the usb to serial FTDI break out board from sparkfun. That will allow simple usb interface making programming a purely software oriented endeavor. Once that is installed I must send it on a test run with some friends to see how it performs without my help. After that its just a matter of replacing hot glue with epoxy and making sure every step needed to operate the box is documented in pictures and text or even video on the usb drive. If I place in the contest I will put the money to making another one!


Thanks for the comment! Don't forget to rate the instructable and share it with anyone you think may be interested. =)

 **goblindust** says: Feb 6, 2011. 3:19 PM [REPLY](#)
I look forward to seeing the finished product. I think you have come up with a super original idea and very different kind of cache. Really like these "different" kinds. Hope when it is done it gets out Seattle way.


 **williamanos** says: Feb 6, 2011. 3:38 PM [REPLY](#)
It would be great if it made it across the continent (I'm in MA) without using planes or breaking the documentation logging (or the box!). If the first one goes well I will make the second one more sturdy (maybe an ammo box or something?). I'm trying to balance a non threatening appearance with durability.

Thanks for the rating!


 **goblindust** says: Feb 6, 2011. 4:12 PM [REPLY](#)
Check out the metal, first aid boxes as these are pretty tough, hinge on one side and come in different sizes. Only problem I see with them for your use is that you will have do a bit more work on the insulation to keep things from shorting out on the metal box. Durable and look safe.


 **williamanos** says: Feb 6, 2011. 4:17 PM [REPLY](#)
That's a good idea! Insulation isn't so hard with a layer of hot glue on the bottom of the board and over other exposed connections! Medical boxes are a good avenue I think. I could always get a pelican case but there is so much aesthetic potential in re-purposed boxes.


Thanks for the advice!


 **goblindust** says: Feb 6, 2011. 5:56 PM [REPLY](#)
Goodwill, Salvation Army or any second hand shop are great for finding such things as metal containers. If I find any links for cheap ones I will send them your way.
Keep up the great work on these type of ideas!

 **Nutrition Man** says: Feb 3, 2011. 12:51 PM [REPLY](#)
So, it like walks around?
I don't get it!

 **williamanos** says: Feb 3, 2011. 12:54 PM [REPLY](#)
No it doesn't walk around. You give it to someone who travels to the destination. When they open the box and take their prize they reprogram it and give it to another person and so on.

 **Nutrition Man** says: Feb 4, 2011. 12:31 PM [REPLY](#)
Nice, but what if someone picks it up and doesn't know about arduinos?
(Like me)

 **williamanos** says: Feb 4, 2011. 12:42 PM [REPLY](#)
All you need to do is change the gps coordinates in the code where it is clearly marked. There are a bunch of things you can change as listed in step 13!

 **williamanos** says: Feb 4, 2011. 12:39 PM [REPLY](#)
That is why the usb drive inside has the full pdf of this instructable and all the arduino software as well as a beginning arduino manual! =)

I'm still waiting on parts from sparkfun to get the usb to serial interface working to make it super easy to reprogram.
