



Secret Knock Detecting Door Lock
by [Grathio](#) on October 12, 2009

Table of Contents

Author: Grathio 2

License: Attribution Non-commercial Share Alike (by-nc-sa) 2

Intro: Secret Knock Detecting Door Lock 2

step 1: Tools, Supplies, And Skills 3

 (IF this all looks too challenging, you might consider signing up to the kit mailing list which, when available, will be much easier and a lot more simple.) Time
 : 3

 Skills : 3

 Tools: 3

 Materials : 3

 Electronics : 3

 Case: 4

step 2: Program The Arduino 4

 File Downloads 5

step 3: Lay Out And Test The Circuit 5

step 4: Prepare The Case 7

step 5: Make The Lock Turning Clamp 8

step 6: Make The Knock Detector Spring 9

step 7: Soldering The Circuits 10

step 8: Assembling The Case 15

step 9: Mounting, Testing, and Use 18

step 10: Epilog: Changes And Improvements 18

 Did you build this? 18

 Masters of Secret Knocks: 18

Related Instructables 19

Advertisements 19

Comments 19

Author: Grathio



Author: **PRO**Grathio **Grathio Labs**

Creative swashbuckler. Writer for MAKE Magazine, presenter of inventions on TV, radio, magazines and newspapers. Professional problem solver. Annoyingly curious. Hacker of all things from computers to clothes to cuisine.

License: [Attribution Non-commercial Share Alike \(by-nc-sa\)](#)   

Intro: Secret Knock Detecting Door Lock

Protect your secret hideout from intruders with a lock that will only open when it hears the secret knock.

This started out as a bit of a joke project, but turned out to be surprisingly accurate at judging knocks. If the precision is turned all the way up it can even detect people apart, even if they give the same knock! (Though this does trigger a lot of false negatives, which is no fun if you're in a hurry.)

It's also programmable. Press the programming button and knock a new knock and it will now only open with your new knock. By default the knock is "Shave and a Haircut" but you can program it with anything, up to 20 knocks long. Use your favorite song, Morse code, whatever.

Maybe a video will explain it better:



Important Notes:

(I hate to even have to say this, but since someone's going to say it, I'll say it first:)

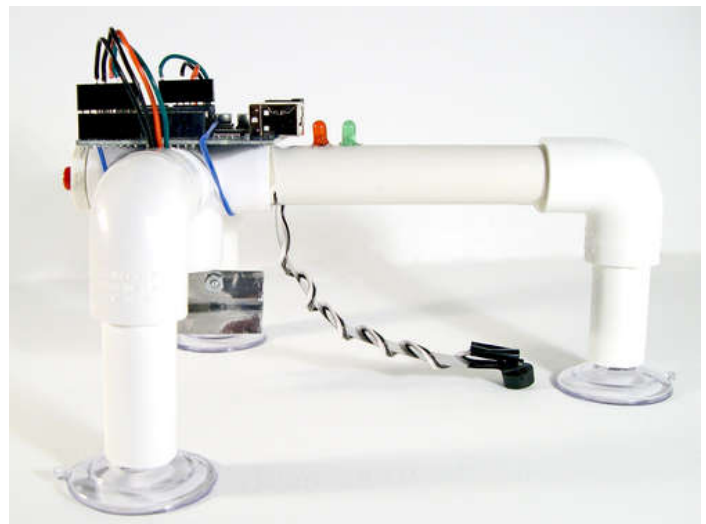
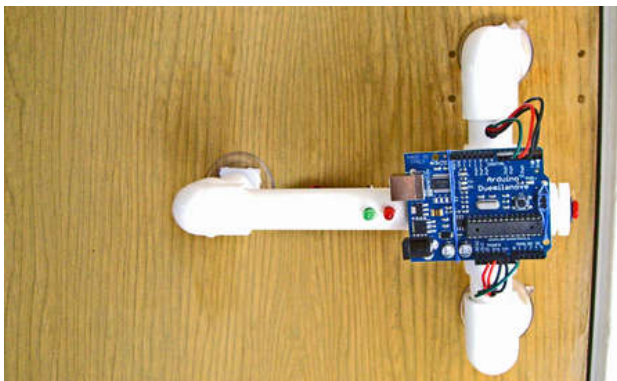
1) *This is for entertainment purposes only. Really. This decreases the security of your door by adding another way to unlock it, and it makes your unlock code known to anyone who can hear. If you put this on your door, be sure to carry your key too. The batteries might die, the suction cups might fail or you might forget your knock. Don't complain to me if someone imitates your knock and steals all your stuff, you've been warned.*

For obvious improvements to safety, security and whatever, see the final page of the Instructable.

2) This is not a project for a beginner! Read through it *carefully* and be sure you understand it **before** you start! I will not take time to answer questions that are already in the instructions or from people who have gotten in over their head.

(If you think this project is too complex you might [go here](#) and sign up for the kit mailing list. The kits will be much more simple than this.)

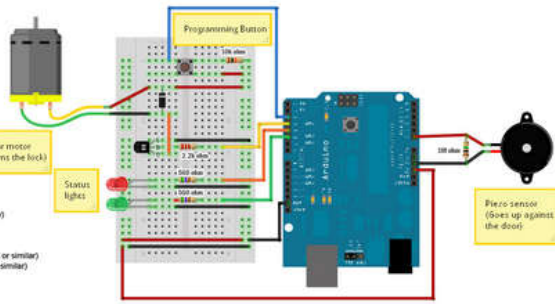
Sorry about that. Now that that's out of the way, lets get to work.



Secret Knock Detecting Lock

Parts:

- Arduino Microcontroller
- 1 Piezo speaker
- 1 High torque gear motor (5v)
- 1 Momentary pushbutton
- 1 Red LED
- 1 Green LED
- 1 NPN Transistor (P2N2222A or similar)
- 1 Rectifier Diode (1N4001 or similar)
- 1 2.2k ohm resistor
- 1 10k ohm resistor
- 1 1M ohm resistor
- 2 560 ohm resistors



grrthls.com

Made with Fritzing <http://fritzing.org>

step 1: Tools, Supplies, And Skills

(If this all looks too challenging, you might consider signing up to the kit mailing list which, when available, will be much easier and a lot more simple.)

Time:

This project will take several hours to complete.

Skills:

To complete this project you should be able to do the following:

These are important! If you're not sure if you have these skills, read through the entire Instructable and make sure you understand it before starting anything!

- Basic soldering.
- Read a basic schematic.
- Basic knowledge of microcontrollers (I'll be using the Arduino.) This means you know what one is, how to upload data to it, and how to make minor changes to code.
- Improvisation. There are many ways to do this project, and you will have to make changes based on how your door and lock works.

Tools:

- Drill (ideally a drill press) and an assortment of drill bits.
- Saw capable of cutting PVC pipe. (ie: Pretty much any saw.)
- Soldering iron and solder.
- Pliers.
- Screw drivers.
- Heat-shrink tubing and/or electrical tape.
- Wire stripper.
- Vice.
- Safety glasses.
- Gloves.

Other things you might find handy: a ruler/tape measure, a multimeter, a breadboard, some tape, a magic marker, sand paper, files, hot glue. And if you're like me a well stocked first aid kit.

Materials:

(The links are for example only, I don't necessarily recommend or have experience with any of these vendors. Feel free to suggest other sources in the comments.)

Electronics:

- 1 Arduino Duemilanove (Or compatible. Or really any microcontroller with at least 1 analog input and 3 digital outputs.) Buy from [here](#), [here](#), or [here](#). And other places.
- 1 5v Gear reduction motor. The higher torque the better. [Here's a good one.](#) (14-16mm diameter is ideal because it fits inside of 1/2" PVC pipe.) I recommend one with at least 15oz/in (11 N-cm) of torque at 5v to turn a basic lock. ¹
- 1 Piezo speaker. (30mm) similar to [this](#). You can use larger or smaller ones, smaller will be less sensitive.
- 1 SPST momentary pushbutton. (normally "off")
- 1 Red LED
- 1 Green LED
- 1 NPN Transistor P2N2222A like [these](#) or [these](#) (or similar).
- 1 Rectifier Diode (1N4001 or similar) [this](#) or [this](#) will do.
- 1 2.2k ohm resistor (1/4 watt)
- 1 10k ohm resistor (1/4 watt)
- 1 1M ohm resistor (1/4 watt)
- 2 560 ohm resistor (Or whatever will run your red and green LED's at 5v. [How to tell.](#))
- 1 small piece of perf board. 5x15 holes or longer. ([example](#))
- 1 9 volt battery clip and 9v battery. (Or any other way you can think of to get 7-12v to the Arduino. A wall adapter like [this](#) is a great option so you don't have to worry about batteries running out. 6 AA's would be another option for longer lasting power, but it will bring down the suction cups.)
- Connector wire. 20 gauge or narrower flexible wire in a number of colors for connecting the electronics together.

It's also a good idea to have a breadboard for setting up and testing the circuit before you solder it. We'll be doing this in step 3.

Case:

(These items are to make the project as pictured. Feel free to build a completely different and more functional case.)

- 20" PVC Pipe 1/2".
- 3 right angle 1/2" PVC connectors.
- 1 5-way 1/2" PVC connector. (example)
- 2 1/2" PVC end plug.
- 3 1 1/2" suction cups. (Available at hardware stores and craft centers.) *NOTE: If your door is unsuitable for suction cups then replace these with three end caps and you can use adhesive strips or screws to mount the lock.*
- 6" of 1/2" wide by 1/64" thick metal strip (steel, tin, copper, etc.) (available at hardware, craft, and art supply stores.)
- 4.5" of 1" wide metal sheet, 1/32" thick (steel, tin, copper, etc.) (available at hardware, craft, and art supply stores.)
- 2 3/32" x 3/8" screws with nuts. (1/8" will work too if you can't find the smaller ones.)
- 2 1.6M (metric) 16mm screws. Ideally with countersunk heads if you can find them. (For securing the motor. Check your motor specs to see what screws it needs. One motor I tried used 1.6M, the other 2M. You'll probably have to buy long ones and cut them to length.)

¹ If you have a torque meter or a torque wrench, apply it to your door lock to get an idea of what torque it will take to open your lock. Use a online conversion tool to convert between foot/pounds, N/m, etc.

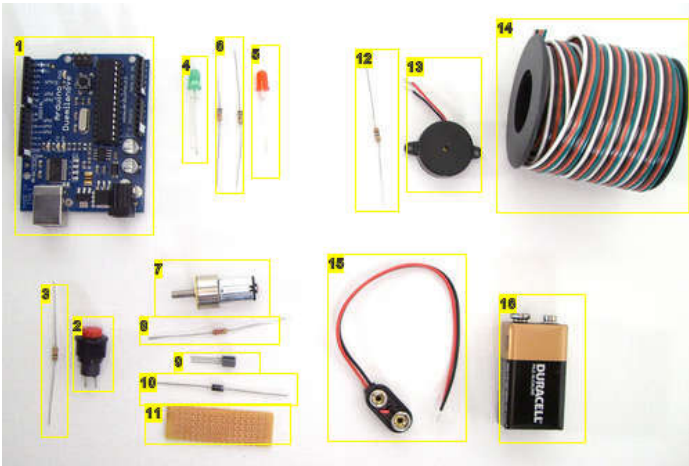


Image Notes

1. Arduino microcontroller.
2. Momentary pushbutton
3. 10K ohm resistor (brown, black, orange)
4. Green LED
5. Red LED
6. 560 ohm resistors. (Green Blue Brown)
7. Gear motor
8. 2.2K ohm resistor (red red red)
9. 2n2222 Transistor (NPN type)
10. Rectifier diode (1N4001)
11. Perf board 5x15 holes.
12. 1M ohm resistor (brown, black, green)
13. Piezo speaker
14. Some wire, 20-22 gauge. The more colors the better.
15. 9v battery connector
16. 9v battery. You can also run this project from the appropriate wall plug.

