You can watch a video of it here:
http://www.youtube.com/watch?v=8BghleelvtY

Welcome to a step-by-step guide in building a "Barking Desk Guard Dog" toy using a stuffed dog toy integrated with the Seeedstudio Grove Toy Bundle!

The Barking Desk Guard Dog will bark at anyone who will try to move the object in front of it. You can place it on your desktop when you are not around and when someone comes near your place (maybe in your work) then it will alert all your colleagues when that someone tries to take something from your desk.

As we would want to stay consistent with this guide, instead of listing out everything, we will be enumerating the materials needed as we go along – for you to be able to be allowed to come up with other neat ideas of your own! After all you can hack this toy hack too.

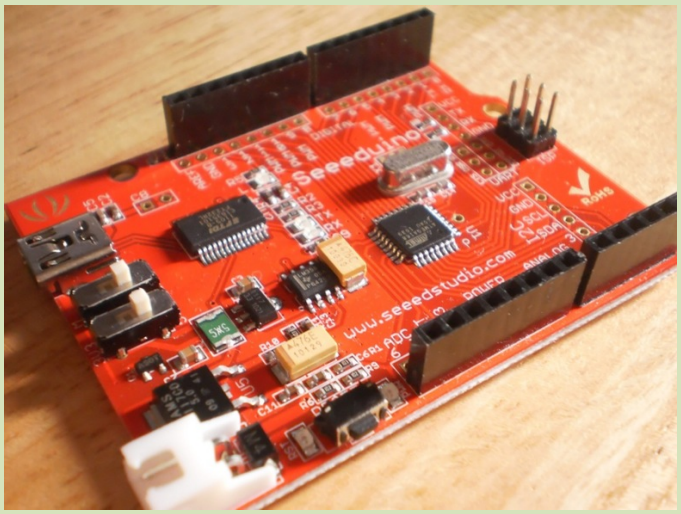
Also, we would want to be minimal with our toy, and you wont have to worry about other components (except for an LED, some wires and our controller board) because we would be using the stuff that we would get out of the box of the **Grove Toy Bundle Kit!**

Prerequisites:

Basic C programming skills
Familiar with the Arduino IDE
Familiar with the GROVE system
Very basic electronics
Very basic sewing skills
Basic soldering

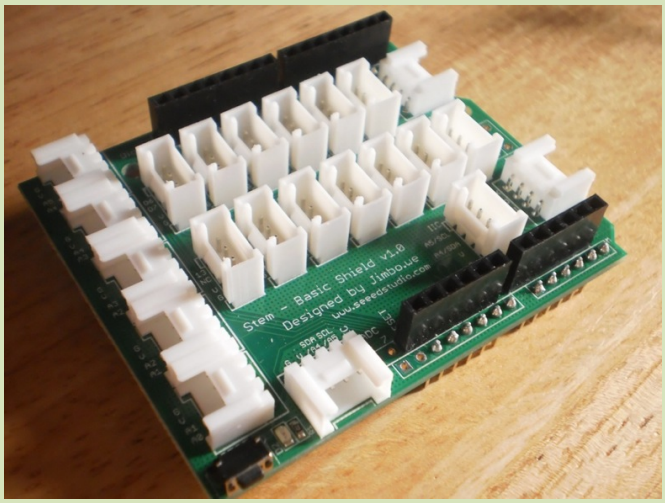
Above are the skill sets that will help us with this project. Do not worry if you are not into some of these because by the end of this guide, you will have experience all of it.

Project Base



Materials:
1 Seeeduino

We will start with our Seeeduino v2.21 board. The one pictured here is the ATMega168 version, but it would be enough for our toy to work. You can also use other Arduino compatible boards but if you would choose to, you may expect the code from this guide to look different and may not work with your other controller board.

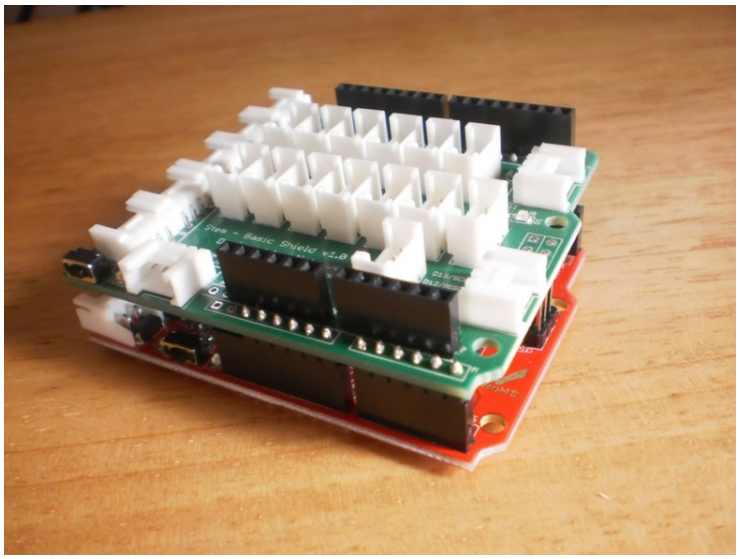


Materials:
1 Stem Basic Shield

Then we would need the Stem Basic Shield for us to attach our Twig Components.

This is important to take note: we cannot use 2 adjacent ports for 2 sensors, for example attaching a Twig Sound Recorder to port 2 and then attaching a Twig Variable Color LED to port 3 as it will interfere with each other. This guide has taken that step to make sure no sensors are competing with pins. I hope this will answer your questions later about “why did we choose that port/pin?”

Let us now start!



Stack up the Stem Basic Shield on top of the Seeeduino. Make sure that it is fit snugly and rests flat with the Seeeduino.

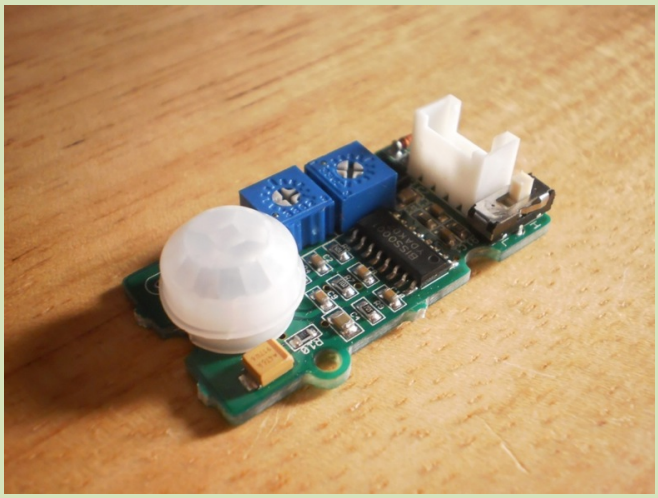
Note:
For most of the pictures, this guide does not show the Seeeduino connected to the USB port. This is to avoid confusion with the connections we will see later.

Checkpoint!

Just to test out, launch Arduino and run the Blink sample program just to check if everything is working. You can even place an LED on PIN 13 and GND of the Stem Base Shield.

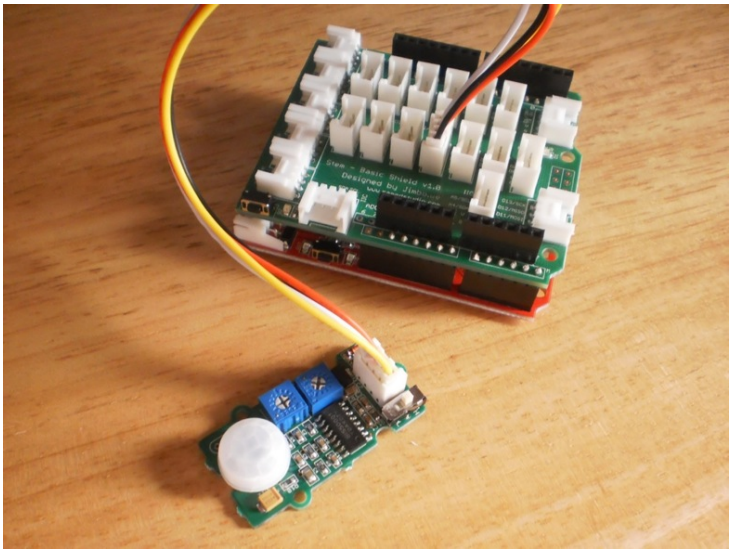
You will see a lot of these "Checkpoints" as we go along to make sure that things are working

Motion Sensor

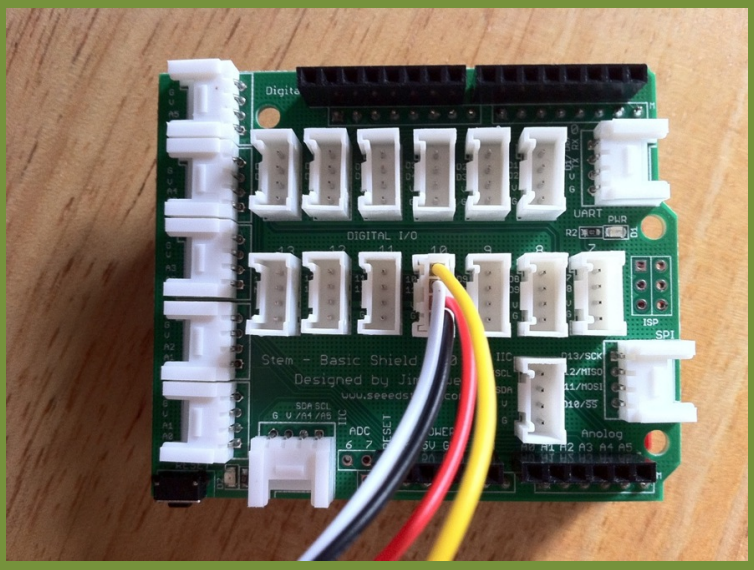


Materials:
1 Twig PIR Motion Sensor
1 Grove Universal 4 Pin Cable

First, we will make our motion sensor work. Why are we starting with it? Because it will be the main trigger for the entire system. Our goal is to test out that when there is movement, then an LED will light up.



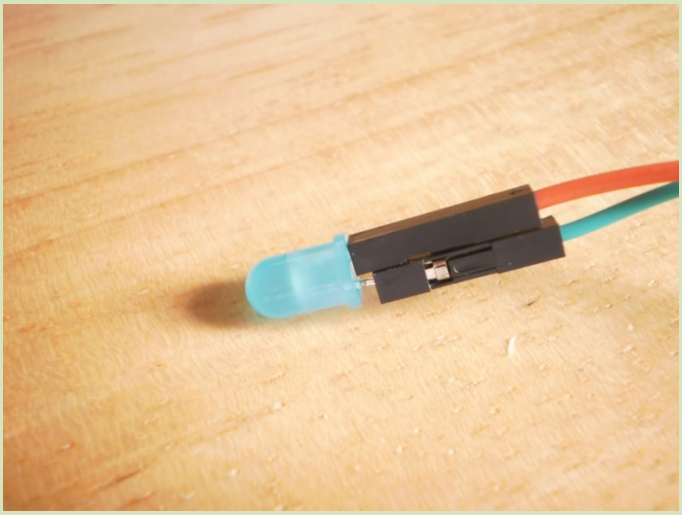
Connect the Twig PIR Motion Sensor to port 10 of the Stem Base Shield.



Checkpoint!

Take a look at the connections and compare it with the one on the right. Are they correct?