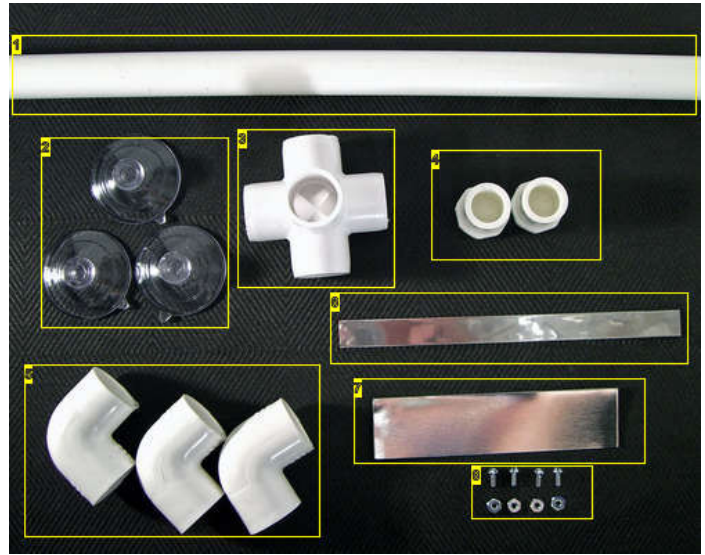


Image Notes

1. 20 inches of PVC, 1/2"
2. 1 1/2" diameter suction cups.
3. 5-way PVC connector. (Can be a little tricky to find.)
4. PVC end plugs
5. PVC right angle (90 degree) connectors
6. 1/2" wide metal strip used for a spring to press the detector to the door.
7. This metal strip will be made into the widget that connects our motor to the lock.
8. Screws and matching nuts. (Turns out you only need 2 pairs, not 4.)

step 2: Program The Arduino

This section assumes that you know how to connect your Arduino microcontroller to you computer, compile and upload a sketch. If you don't know how to do that you probably shouldn't be doing this Instructable. But spending some time on this page and doing some of the examples and tutorials there might bring you up to speed.

We're going to upload our sketch before doing any of the electronics so we can test the electronics as we go.

#1: Download

Download the file `secret_knock_detector.pde` at the bottom of this section and copy it to your sketchbook. (Or view the text and cut and paste it into a new sketch.)

(Tip: If the name of the downloaded file is something like "BARS5HS13H8SW.tmp" simply rename it to `secret_knock_detector.pde`. and you're good to go.)

#2:

Open the sketch and compile it. It should compile properly the first go, but it's good to be sure.

#3:

Connect your Arduino and upload the sketch.

If you have any trouble, check the troubleshooting section at the Arduino site.

Code overview:

For the curious, here's a look at a few bits of code if you're interested in tinkering: (If you're not curious, go to the next section)

```
about Line 28: const int threshold = 4;
```

This is the sensitivity of the knock detector. If you get a lot of noise, raise this (up to 1023), if you're having a hard time hearing knocks you can lower it (as low as 1).

```
about Line 29: const int rejectValue = 25;
about Line 30: const int averageRejectValue = 15;
```

Both of these are used to determine how accurately someone has to knock. They are percentages and should be in the range of 0-100. Lowering these means someone must have more precise timing, higher is more forgiving. `averageRejectValue` should always be lower than `rejectValue`.

Settings of about 10 and 7 make it hard for two people to knock the same knock even if they know the rhythm. But it also increases the number of false negatives. (ie: You knock correctly and it still doesn't open.)

```
about Line 31: const int knockFadeTime = 150;
```

This is a crude debounce timer for the knock sensor. After it hears a knock it stops listening for this many milliseconds so it doesn't count the same knock more than once. If you get a single knock counted as two then increase this timer. If it doesn't register two rapid knocks then decrease it.

```
about Line 32: const int lockTurnTime = 650;
```

This is now many milliseconds we run the motor to unlock the door. How long this should be depends on the design of your motor and your lock. It's okay if it runs a little bit long since I've designed a simple slip clutch into the design, but it's better for all the parts if it doesn't run too much.

```
about Line 34: const int maximumKnocks = 20;
```

How many knocks we record. 20 is a lot. You can increase this if your secret hideout is protected by devious drummers with good memories. Increase it too much and you'll run out of memory.

```
about Line 35: const int knockComplete = 1200;
```

Also known as the maximum number of milliseconds it will wait for a knock. If it doesn't hear a knock for this long it will assume it's done and check to see if the knock is any good. Increase this if you're a slow knocker. Decrease it if you're a fast knocker and are impatient to wait 1.2 seconds for your door to unlock.

```
about Line 39: int secretCode[maximumKnocks] = {50, 25, 25, 50, 100, 5....}
```

This is the default knock that it recognizes when you turn it on. This is weird rhythmic notation since every value is a percentage of the longest knock. If you're having a hard time getting it to recognize "shave and a hair cut" change this to `{100,100,100,0,0,0...}` and a simple sequence of 3 knocks will open it.

Debugging:

```
about Line 51: Serial.begin(9600);
about Line 52: Serial.println("Program start.");
```

Uncomment these lines to see some debug info on the serial port. There are a few other lines of debugging code set throughout the rest of code that you can uncomment to see what's going on internally.

Be sure to set your serial port to the right speed.

The rest of the code is commented so you can see how it works but you probably won't need to change it if you aren't changing the design.

File Downloads



[secret_knock_detector.pde](#) (9 KB)

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

step 3: Lay Out And Test The Circuit

We're going to breadboard the electronics to make sure everything works. If you never make mistakes you can skip this step.

I've provided both a schematic and a layout diagram for the breadboard. Follow whichever one you're the most comfortable with.

We're going to go slowly and check as we go.

#1: Wire the Piezo Sensor

Solder a pair of 12" (30cm) leads to the Piezo speaker. Connect it between Analog pin 0 and the ground. Also attach the 1M ohm resistor between Analog pin 0 and the ground.

Test: With your Arduino plugged into your computer (via USB or Serial cable) and open the Serial Monitor window. (That's the button furthest to the right at the top of the Arduino development environment.) With the Arduino powered on you should see the text "Program start." Tap the piezo speaker and you should see the text "knock starting" and "knock" each time you tap it. Stop for a second or two and you'll probably see "Secret knock failed" or "Door unlocked!"

If you don't see anything or see junk, make sure your serial port is set to 9600 baud and reset the power on the Arduino. If you're sure it's right, then try tapping *Shave and a Haircut* (Don't forget the two bits. See the video if you don't know it.) and see if you can get the "Door unlocked!" message.

If you get knock messages without tapping it may be too sensitive. If so you'll need to edit the sketch. Around line 27 raise the value of `threshold`. This can be raised as high as 1032 if you have a very sensitive detector.

```
const int threshold = 3; // Minimum signal from the piezo to register as a knock
```

Once you have it working the way you want it you can comment out (or delete) the lines that start with `serial...` We shouldn't need them any more.

#2: Wire up the LEDs

Lets wire up some LEDs so we don't have to use a serial cable to see what's going on.

Connect the red LED to digital pin 4 and green LED to digital pin 5 with their corresponding 560* ohm resistors in line.

Test: If you power the circuit the green LED should light. If not, check your connections and make sure the LED is the right way around. Every time you tap the green led should dim. After tapping the correct sequence the green led should blink a few times. Tapping the wrong sequence should blink the red one.

If none of this happens, check the polarity on your LEDs and all of your connections.

<http://www.instructables.com/id/Secret-Knock-Detecting-Door-Lock/>

* Your LEDs might require different resistance.

#3: Wire the programming button

Solder 8" leads to the button. Connect one side of the button to +5v. The other pin on the button connect to digital pin 2 and, with a 10K resistor to the Ground.

Test: Apply power. When you press the button the red light should come on. Hold down the button and tap a simple sequence. When tapping while programming both LEDs should blink. When you're done the pattern you just tapped should repeat on both lights. After playback is complete, the new knock code is saved and the lights will alternate red and green to tell you so.

#4: Wire in the motor

Solder 8" of leads to the motor and follow the design/schematic. Be sure to get the diode going the right way and you might want to check the pins on the transistor to be sure they match the diagram. (Some transistors might have the pins in different order.)

Test: Power the circuit. Tap the default "Shave and a Haircut" knock. The motor should run for about half a second. If not, check your connections as well as the polarity of the diode.

Extra Troubleshooting tips:

1) If the motor turns *very* weakly the diode might be reversed.

2) If you need more power on your motor make the following circuit change: Move the wire that goes from the motor to +5v to the Vin pin on the Arduino. This will supply the motor with 9v (or whatever voltage you're supplying to it.)

Tip: Check which way the motor turns. It should turn the same way as you turn your deadbolt lock to unlock it. If not, switch the motor's leads which should reverse the motor.

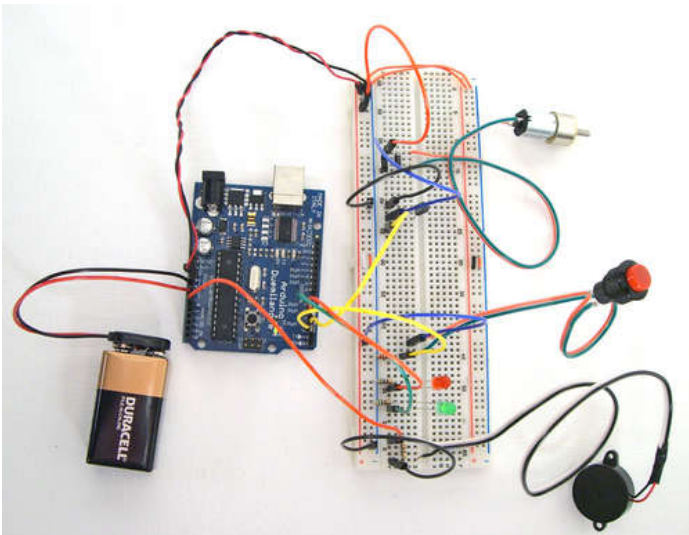
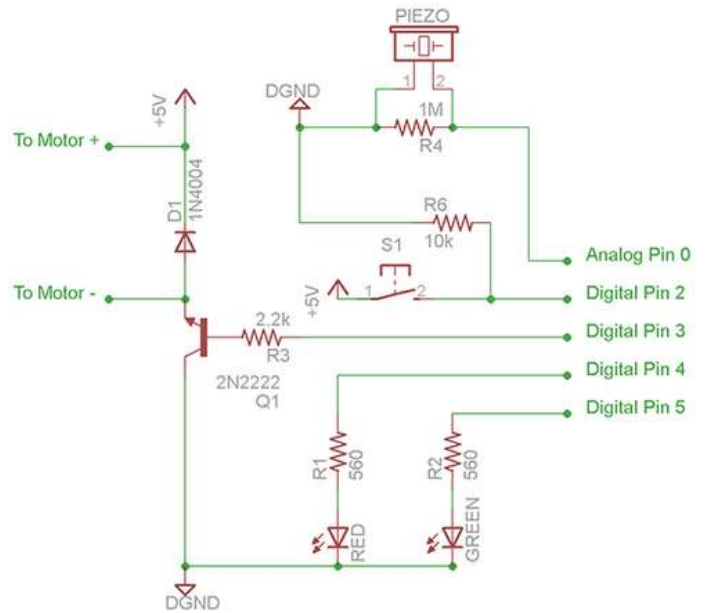
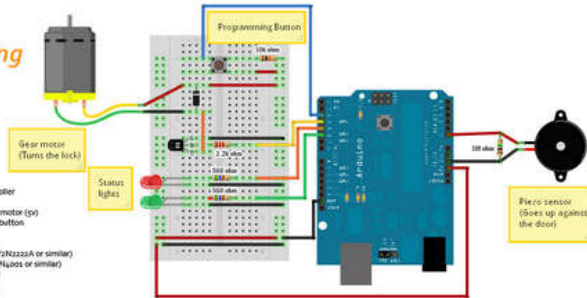
Congratulations! You have a working secret knock detector!

Now we have to put it into something more permanent that we can stick on our door.

Secret Knock Detecting Lock

Parts:

- Arduino Microcontroller
- 1 Piezo speaker
- 1 High torque gear motor (5v)
- 1 Momentary pushbutton
- 1 Red LED
- 1 Green LED
- 1 NPN Transistor (2N2222 or similar)
- 1 Resistor Diode (1N4004 or similar)
- 1 2.2k ohm resistor
- 1 10k ohm resistor
- 1 1M ohm resistor
- 1 560 ohm resistor
- 1 560 ohm resistor



step 4: Prepare The Case

If you're making your own case you can skip this step. Otherwise grab your PVC and saw and lets get cracking.

Important!

We're just testing for fit here. Don't glue or fasten anything yet!

#1: The Button mount

Take one of the PVC end caps and drill a hole through the center appropriate for your push button. For my button that was 3/8". Secure the button in the hole.

Plug this into one of the 4 radial holes in the 5-way connector. (ie: not the one that points down.)

#2: The Motor mount

Take the other PVC end cap and drill a hole big enough for the shaft of your motor to pass through. You might also need to make it even bigger if your motor has a bearing that sticks out.

Depending on the design of your motor you may want to sand down the thickness of the plug so that you have more of the motor shaft to work with. Test the fit by placing the motor through the back of the plug. If it's too tight you might have to sand/file/grind the inside of the plug so it will fit.

Use a paper template to place the holes for the fastening screws, drill the holes and attach the motor to the plug. (In my case using the two 2M screws.) Countersink the screws if possible.

Plug this into the "down" facing hole on the 5-way connector.

#3: The 'arms'

Cut one piece of PVC pipe 5 inches long. We're going to call this the "long arm". Put a right angle connector on one end. Plug the other end into the 5-way connector opposite the button.

Cut two pieces of PVC pipe two inches long. We'll call these the "short arms". Half way along their length drill a 1/4" hole through one side. Put right angle connectors on one end of each arm. Plug these into the two remaining holes on the 5-way connector. You should really start to see it take shape.

#3b: A Few Extra Holes

with a pencil or marker draw a line down the center of the top and the bottom of the long arm. On the top side, make marks for two holes, one 3/4" from the 5-way, and another 1 1/2" from the 5-way. Drill a 3/16 (5mm) hole at each of these places. This is where our LEDs are going.

Also make a line along the bottom where the long arm connects to the 5-way. Using a saw, cut a short way through the pipe, from the bottom up, until there is about a 1/2" hole into the pipe. (this is where the spring for our detector will attach. Also on the bottom, drill a 1/8" hole 1/4" further along the pipe (Away from the 5-way). We will thread the sensor's wires through here.

#4: The 'legs'

These are the parts that attach to the door. You may not want to cut these yet, The length depends on the design of your door lock, the length of the shaft on your motor and the final design of the Lock Turning Clamp in the next step. All three of mine were 2 5/16" long, but you're better off cutting them long and trimming them down to size later

If they're too long the motor won't reach the lock to turn it. If they're too short the suction cups won't reach the door.

When you do cut these, hot glue the suction cups in one end and stick the other ends in the right angle connectors on the ends of our legs.

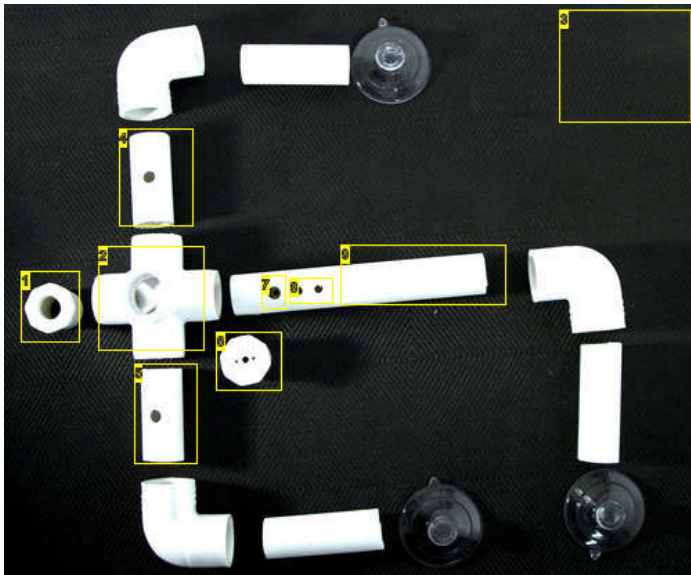


Image Notes

1. Button mount. End plug with hole drilled for the panel mount button.
2. 5-way connector. The motor goes in the center of this.
3. This is an exploded view with all of the parts laid out how they attach (though the angles are wrong on some parts).
4. Top arm. Hole drilled to let wires through.
5. The "Board Arm" This is the arm we're putting the circuit board into. Hole drilled to allow wires through.
6. Motor mount. Will go in the bottom hole on the 5-way.
7. Ignore this hole. Not sure what I was thinking.

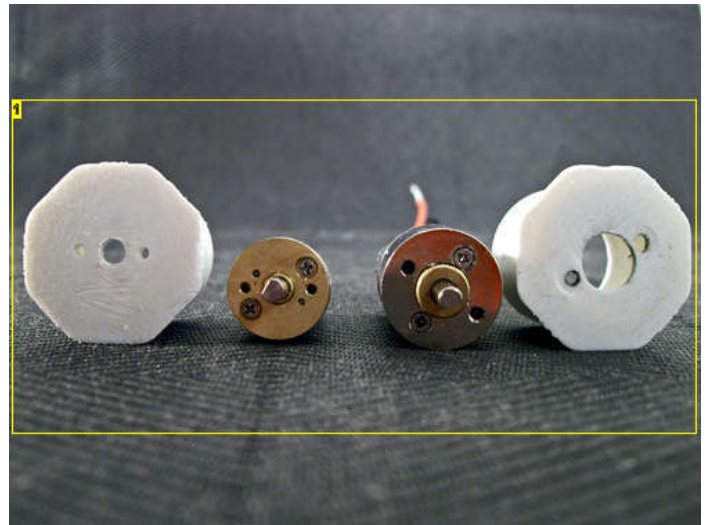


Image Notes

1. Two sample motor mounts for two different motors. The one on the left is smaller but actually more powerful. The one on the right has a larger hole to allow clearance for the bearing. The facing surface on both of these have been ground down so more of the shaft can stick out.

8. holes for LEDs.
9. The "Long Arm". (It's longer because it provides more holding power against the torque of the turning lock.)

step 5: Make The Lock Turning Clamp

In this section we make the all important part that connects the motor to the lock.

(This is a reasonably clunky way to do this, but it's simple and cheap. If you think of a better way, please mention it in comments.)

What we're making is a clamp that attaches to the D-shaft of our motor and fits easily over the lock latch so that it can turn the lock. It attaches securely to the motor, but there is some give in it so that it can slip if it finds its self between a rock and a hard place. (Which we prefer to wrenching the project to pieces.)

Drilling the holes:

First take the piece of metal that's 4 1/2" long and 1 1/4" tall. and cut it in half so you get two 2 1/4" pieces.

Tape them together, mark each side so you know which side is "out", and mark one of the long edges as "up". This will all help you keep everything lined up as you go.

Measure and mark the center line from top to bottom. 1/2" each side of this center line and 1/4" from the top mark holes for drilling.

Drill 1/8" holes at these marks. Marking the points with a punch, or giving it a whack with a hammer and nail will make your drilling more accurate.

The edge with the holes is the side that attaches to the motor.

Bending the metal

Measure the width of your lock latch (the narrow way) and divide by 2. This distance is how much *zig* we're going to bend into each piece of metal. Mark this zig distance along each strip. Bend one piece so it zigs to the left, the other so it zigs to the right. Make sure that the screw holes at the top of the pieces stay lined up and the bends don't keep the pieces from meeting at the top.

Finishing and sizing

For this part you'll need your motor, the two 1/8" screws and a couple matching nuts.

Put the screws through the holes in the top of the plates so it makes an upside down "Y" (sort of) and place the motor shaft in the top between the screws. Screw nuts on each side and tighten until it's firmly (but not *really* firmly) attached. The small amount of give between the metal and the shaft will let the motor spin if it meets too much resistance. (Rather than breaking something important.)

Check the other end for fit over the lock. It should fit a little loosely over the lock latch. Not so firmly that it's clamped tight, but not so loose that it can turn without turning the lock. Adjust the bend of the flanges if you need to.

After you've got the adjustments right, tighten another nut onto the end of the screws and tighten them up against the first ones. This will help lock them in place.



Image Notes

1. It's not a bad idea to file down the sharp corners.

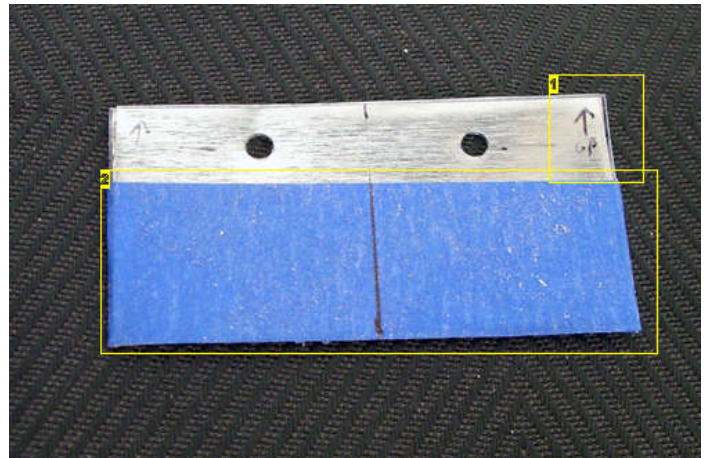


Image Notes

1. A few markings will help keep them straight when you put them back together.
2. Two pieces taped together so we get the holes lined up.

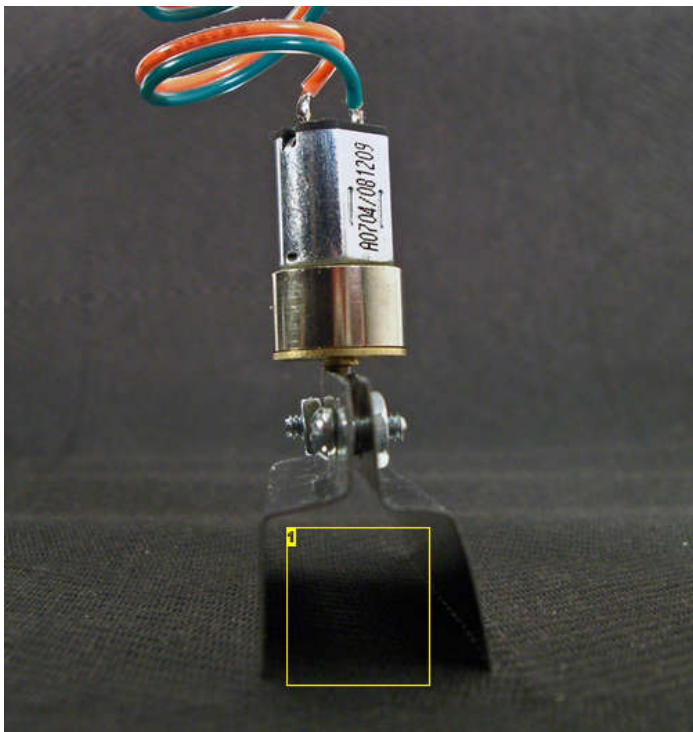


Image Notes

1. Edge view. Your lock will slot in here.

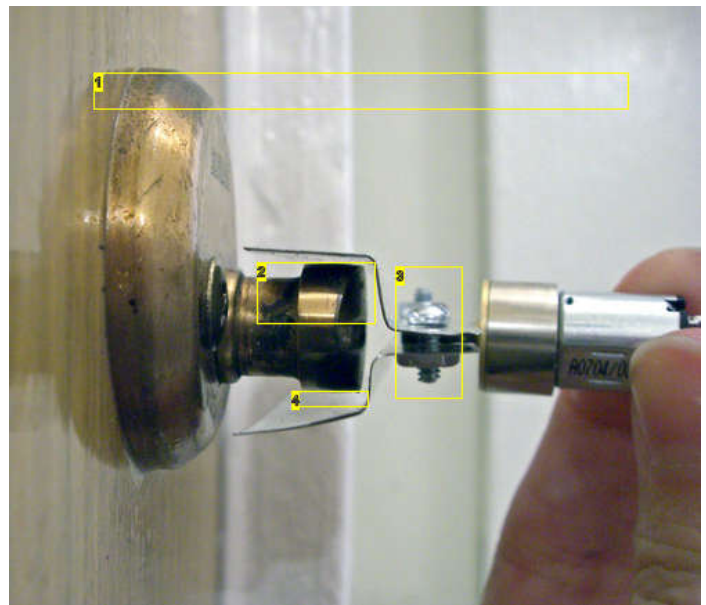


Image Notes

1. In position over the lock.
2. 1/2 of the thickness of the lock handle. This is how much we "zig" each flange.
3. Tighten the screws so they clamp down on the D-shaft.
4. Having a little bit of space is fine and will make it easier to put on.

step 6: Make The Knock Detector Spring

In this section we're going to put our knock detector on the end of a little springy bit so it presses securely up against the door. You could just use a piece of tape or even glue or screw it straight to your door, but doing it this way keeps it portable.