LED Indicator

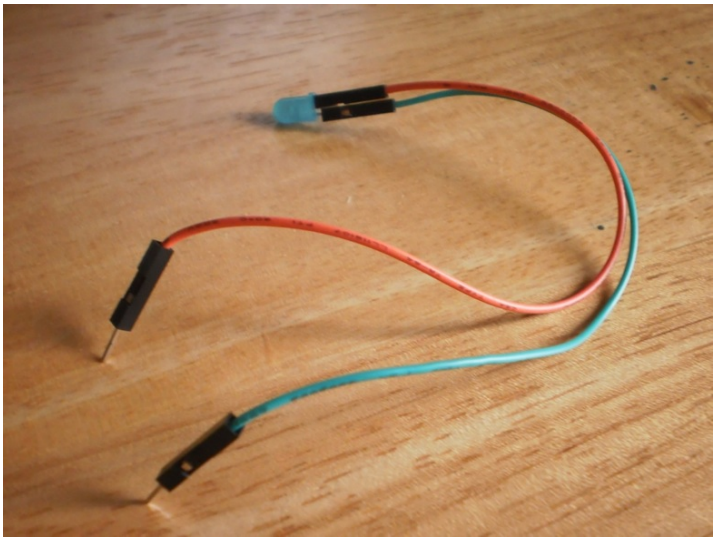


Materials:

1 LED

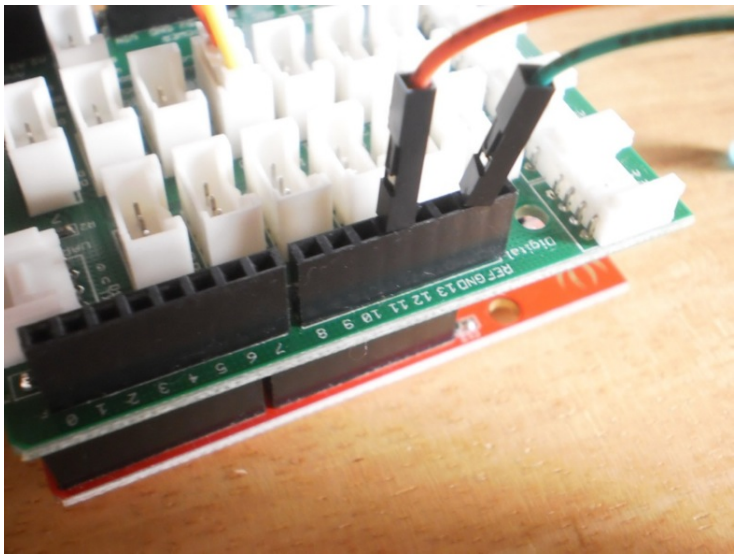
2 Hookup wires (or any wire)

Connect the 2 hookup wires to an LED.



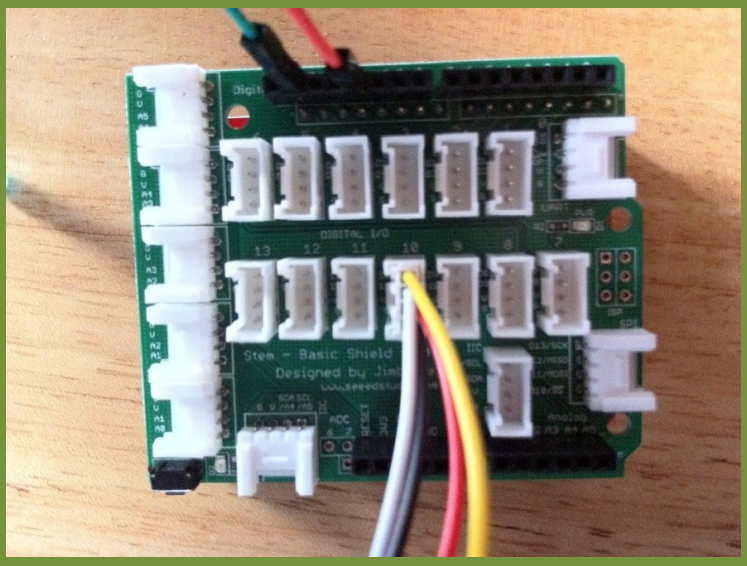
The hookup wires shown here are 1 pin female jumper wires. But if you don't have one, breadboard jumper wires are good but you will have to solder on to the leads of the LED. Color-coded also here are RED for the input voltage (or signal) and GREEN for ground.

*If you would like to know a shortcut on how to determine which is ground: Look closely inside a conventional LED, the **bigger electrode** inside connects to the ground pin.*



Then connect the ground wire (GREEN) to the GND pin and the input voltage (RED) to pin 11 along the **digital** pins of the Stem Base Shield.

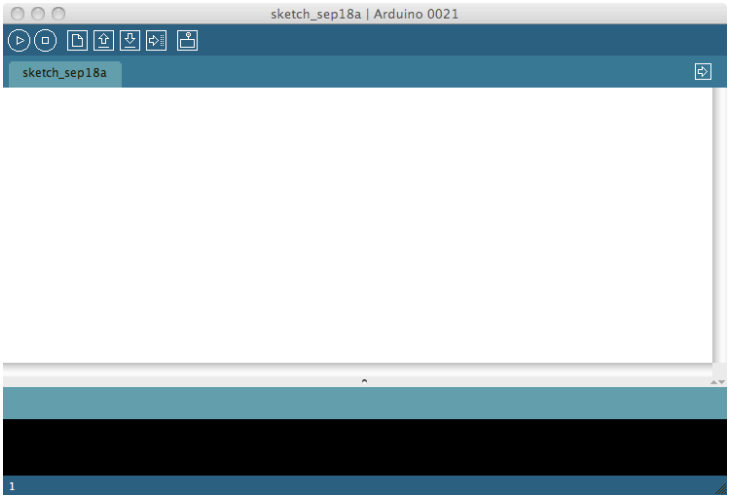
Why pin 11? Because from the Grove cable, the Motion Sensor uses both pins 10 and 11, but it says that pin 11 is not used. So we are now using this spare pin as an indicator for now.