#1: The parts:

You'll need

- Your strip of thin metal (which is 6" of thin 1/2").
- The piezo sensor (which should have about a foot of leads soldered on).
- The piece of PVC I've been calling the "long arm".

The PVC pipe segment, on the bottom side, should have a slot cut 3/4" from the end and a 1/8" hole just inside of it.

#2: Attach the sensor to the metal strip.

Using glue, hot glue, tape, etc and fasten the piezo sensor to one end of the metal strip. Wrap some of the remaining wire around the strip so that it stays out of the way.

(If your piezo sensor has its leads on the back then drill a hole through the strip. be sure to cover the leads with insulating tape or heat-shrink.

#3: Attach the metal strip to the PVC.

Thread the free end of the wire through the bottom hole on the PVC and then insert the free end of the metal strip in the slot. Bend the strip as shown so that the sensor faces out and down and will lay flat on the door.

The strip should stick in the slot with friction, but if not, take some pliers and bend over the end of this strip that's inside the pipe.

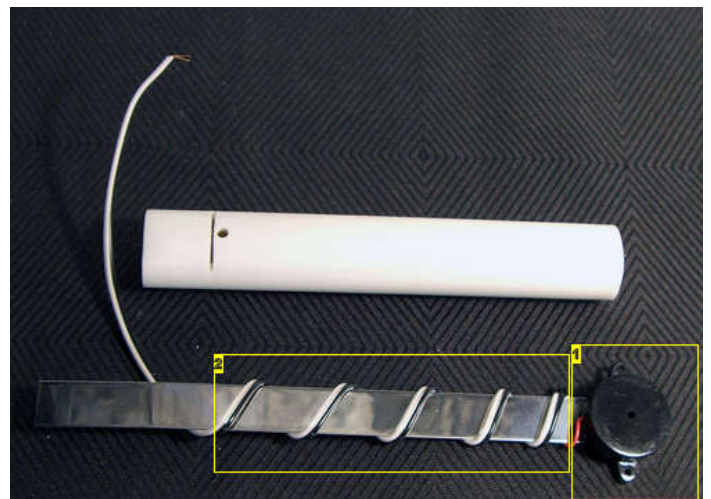


Image Notes

1. Hot glue (or otherwise attach) the sensor to the end of the metal strip. Make

Image Notes

1. The bottom side of the Long Arm with the 1/2" slot for the metal strip and a hole for the wire.
2. 4-6" flexible metal strip.
3. Piezo sensor with about a foot of wire attached.

sure the "open" end is facing out.

2. Wrap some of the remaining wire around the strap to keep it out of the way.

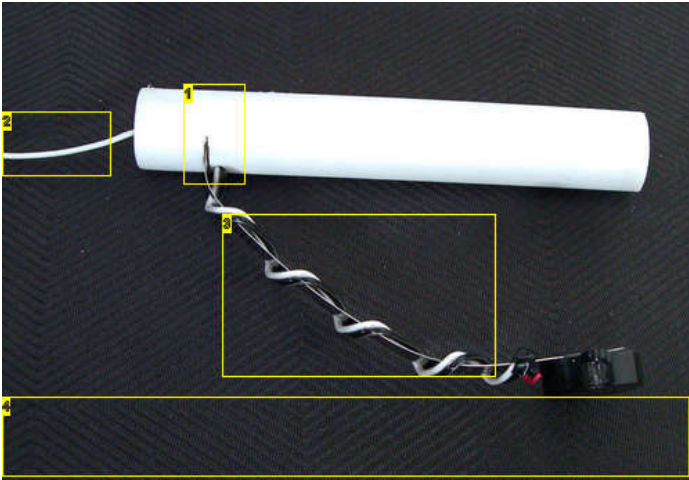


Image Notes

1. Thread the wire through the round hole and slide the free end of the metal strip into the slot.
2. The free end of our wires.
3. Bend the strip more or less like this.
4. This area is where the door will be, Try to bend it so the sensor lies more or less flat against the door.

step 7: Soldering The Circuits

Due to the needlessly complex nature of my case, soldering and case mounting are somewhat intertwined, but I'll try to break it down so it makes sense.

I recommend that after each step you plug in and test the circuit to make sure you didn't make a mistake, moving each bit from the breadboard one at a time. Having to desolder components is no fun.

Tip: Use wires in as many colors as you can get so you can keep things straight. I also usually put labeled bits of tape on the ends of the wires to help me remember.

Tip: If you're using stranded core wire, be sure to tin the ends. It will help with your joints and make it easier to stick them into the breadboard for testing.

#1 Solder leads to the LEDs. (Hey, that almost rhymes!)

About 10" or so should work.

Okay, so much for the easy steps. After this it gets more complex because most of the components need to be threaded through various holes in our case before they're soldered. Of course if you made a different case then you don't need to worry about most of the tedium and can get right to soldering the components to the perfboard.

Tip: Mount the components as closely as possible to the perfboard. There isn't much clearance inside the pipe.

#2 Prepare the perfboard.

We're using a perfboard with 0.10" spacing where each hole has an individual copper pad. Cut the perf board to size (5x15 holes) and then sand/file/grind off some of the edges so it fits easily into the 1/2" PCV pipe.

For future reference we're calling the side with the copper "back" and the side with the components the "front".

#3: The ground line

Since the ground is the most common terminal in the project we're going to run a ground line across the back for the components to connect to. To make this I too a 9 inch piece of solid core wire that I'm using as my ground wire and stripped about an inch off one end. The soldered between hole 1 and hole 10 (see the attached diagram). Then I'll tack the other grounded components to it as the come through the board. (You can also just bridge the connections with solder, but I hate doing that because it can get messy. My soldering is messy enough.)

The other end of this wire will go to a Ground pin on the Arduino. (This is a good time to label the other end with a piece of tape.)

#4: The +5v line.

There are also a couple points where we want to supply +5v. This is the same idea as the ground line but we only need about half an inch stripped.

The other end of this will connect to the +5v pin on the Arduino.

#5: The LEDs.

Solder the LED's resistors (560 ohm by default) in place as shown.

You have two choices of how to deal with the LED's. You can mount them on top (the easy way) or you can mount them from the bottom, which looks better, but is a pain because 1/2" PVC doesn't give you much room to work. If you mount them from the top, be sure to thread the leads through the holes before soldering.

Thread all 4 leads from the leds out through the near end of the 'long arm' through the 5-way and through the 'board arm'.

Solder the cathode (-) lead (the short one) from each LED as indicated. The anode will connect to digital pin 4 (red) and 5 (green) on the Arduino. (Thread the Arduino leads through the "short arm" of the 5-way.)

#7: The knock sensor.

Solder the speaker's 1M ohm resistor in place on the board.

Make sure you have the speaker mounted firmly at the end of the spring and the wire is wound a few times around it to keep it out of the way. Thread the wire through the long leg, through the 5-way and into the short arm that we're keeping the circuit board.

Solder one end of these leads to each side of the 1M resistor. Then solder a 8" lead from the ungrounded end of the resistor. This will go to Analog pin 0.

#6: The button.

Solder the 10K ohm resistor in place as shown.

Fasten the button through the hole on the end plug, then put the plug on the 5-way connector and thread both wires through to the 'board arm' hole.

Solder one lead from the switch to the resistor. The other end to the +5v wire.

Solder a length of wire from the resistor according to the diagram and label it "Digital 2".

#8: The motor.

Nearly done.

Solder the diode, transistor and resistor in place. (Make sure you get the direction right on the diode. And the transistor for that matter.)

To the free end of the 220 ohm resistor solder a 8" lead that will go to digital pin 3.

Put the motor in place in the bottom hole of the 5-way connector, thread the leads trough and solder them in place on either side of the diode, making sure you've got the motor wires in the right order so when it runs it will turn to unlock.

#9: The Arduino pins

Connect the labeled wires to their appropriate places on the Arduino.

Test:

Wait, you don't need to do this, right? You've been testing as we go, haven't you?

Plug some power into the Arduino and make everything works. Especially make sure that the motor spins in the right direction to unlock your lock.

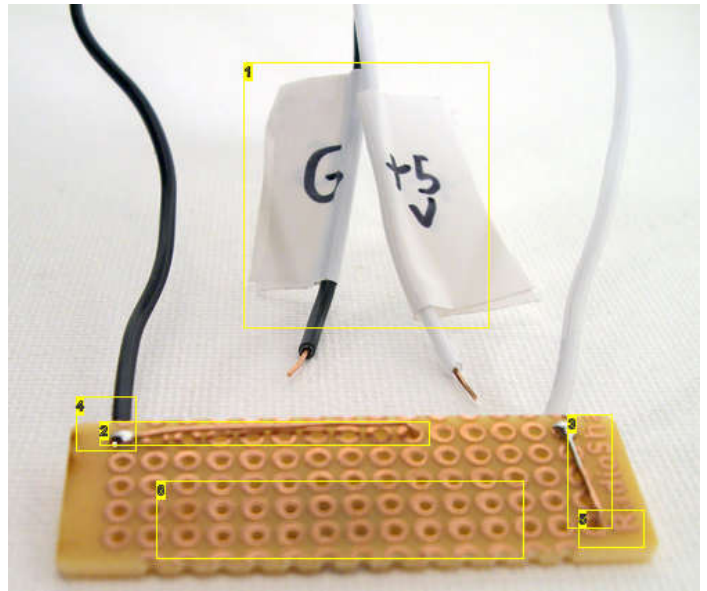
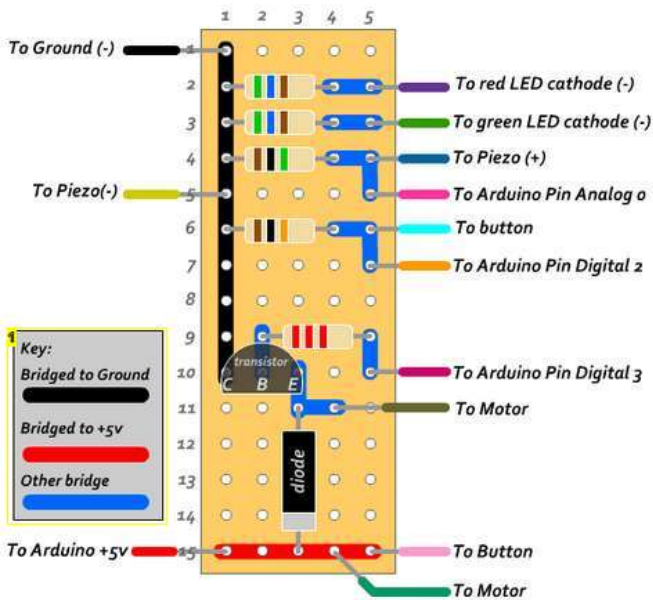


Image Notes

1. Tape labels are a good idea for this project if you're making the PVC pipe case. Misrouting wires is a real possibility.
2. The ground wire. We'll tack other components to this as we add them to the board.
3. The +5v wire. We'll tack other wires to this as we add them to the board.
4. In the soldering diagram this hole is hole #1,1
5. On the soldering diagram this hole is #5,15.
6. This is the bottom of the board, compared to the soldering diagram.

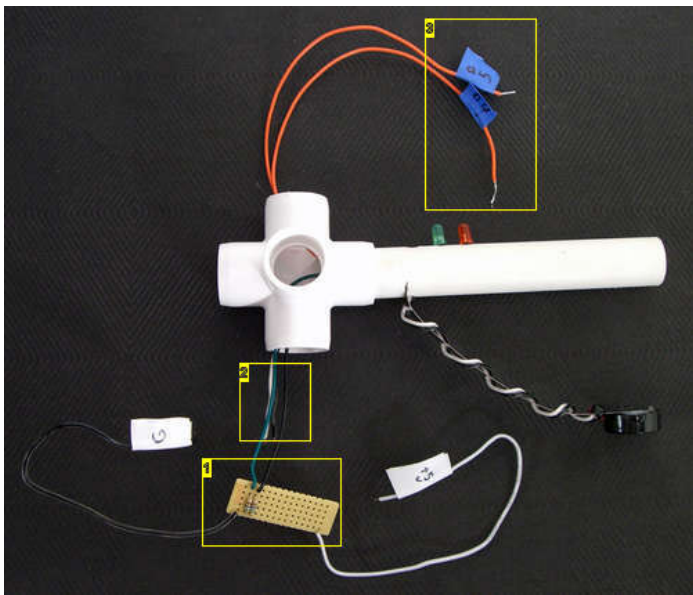


Image Notes

1. #5: The solder the resistors in place to the ground line, and the short leads of the LEDs to the resistors. Pass the leads through the Board Arm hole of the 5-way.
2. While we're threading wires we have threaded both the Piezo sensor wires through for the next step.
3. The + leads for the LEDs come out up here. And we mark them with tape so we don't confuse them later.

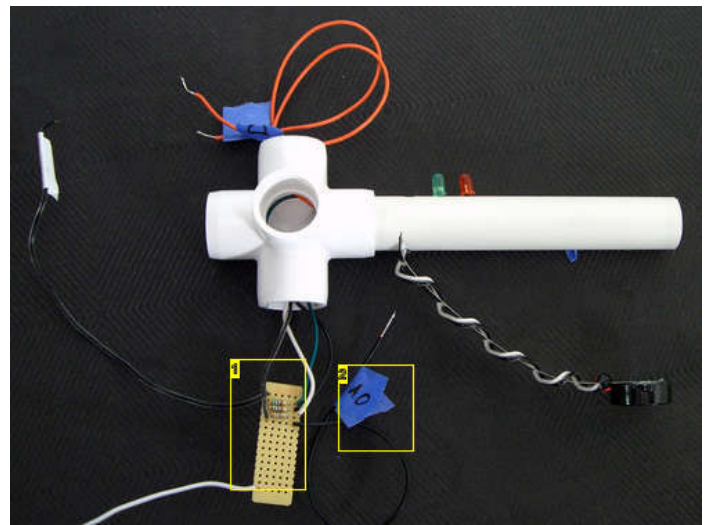


Image Notes

1. The 1M resistor soldered in with the piezo sensor attached either side.
2. A lead attached that will go to Analog 0. (We don't need to thread this anywhere for right now.)

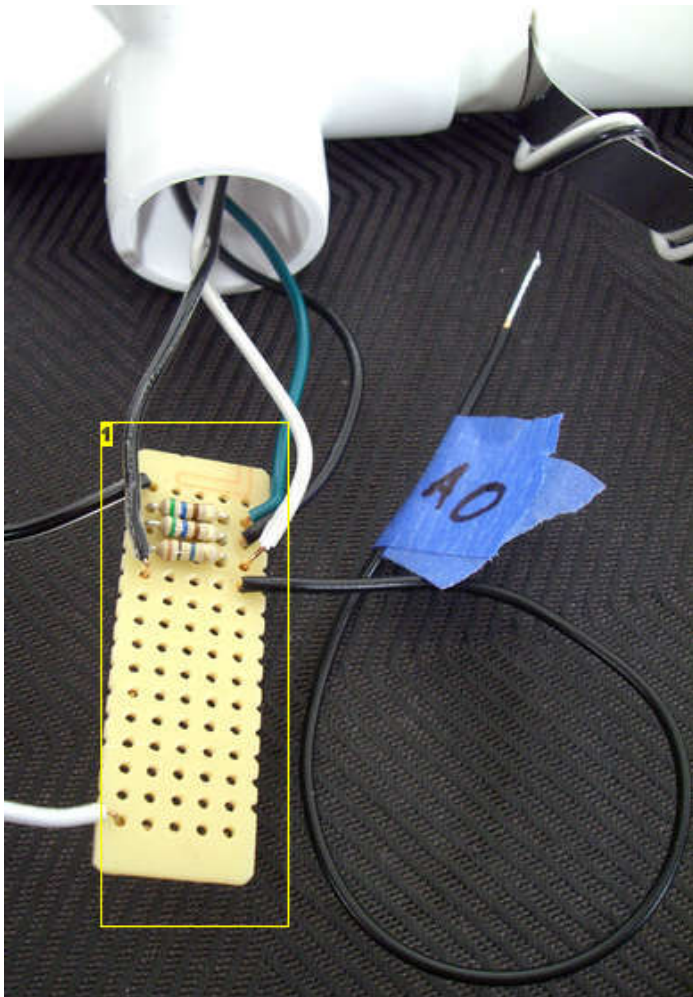


Image Notes

1. Closer look at the resistors and their connections. This is what it should look like after #7, above.

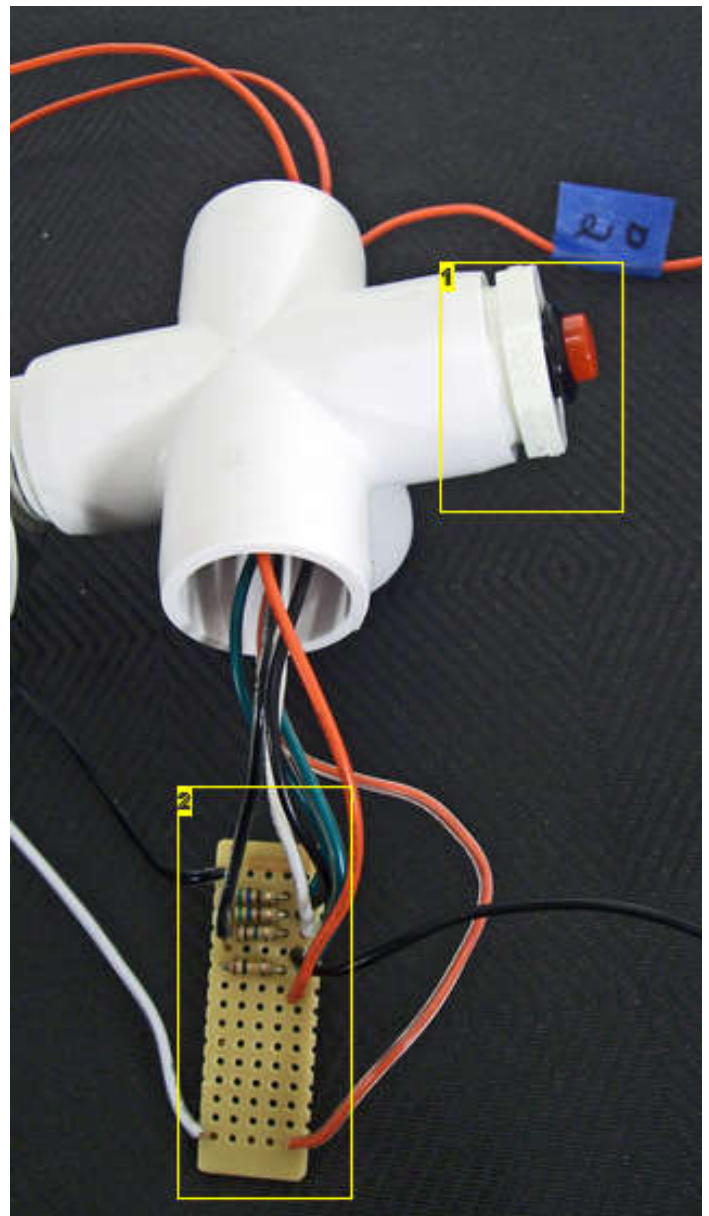
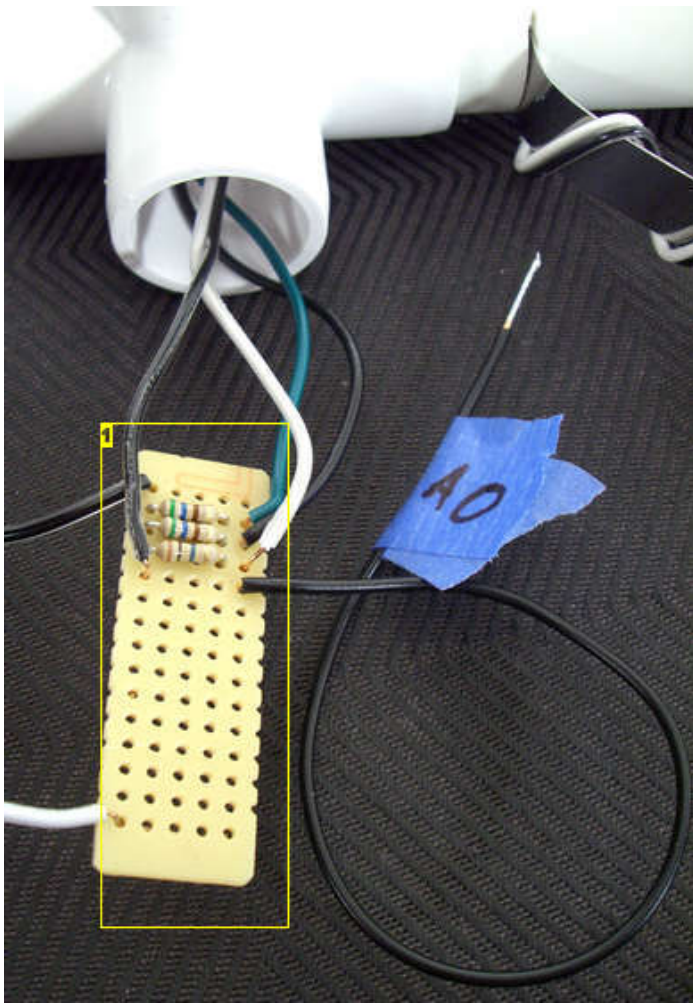


Image Notes

1. The button in place with the wires fed through the 5-way.
2. The resistor and button connections.

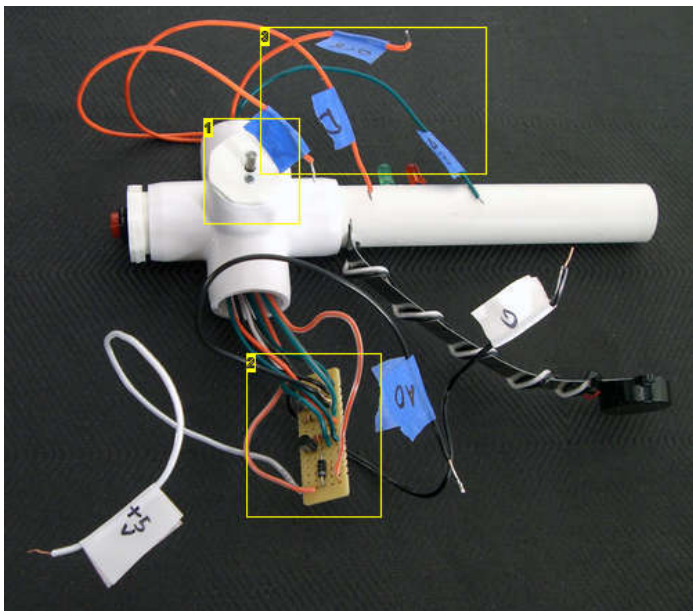


Image Notes

1. The motor in place.
2. The final circuit with the wires fed through.
3. All the connections that will go to the Digital side of the Arduino we thread through to the Short Arm side.