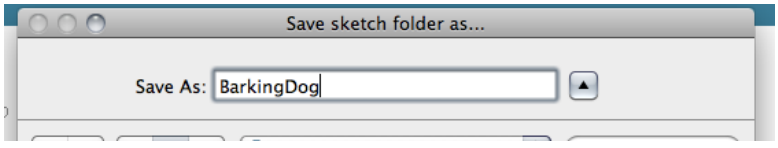
Checkpoint!

Take a look at the connections and compare it with the one on the right. Are they correct? Ensure also that the LED connections are tight and does not plug off easily. This is essential when we stuff everything inside the stuffed toy.

Arduino



Now on to the code! Let us open our Arduino IDE and then we will expect a new code window to be opened.



Let us save our sketch as “BarkingDog”

Let us test out that when there is movement, then an LED will light up.

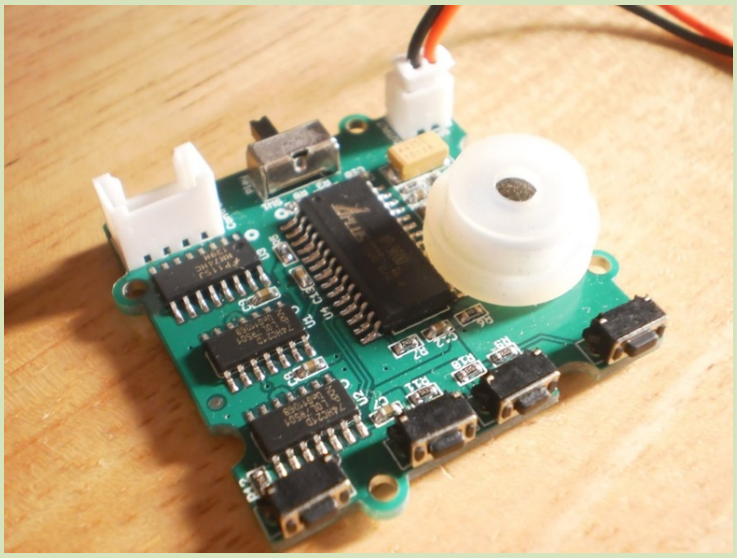
Our code will look like below:
Code available online at:
https://github.com/wendellinfinity/GroveBarkingDog/blob/master/_01_Motion_Sensor/_01_Motion_Sensor.pde

```
#define PIR 10          // pin 10 for PIR signal
#define PIRIND 11       // recycle N/C pin from PIR to led test indicator

// keep states
boolean isMoveDetected; // if PIR triggered

void setup() {
     // initialize motion sensor pin
     isMoveDetected = false;
     pinMode(PIR, INPUT);
     pinMode(PIRIND, OUTPUT);
}

void loop() {
     if(digitalRead(PIR) && !isMoveDetected) {
          digitalWrite(PIRIND,HIGH);
          if(!isMoveDetected) {
               isMoveDetected = true;
          }
     }
}
```

Try to load this up to our Seeeduino and look at how it works.
Does the LED light up when it detects your movement?

Take time also to adjust the sensitivity of the PIR sensor at this point.

Sound Recorder

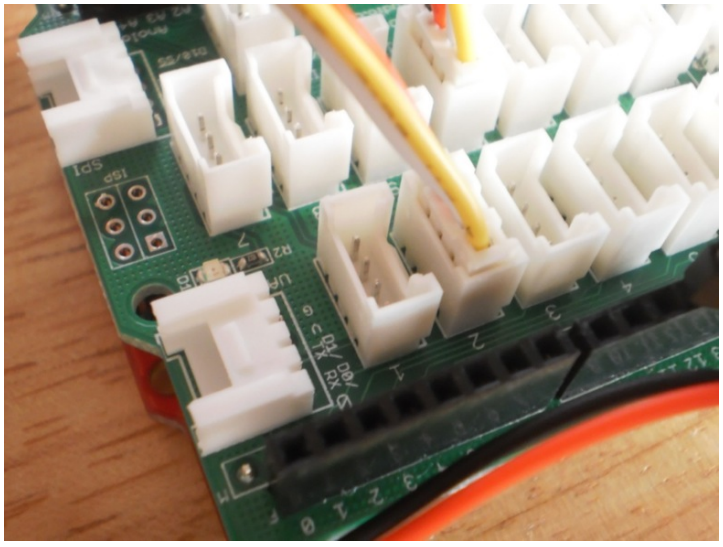


Materials:
1 Twig Sound Recorder
1 Grove Universal 4 Pin Cable

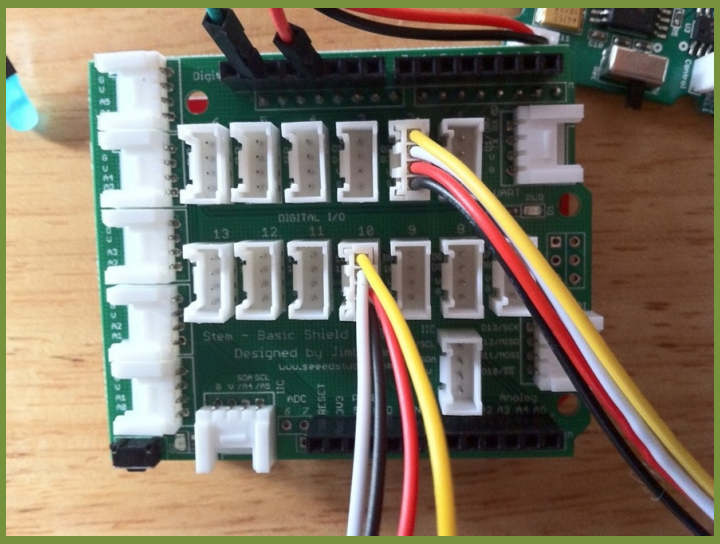
We will try out our sound recorder and integrate it with the motion sensor. When we detect movement, then the sound recorder will play it's #2 track.