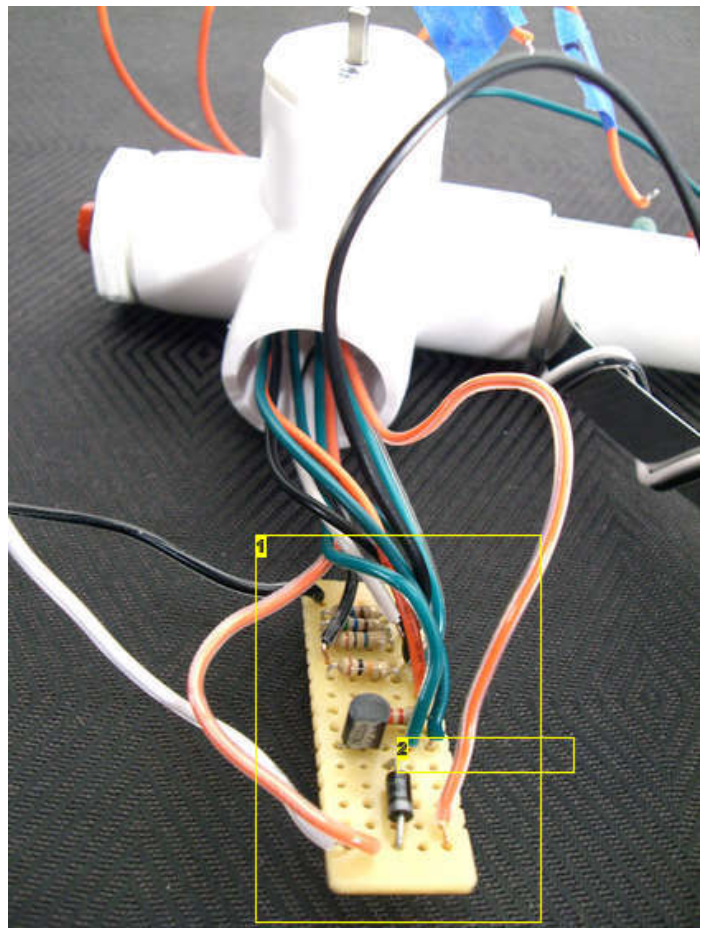


Image Notes

1. Close up of the completed board.
2. Embarrassing note: the motor connection (the green connection on the left) is not in the same hole as the diagram. It's connected the same, just in the wrong hole.

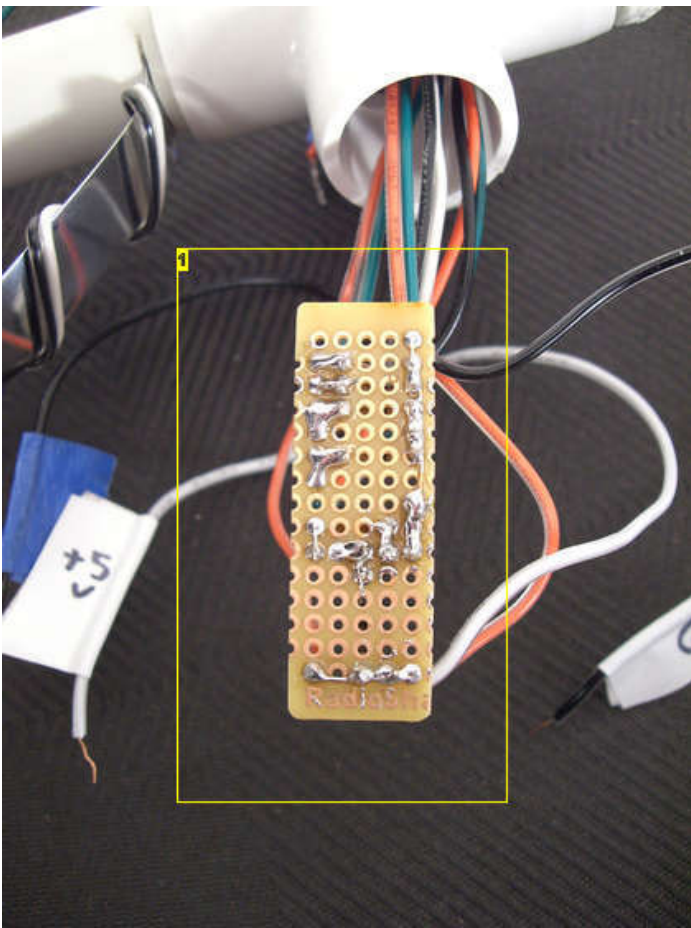


Image Notes

1. Here's the back of the board so you can see the bridged connections. And a lot of messy soldering.

step 8: Assembling The Case

All of your components should be in place and the circuit should be working. We're almost there, all we have to do is cram all that wire and the circuits into the pipe.

#1 Getting the wires to the correct side.

If you had the wires connected to the Arduino, unplug them.

The wire to Analog 0 and to Ground and +5v will come out of the Board Arm, so we don't do anything with those yet.

The other wires to Digital 2,3,4 and 5 thread through the 5-way to where the Short Arm will go.

#2: The Short Arm

Speaking of the short arm... Pull the wires for Digital 2,3,4 and 5 through the hole in the middle of the Short Arm pipe.

#3: The Long Arm.

The long arm has the LEDs and the sensor in it. Using needle nose pliers, (or a bit of coat hanger with a small hook on the end, or a crochet hook) pull up the slack on these wires as you plug it into the 5-way.

#4 The circuit board.

The circuit board should be the first part put in place inside the pipe for the Board Arm. Thread the wires for Ground, +5v and, Analog 0 through the small hole on top of the arm.

Now make a tight bundle of the wires around the circuit board putting even pressure on it, being careful not to bend, break or spindle the thing. Gently slide it into the Board Arm. If you have a lot of extra wire lengths you might want to push it out the far side about half an inch so there's more room for wire on the inside.

When it's in place, plug this short arm into the 5-way.

#5 The Motor.

The motor should already be in place in the bottom of the 5-way. But if it's not now's the time to put it there.

#6: The Button.

The button should also be in place, but if not, put it in. If you have a bunch of extra wire getting jammed up inside the 5-way, you can try pulling some of it (gently!) to the button side of the 5-way since it doesn't take up much space.

#7: The legs and suction cups.

The arms should all be plugged in. Attach 90° turns to the ends of the 3 arms, and plug the legs in to the other end.

Suction cups should fit snug into the bottom of the legs. If not, some hot glue will get them into shape.

(If you're not using suction cups then this is where you use your alternate solution.)

<http://www.instructables.com/id/Secret-Knock-Detecting-Door-Lock/>

#8: The Arduino and battery

Yes, this is ugly as sin. I works, but... yeah. If you come up with anything better, you're welcome to it.

Stick the Arduino onto the top of the frame. I used lengths of insulated wire. It worked...

Attach the battery in a similar way somewhere where it can power the Arduino. Again, I used insulated solid core wires. At one point I used rubber bands which also worked just fine.

Tape? Yes, that would work too.

Plug in the wires in where they labels say they should go. Might as well test again it since it's all hooked up.

Whew! Now we're ready to attach it to the door!

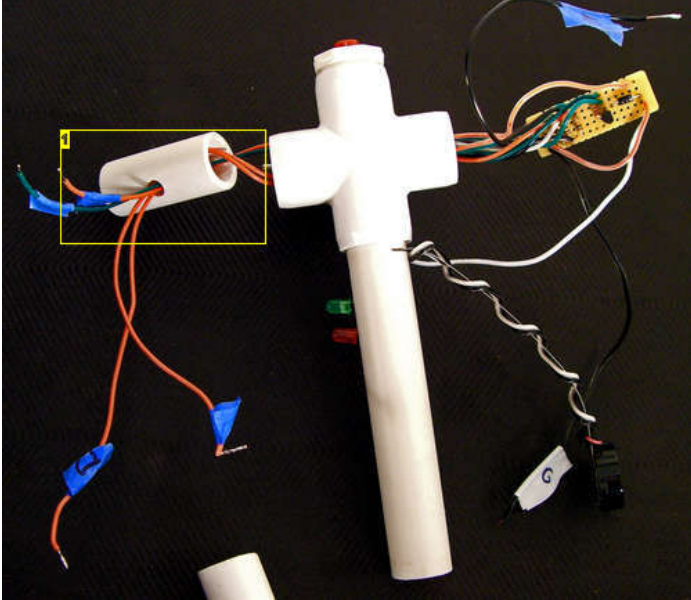


Image Notes

1. The digital pin wires threaded through the Short Arm

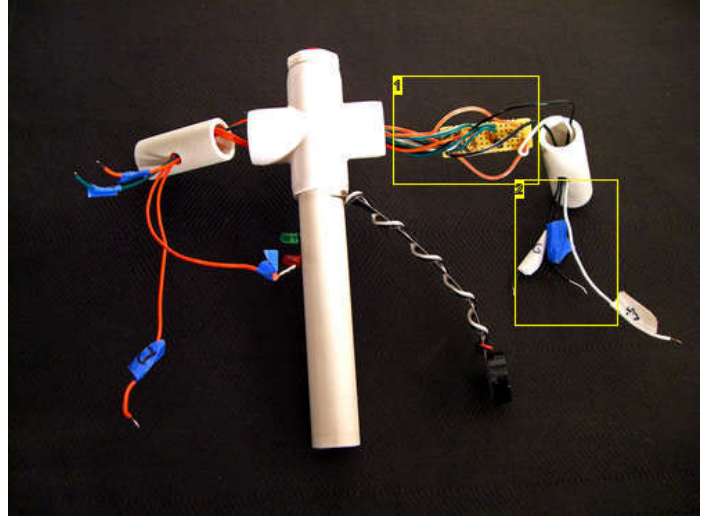


Image Notes

1. Circuit board getting ready to be crammed into the Board Arm.
2. The wires for Analog 0, Ground and +5v threaded through the hole in the Board Arm.