This is the time to go out too and use the sound recorder to record a real barking dog! To do this, we can follow Seeedstudio's wiki here:

How to control it manually
http://seeedstudio.com/wiki/index.php?title=Twig_-_Sound_Recorder_v0.92b#How_to_controll_it_manually



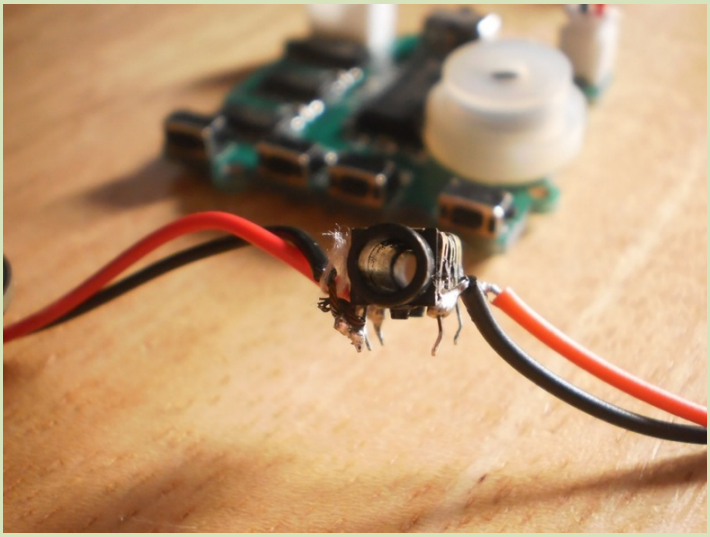
Connect the Sound Recorder to port 2 of the Stem Base Shield.



Checkpoint!

Take a look at the connections and compare it with the one on the right. Are they correct?

External Speakers

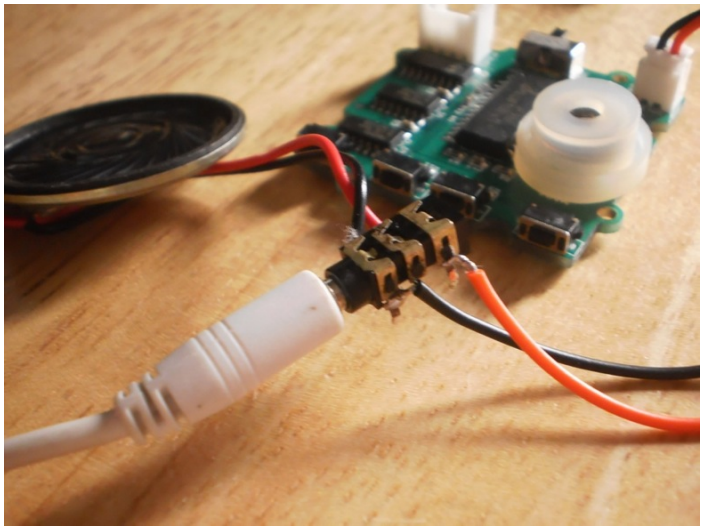


Careful! Ask for help or let someone teach you if you are not familiar with soldering.

Materials:

1 Audio jack
Wire cutters
Soldering Iron
Soldering lead
Audio jack extension cable
Any external speaker connection

You might now be satisfied with how the speaker sounds because it may not be that loud. We would want to extend the dog's barking volume to external speakers, which will really alert those around. Then we can hide this deep within the dog so they cannot plug it out.



The audio jack shown here is the stereo jack extension cable attached, but the output will only be mono from the sound recorder. What we can do is to connect just only 1 (the farthest terminal) to the + pin of the sound recorder and then the – pin will be connected to the nearest terminal of the plug.

Trivia: In audio jack terms, we call it the TIP, RING, and SLEEVE. In the male stereo jack, the TIP is the pointy end and the ring is the middle one. The sleeve is the one closest to the wire. In our case, we will connect the sound recorder's + to the TIP and the – to the SLEEVE. We can leave the RING not connected for now.

Let us go back to our Arduino IDE and test our code. We will make the sound recorder play sounds when the motion sensor is triggered. We will use one of our pre-made libraries to make the sound recorder work. Let us check a separate guide on how it is done here:
http://wendellinfinity.wordpress.com/2011/08/29/grove-sound-recorder-library/

Our code will now look like below:
Code available online at:
https://github.com/wendellinfinity/GroveBarkingDog/blob/master/_02_Sound_Recorder/_02_Sound_Recorder.pde

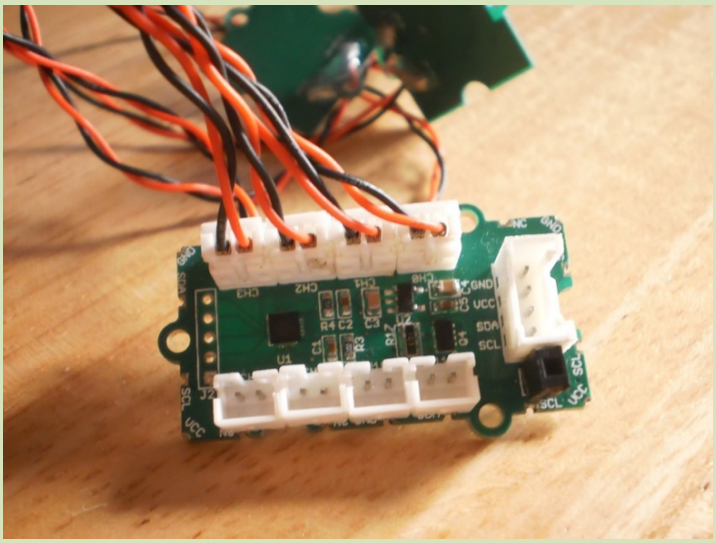
```
#include <GroveSoundRecorder.h> // include our sound recorder library

// define sensor pins
#define SOUNDRECSEL1 2 // Grove pin 2 for sound recorder
#define PIR 10         // pin 10 for PIR signal
#define PIRIND 11      // recycle N/C pin from PIR to led test indicator

// initialize a recorder
GroveSoundRecorder recorder(SOUNDRECSEL1);

// keep states
boolean isMoveDetected; // if PIR triggered

void setup() {
     // initialize the sound recorder
     recorder.initialize();
     // initialize motion sensor pin
     isMoveDetected = false;
     pinMode(PIR, INPUT);
     pinMode(PIRIND, OUTPUT);
}

void loop() {
     if(digitalRead(PIR) && !isMoveDetected) {
          //Serial.println("Something moved");
          digitalWrite(PIRIND,HIGH);
          if(!isMoveDetected) {
               recorder.beginPlaybackLoop(TRACK2);
```

```
                isMoveDetected = true;
            }
        }
}
```

Try to load this up to our Seeeduino and look at how it works.

Do the LED light up and the sound recorder play TRACK 2 when it detects your movement?

Touch Sensors

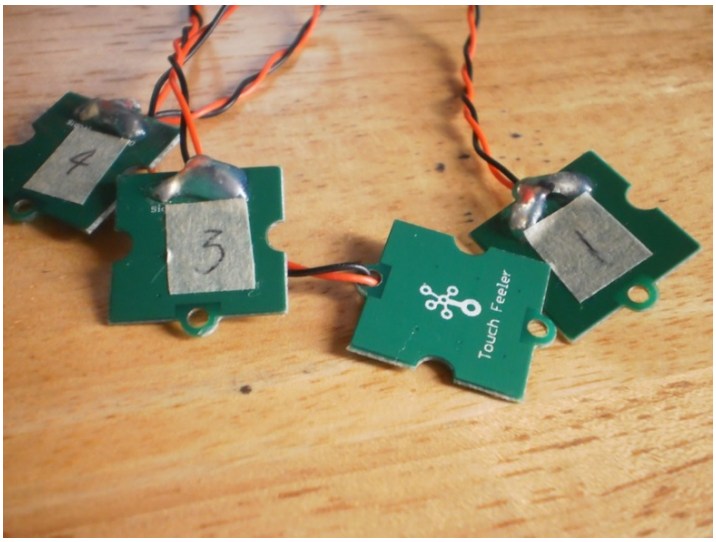


Materials:

1 Twig I2C Touch Sensor
1 Grove Universal 4 Pin Cable
Masking tape
Marker pen

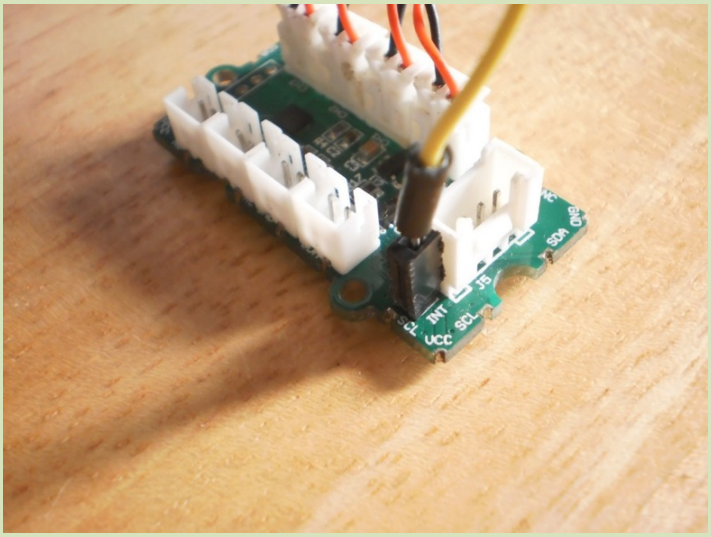
We will now use the touch sensor to disable the alarm with a series of touch combinations. This sensor is most challenging part of this guide because we will be doing a 3 digit password challenge to turn off the barking after the motion sensor is triggered.

We only have 4 touch feelers and we would be reserving 1 for an ENTER CODE key.



Matching the ports on the sensor, place a label on each of the touch feelers:

1 for CH0
2 for CH1
3 for CH2
4 or ENTER for CH3

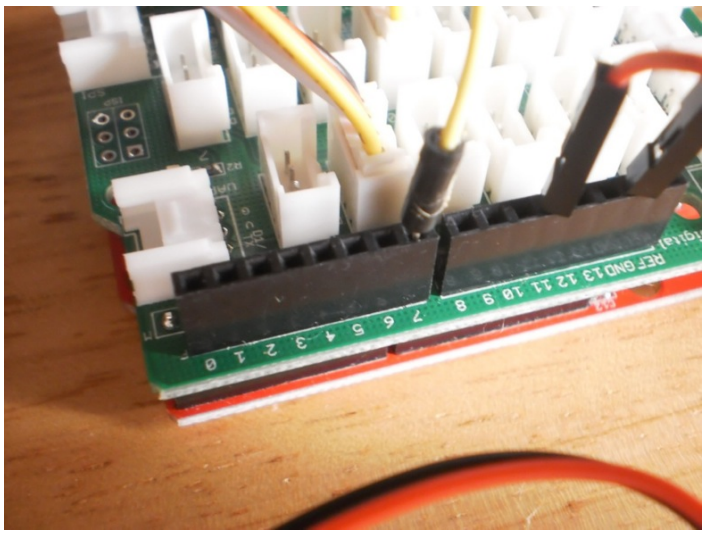


Careful! Ask for help or let someone teach you if you are not familiar with soldering.