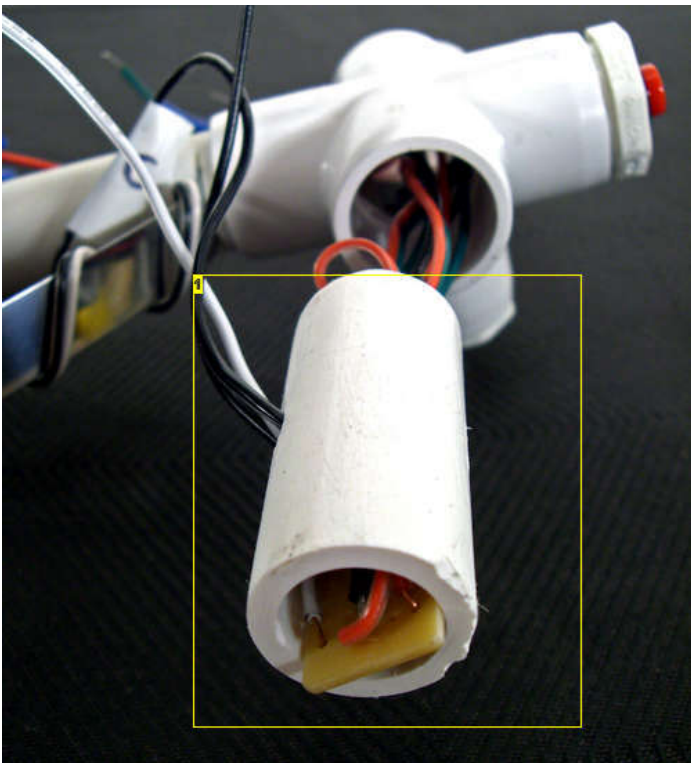


Image Notes

1. The circuit board has been pushed into the Board Arm. Having a little sticking out the far side is fine, it'll be hidden in the right angle adapter.

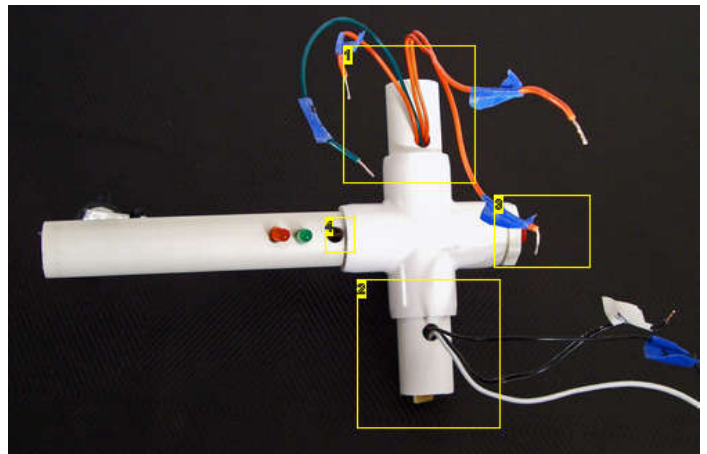


Image Notes

1. Short Arm plugged in.
2. Board Arm plugged in.
3. Button plugged in.
4. Once again, ignore this hole.

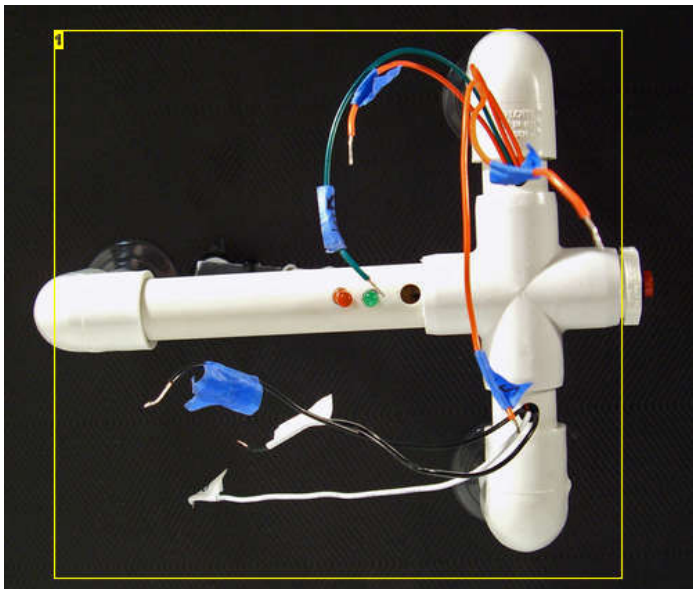


Image Notes

1. Legs and suction cups in place. at the end of the arms.

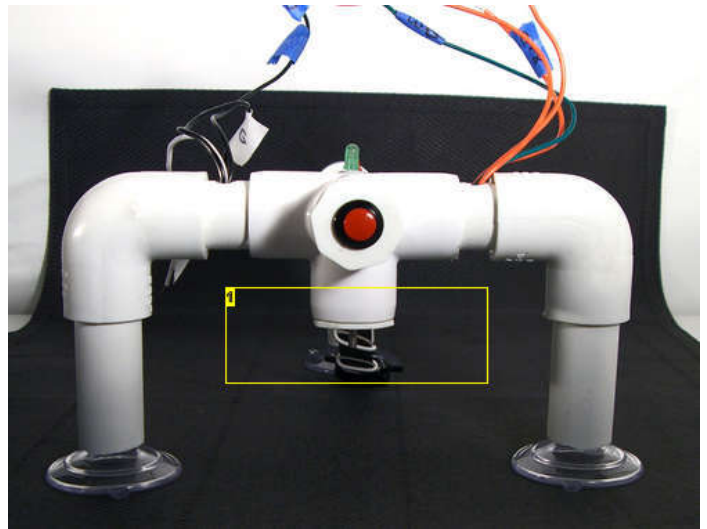


Image Notes

1. Side view. We haven't attached the lock clamp yet.

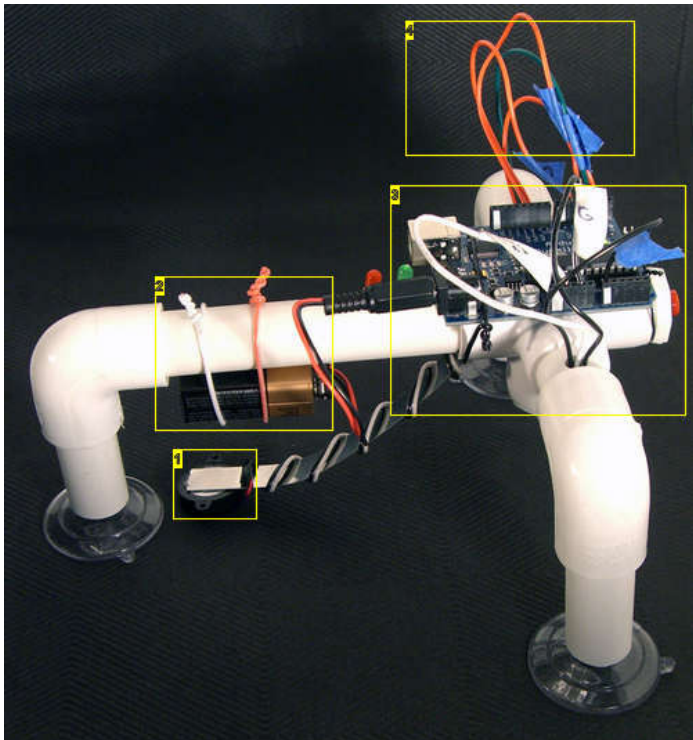


Image Notes

1. The knock sensor is arranged so it will press flatly against the door.
2. Battery attached.
3. Arduino attached and wires plugged.
4. Here is where a neat person would shorten the leads.

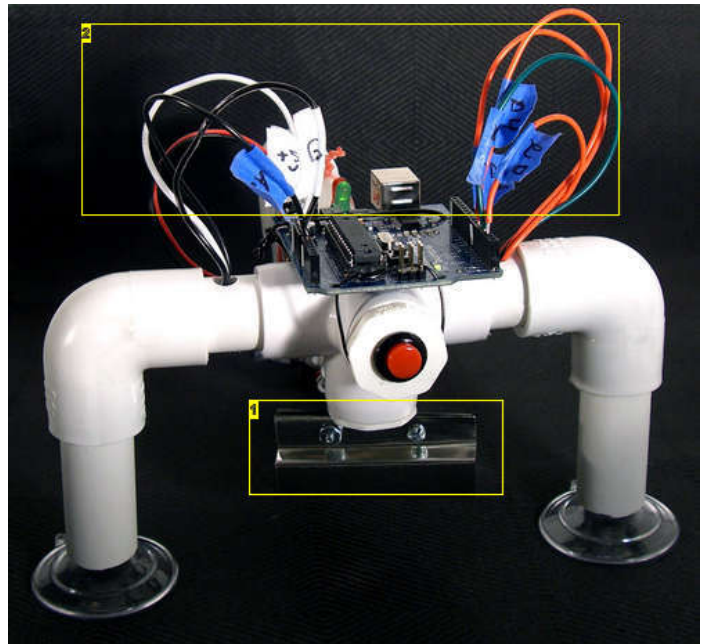


Image Notes

1. Side view. We've attached the lock clamp. Now we're ready to test it on the door!
2. Shortening the wires and removing the labels would make it look nicer, but you'll have to do that on your own time. I have an Instructable to finish!

step 9: Mounting, Testing, and Use

First lets make sure our legs are the right length. Be sure the lock clamp is attached to your motor and the legs are in place. Attempt to fasten it to the door so the clamp fits over your lock. If you're lucky the legs will be the right length and you're ready to go! If you're unlucky the legs are too short and you'll have to cut new ones. But chances are your legs will be too long. Either grind/file/sand/cut them to the right length so the clamp fits right over the deadbolt lock handle.

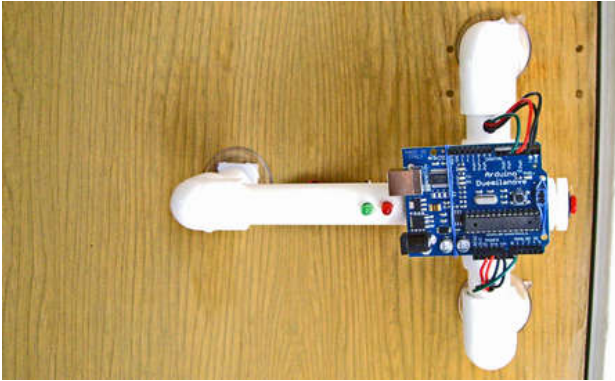
You don't have to use suction cups and in fact many doors are immune to their sucking. Two other options are to put PVC end caps on the legs and then secure them to the door with double sided foam tape (like this) or with screws through the caps, if you don't mind putting holes in your door.

Now that's its on you can give it a test. Do the first test from the inside. Lock your door and power it up. When the green light is on give it the old *Shave and a Haircut* and it should unlock!

Now program in a less obvious knock (or not) and your tree house will finally be safe from that smelly kid!

If the motor doesn't turn far enough to unlock your door you'll need to update the sketch to run the motor longer. (See Step 2!)

For additional Troubleshooting: See the bottom of **Step 2**. It also includes a bunch of other tweaks that might help you.



step 10: Epilog: Changes And Improvements

There are many ways to improve or change this project if you're feeling ambitious. Here are a few to get you started, feel free to add more in the comments.

- Add an H-Bridge to the circuit so it can lock and unlock the door.
- Make it work in silent mode by removing the knock sensor and attach a capacitance (touch) sensor to the doorknob and record sequences of touches.
- Use a servo to unlock the door rather than this hacked together gear-motor+slip transmission.
- Add a potentiometer to adjust the knock sensitivity and other values.
- Build it into an actual door knocker.
- Use a more economical microcontroller and enable sleep mode for better battery life.
- Make the whole package small enough to fit inside the door.
- Store several knocks so several people can have their own 'private' knocks.
- Add a real-time clock and using different knocks for different days of the week.
- Add a knocker to provide feedback through the door. It could then offer a challenge-response security where the door starts a knock sequence and the user has to finish it correctly.
- Remove the knock sensor and record pushes of the doorbell or other hidden button.
- Remove the knock sensor and put a photosensor in the peephole, send the open code through the peephole with a keychain flashlight.

And here's a zero-technology solution to the "*Yeah, but someone'll overhear your secret knocks!*" problem: Scream while knocking. No one will overhear the knock over the racket you're making.

Did you build this?

Post a photo (or better yet a video!) photo of it mounted on a door will earn a Master Of Secret Knocks patch*!

*As long as I have patches left to hand out. Which I do.

Masters of Secret Knocks:

- josiasfilho added a servo and locking ability.
- Jorad unlatches his door and added feedback in the peephole.
- Crimson-Deity added a pushbutton.