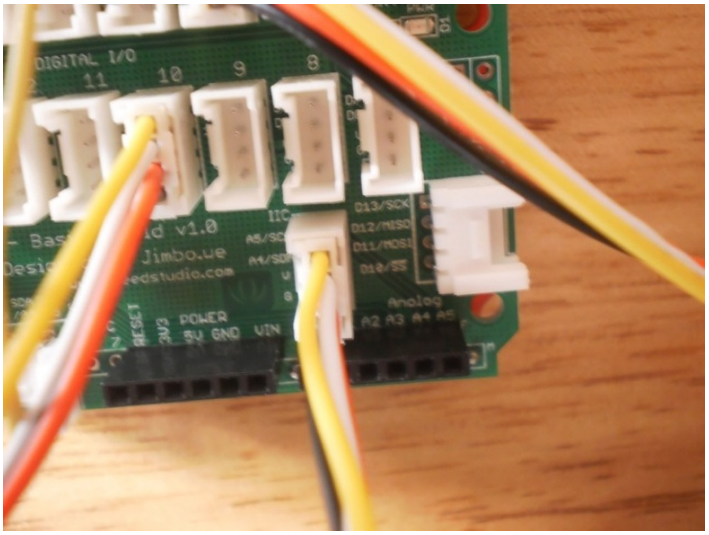
Materials:

1 female pin header (or hookup wire)
Soldering iron
Soldering lead

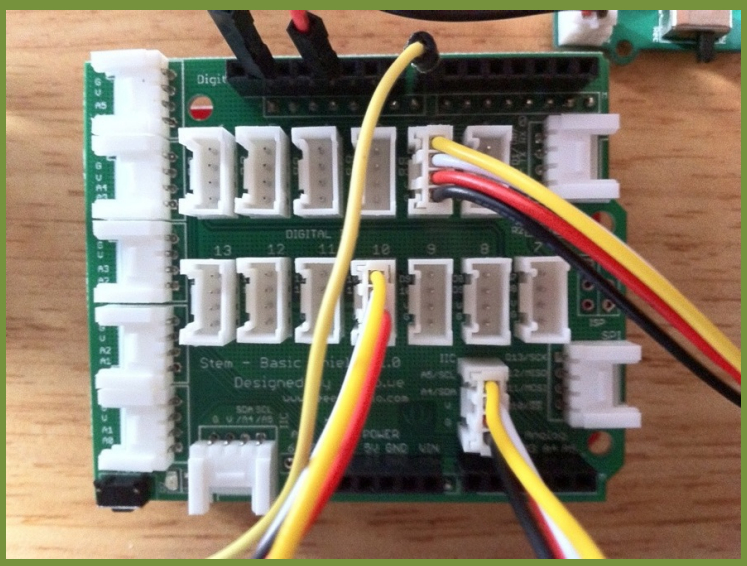
To make the sensor work, we will need to solder out a female pin header to the INT pin on the I2C Touch Sensor board. This is a required step because we will save a ton of code in doing so. If you don't have a female pin header then you can solder on a hookup wire onto it.



Connect the INT pin from the touch sensor to the PIN 7 of the Stem Base Shield.



Then connect the grove cable to any of the 2 I2C ports in the Stem Base Shield.



Checkpoint!

Take a look at the connections and compare it with the one on the right. Are they correct? Take note of the INT pin connection, it must be tight and does not plug off easily on both ends (sensor and the base).

We will now make the barking stop by using a 3 digit code combination. We will again use one of our pre-made libraries to make the Twig I2C Touch Sensor work. Let us check a separate guide on how it is done here: http://wendellinfinity.wordpress.com/2011/08/27/grove-i2c-touch-sensor-library/

Our code will look like below:
Code available online at:
https://github.com/wendellinfinity/GroveBarkingDog/blob/master/_03_Touch_Sensors/_03_Touch_Sensors.pde

```
#include <Wire.h> // include I2C library
#include <GroveMultiTouch.h> // include our Grove I2C touch sensor library
#include <GroveSoundRecorder.h> // include our sound recorder library

// define sensor pins
```

```
#define SOUNDRECSEL1 2 // Grove pin 2 for sound recorder
#define TOUCHINT 7     // arduino pin 7 for I2C touch interrupt
#define PIR 10         // pin 10 for PIR signal
#define PIRIND 11      // recycle N/C pin from PIR to led test indicator
#define PASSCODELEN 3  // length of passcode

// initialize a recorder
GroveSoundRecorder recorder(SOUNDRECSEL1);
// initialize the Grove I2C touch sensor
GroveMultiTouch feelers(TOUCHINT);
// keep track of 4 pads' states
boolean padTouched[4];

// keep states
boolean isMoveDetected; // if PIR triggered
boolean isCodeCorrect; // if code is good
byte inputcode[PASSCODELEN]; // user input code
byte password[PASSCODELEN] = { // the secret code is 3, 1, 2
     3,1,2};
int inputcounter;

void setup() {
     Wire.begin(); // needed by the GroveMultiTouch lib
     // initialize the containers
     for(int i=0; i<=3; i++) {
          padTouched[i]=false;
     }
     // initialize the touch sensors
     feelers.initialize();
     inputcounter=0;
     for(int i=0; i<PASSCODELEN; i++) {
          inputcode[i]=0;
     }
     isCodeCorrect=false;
     // initialize the sound recorder
     recorder.initialize();
     // initialize motion sensor pin
     isMoveDetected = false;
     pinMode(PIR, INPUT);
     pinMode(PIRIND, OUTPUT);
}

void loop() {
     if(digitalRead(PIR) && !isMoveDetected) {
          //Serial.println("Something moved");
          digitalWrite(PIRIND,HIGH);
          if(!isMoveDetected) {
               recorder.beginPlaybackLoop(TRACK2);
               isMoveDetected = true;
          }
     }
     if(isMoveDetected) {
          feelers.readTouchInputs(); // test read the touch sensors
          // loop through our touch sensors 1 to 4
          for(int i=0; i<=3; i++) {
               // get the touch state based on pin #
               if(feelers.getTouchState(i)) {
                    // flag the touch sensor state
                    padTouched[i]=true;
               }
               else {
                    if(padTouched[i]) {
                         if(inputcounter==PASSCODELEN) {
                              inputcounter=0;
                         }
```

```
                    switch(i) {
                    case 0:
                         inputcode[inputcounter]=1;
                         inputcounter++;
                         break;
                    case 1:
                         inputcode[inputcounter]=2;
                         inputcounter++;
                         break;
                    case 2:
                         inputcode[inputcounter]=3;
                         inputcounter++;
                         break;
                    case 3:
                         delay(500);
                         // check if input code is good
                         isCodeCorrect=true;
                         for(int i=0; i<PASSCODELEN; i++) {
                              if(inputcode[i]==password[i]) {
                                   isCodeCorrect=true;
                              }
                         }
                         // check if shutdown code is correct
                         if(isCodeCorrect) {
                              // turn off alarm
                              isMoveDetected = false;
                              digitalWrite(PIRIND,LOW);
                              recorder.stopPlayback();
                              delay(5000); // delay to settle down
                         }
                         // reset user input code
                         inputcounter=0;
                         for(int i=0; i<PASSCODELEN; i++) {
                              inputcode[i]=false;
                         }
                         break;
                    default:
                         break;
                    }
                    delay(300);
               }
               // reset the touch sensor state
               padTouched[i]=false;
          }
     }
}
```