Image Notes

1. Post a picture (or video!) of your secret knock detector in comments and get this fancy Master of Secret Knocks patch

Related Instructables



How to Break Down a Door by drums787



Nintendo Keyless Entry System by action_owl



A bit of safe cracking... by killerjackalope



Arduino RFID Door Lock by pcmofo



Knock Block by jkestner



How to make Keychain Cutaway Padlock by amaze

Advertisements

Comments

50 comments

[Add Comment](#)

[view all 163 comments](#)



josiasfilho says:

Hi Steve!

As you know, I already started to do mine, with a few modifications. I uploaded a new video with a new option:

Dec 10, 2009. 3:39 AM [REPLY](#)

As soon as I finish my project I'll post the code online



Grathio says:

Fantastic work with the servo and the unlock option. (And an off switch!)

It looks like your setup is really sensitive. Really nicely done! Your achievement patch is on the way!

Dec 10, 2009. 12:53 PM [REPLY](#)



Jorad says:

I have been playing around with this code for the past hour or so and found out how you can hard code in your own rhythm if need be.... at the part that looks like this

// If we're recording a new knock, save the info and get out of here.

```
if (digitalRead(programSwitch)==HIGH){  
  for (i=0;i<maximumKnocks;i++){ // normalize the times  
    secretCode[i]= map(knockReadings[i],0, maxKnockInterval, 0, 100);
```

add this little guy

```
  Serial.println(secretCode[i]);
```

and open up your serial monitor....

make sure you un-commented the serial begin up at the setup and then it should spit some numbers back at you... put these numbers into the array and enjoy.

Jan 16, 2010. 6:16 AM [REPLY](#)



Jorad says:

Jan 23, 2010. 4:16 PM [REPLY](#)

Whoot finally got my relays for my motor and heres a neat little Idea for your visual feedback....



Jorad says:

Jan 23, 2010. 4:17 PM [REPLY](#)



Grathio says:

This is awesome! Thanks for sharing your code and video. Having the door unlatch too makes it that much more mysterious. And I like the feedback through the peephole.

Jan 24, 2010. 12:24 PM [REPLY](#)

Great work!

And if I ever have any questions about taping something to something else I'll be asking you. :)



Grathio says:

Oh, and for your awesome work you have been added as an official Master of Secret Knocks. Your patch is on its way!

Jan 24, 2010. 12:38 PM [REPLY](#)



berr08 says:

Did you finish yet? Can't wait to see the code.

Jan 28, 2010. 11:42 AM [REPLY](#)



Grathio says:

I really like your take on it, thanks for sharing! Your patch is on the way!

Sep 7, 2010. 3:23 PM [REPLY](#)

(And here's the video embedded for everyone else to see.)



Dr_Stupid says:

Finally SOMEONE posts a really interesting and USEFUL Instructable. +1

Sep 16, 2010. 10:19 AM [REPLY](#)



Crimson-Deity says:

Ok, I've got it up and running ! All I need is some cable sorting and I'm ready to show the world! Do you have any of those patches left? :D

Aug 30, 2010. 12:28 PM [REPLY](#)



Grathio says:

Fantastic! Any chance you can post a photo or (even better) a video?

Aug 31, 2010. 11:09 AM [REPLY](#)



Crimson-Deity says:

Ok! Its nearly finished! <http://www.youtube.com/watch?v=V6X6jEcNDY8> You can see my current status there ! :D

Sep 5, 2010. 10:21 AM [REPLY](#)




Crimson-Deity says:

I'll upload a video as soon as I'm completely finished ! I also have to find a Wall to USB adapter since a 9volt wall-adapter makes my green indication light blink for some reason :S


Aug 31, 2010. 1:55 PM [REPLY](#)


 **hjuyhgfhjh** says: Aug 31, 2010. 12:29 AM [REPLY](#)
Can I buy this because im not very good with electronics. im UK based


 **godgodgodgo** says: Aug 13, 2010. 12:53 PM [REPLY](#)
How much did this cost to make?


 **Grathio** says: Aug 14, 2010. 12:25 PM [REPLY](#)
I answered this one below, but since there are a lot of comments to read through I'll answer it again:
It cost me hardly anything since it was all stuff I had lying around. To buy all the parts as listed is around \$80 USD. Most of that price is from the Arduino (\$30) and the motor (Around \$15). You can bring that down if you use an Arduino clone like the Boarduino (\$17) or Diavolino (\$13) (But you will need a USB->TTL adapter to program these boards.)
You can also replace the motor with a high torque servo to save a few dollars but you'll have to modify the program and mounting to work with a servo.


 **Chewie42** says: Aug 4, 2010. 7:46 PM [REPLY](#)
totally awesome...would be great for a secret panel somewhere


 **Danne11** says: Jul 18, 2010. 11:05 AM [REPLY](#)
the engine starts when I plug it into the Arduino: (has anyone any idea what I'm doing wrong?


 **chestercito** says: Jul 8, 2010. 12:35 PM [REPLY](#)
hey man..I want to use this but for an electric lock...what do you think i would need no change in the code or in the circuit??


 **B.F.L.M** says: Jun 30, 2010. 12:40 PM [REPLY](#)
it could be more secure if you use layered security (e.g. RFID, knocking, PIN, iris scanner, fingerprint, AND key!!! (in a specific order of course)) p.s. im a little paranoid


 **kriscristian** says: Jun 28, 2010. 4:56 AM [REPLY](#)
pls help.. can I use an atmel 8051 micro?? how would the programming be??


 **nmarbm** says: Jun 22, 2010. 11:38 AM [REPLY](#)
Where do you draw the sketch, the colorful one (with arduino and protoboard)?


 **Grathio** says: Jun 22, 2010. 12:23 PM [REPLY](#)
It was done with the free application Fritzling.
<http://fritzing.org/>
It's super easy to use and can also generate schematics and even make (simple) PCB layouts for manufacturing.


 **nmarbm** says: Jun 22, 2010. 2:25 PM [REPLY](#)
Thank you vey much. And when I am on this page I could say it is a very good project, and I build one yesterday (but only on protoboard) and it works perfectly. And because I am very intrested in arduino, I would like to ask you would you post more new arduino projects?


 **iceng** says: Jun 20, 2010. 11:27 PM [REPLY](#)
I used an accelerometer in my e-lock, this responded to drumming fingertips and made it look as if I was bored as I leaned with my back to the secret door while my hand was sending a complex tap code. This works on a car if you include a temperature adjustment for cold weather. iceng...


 **iceng** says: Jun 20, 2010. 11:14 PM [REPLY](#)
If you are using the 2N2222 transistor (NPN), the Emitter (arrow) should be the pin wired to ground and the Collector wired to the motor ! It may work backwards as the schematic shows, but with very very poor gain. Change this for a better motor drive. Good Luck... iceng...


 **studystudyforever** says: Jun 14, 2010. 7:51 PM [REPLY](#)
Storming idea, but i don't know what happen when AC power cutting which usually done in my country?


 **juanvi** says: Jun 14, 2010. 5:11 PM [REPLY](#)
woha! very nice and clever!! i really like it, but have you ever used it in real life? as going out without keys and returning 5 hours later and be opened by it?? how much does the battery last??


 **gallovdaniel** says: Jun 14, 2010. 12:43 PM [REPLY](#)
Very nice. I thought that it was for opening any doors. Even though, Very clever invention!!!


 **alcurb** says: Jun 10, 2010. 8:05 AM [REPLY](#)
Very clever invention. For increased security, may I suggest to add a touch sensor option connected to door handle/knob. Lightly tap your code on the door handle/knob. The neighbors won't be able to hear that.


 **eng.7oka** says: Jun 5, 2010. 8:15 AM [REPLY](#)
hey.. sorry to bother you again.. I just want to know what the function (map(knockReadings[i],0, maxKnockInterval, 0, 100)); does..


 **Grathio** says: Jun 5, 2010. 10:15 AM [REPLY](#)
Arduino reference for the map() function.
This normalizes the data so the longest time between knocks gets "100" and the rest are adjusted proportionately. We do this so it doesn't matter if you knock fast or slow as long as you get the rhythm correct.


 **eng.7oka** says: Jun 5, 2010. 4:39 PM [REPLY](#)
I read that in the comment.. but the thing is I'm programming using another language.. so I need to understand the mathematical algorithm behind it.


 **Grathio** says: Jun 5, 2010. 7:11 PM [REPLY](#)
The best description of what it does is in the link I provided to the description of the function in the Arduino Reference manual.

 **godgodgodgo** says: Jun 4, 2010. 3:51 AM [REPLY](#)
Do I need a power supply for the Arduino Duemilanove and the circuit board?


 **Grathio** says: Jun 4, 2010. 11:31 AM [REPLY](#)
Reread the 'Materials' section carefully and you can see my recommendations for powering the project.


 **godgodgodgo** says: Jun 4, 2010. 3:40 AM [REPLY](#)
Why when you etched the board out did you not just leave a long line of the copper instead of the wires for the power. Also should you not have done it so that some of the holes are linked?

 **Grathio** says: Jun 4, 2010. 11:23 AM [REPLY](#)
The board is a perfboard, not a custom etched board. Follow the link in the 'Materials' section to see what I'm talking about.

 **nitetime3021** says: Jan 29, 2010. 2:54 PM [REPLY](#)
Could this be modified to work with music. For example, tones played in a certain order unlocks the lock?

 **11ericg** says: May 29, 2010. 5:45 PM [REPLY](#)
yes

 **Grathio** says: Jan 30, 2010. 12:32 PM [REPLY](#)
Unfortunately recognizing tones is a different task from rhythm. The piezo element would need to be replaced with a microphone and amplifier. The code would need some excellent noise removal and pitch recognition, probably some kind of FFT, which is difficult (though not impossible!) to do on the Arduino's limited hardware.
By the time you do all that it would essentially be a different project. A **very cool** project, but a different one.

 **Robot Lover** says: May 23, 2010. 5:35 PM [REPLY](#)
When I download the file and open it in the arduino sandbox it says "Bad file: Processing can only open sketches and other files ending in .pde" Help anyone?



Grathio says:

Sometimes the file that Instructables gives you is a different file name. Just rename it to secret_knock_detector.pde and it should work fine.

May 24, 2010. 12:30 PM [REPLY](#)



Robot Lover says:

I already did but it still does not work. Any suggestions?

May 24, 2010. 3:43 PM [REPLY](#)



Grathio says:

Hm. I'm guessing some computer somewhere is messing with the encoding.

You can also download it directly from my site [here](#). If both of those links don't work for you try a different web browser to download the files.

May 25, 2010. 9:32 AM [REPLY](#)



louwhopley says:

Great project!
Haven't read the instructable, but the video looks **really good** !

May 16, 2010. 12:02 AM [REPLY](#)



eng.7oka says:

Ok..I Think I will use C ... I'm looking for C compiler for my PIC now... Thanks again

May 13, 2010. 1:45 AM [REPLY](#)



darthony says:

Is it possible each time you program a new knock code(on the fly , using the program button) the arduino stores it in the EEPROM so you dont have to re program it when you reset?
can someone help me with this?

May 11, 2010. 8:46 AM [REPLY](#)



Grathio says:

Yup, it is possible, and a good feature. Here's an outline of the changes you'll need to make:

- in Setup() you'll need to read the data from eeprom. (If there is any.)
- in the validateKnock() function (around line 208) you'll need to write the info to eeprom. You could do it as part of this code loop:

```
for (i=0;i<maximumKnocks;i++){ // normalize the times  secretCode[i]= map(knockReadings[i],0, maxKnockInterval, 0, 100); // Write the \
```

Here are examples of how to read and write EEPROM.

You also might want to look at the code for my [Rock-Paper-Scissor Playing Glove](#) for an example that saves and reads info to/from the Arduino EEPROM.

[view all 163 comments](#)