Try to load this up to our Seeeduino and look at how it works.

Do the LED light up and the sound recorder play TRACK 2 when it detects your movement?

When we touch 1, 2, 3 and 4(enter), does the barking go on?

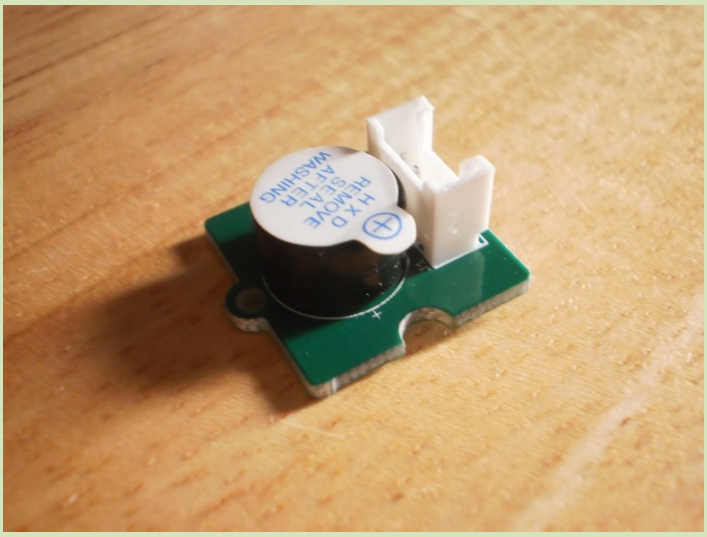
When we touch 3, 1, 2 and 4(enter), which is the correct code, does the barking stop?

When we touch 1 and 4(enter), does the barking go on?

When we touch continuously (more than 4 touches without 4(enter)), does the barking go on?

When we touch 4(enter) without any touches with the code numbers, does the barking go on?

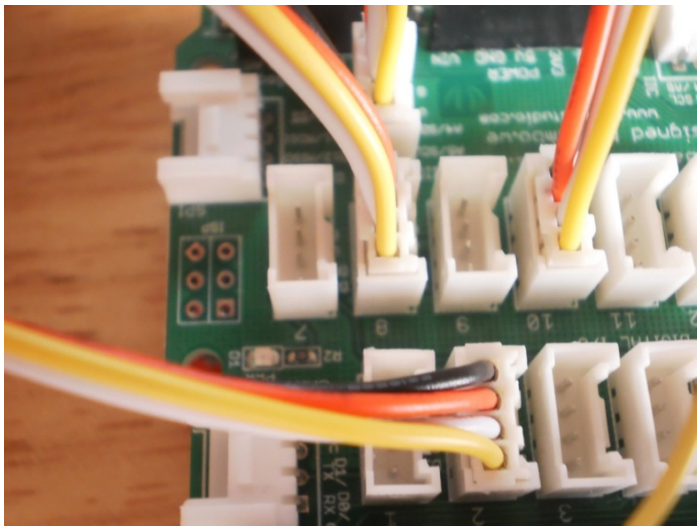
Buzzer



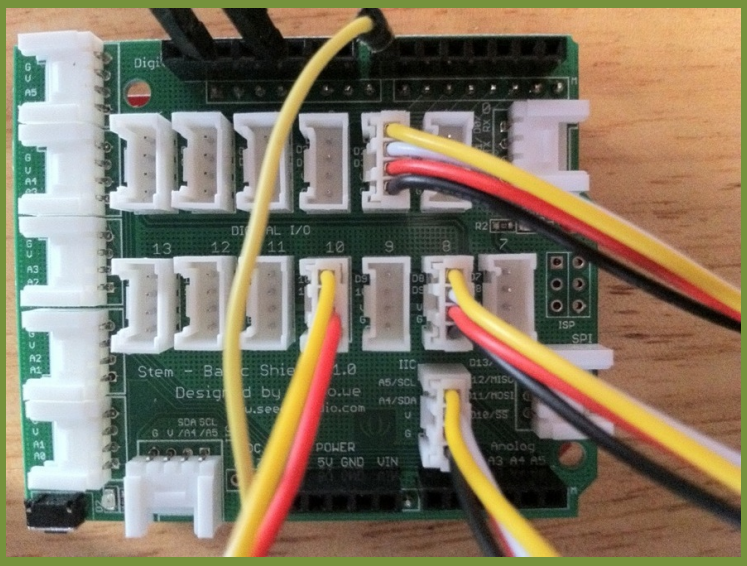
Materials:
1 Twig Buzzer
1 Grove Universal 4 Pin Cable

Now wouldn't it be fun if you would have feedback when touching the touch feelers? For this we will use the Twig Buzzer and make it beep while we are touching the feeler. We will make it stop beeping when we release our touch.

Also the buzzer will notify us if we would enter a wrong key combination with 2 long beeps. If we are correct then we get 3 short beeps.



Connect it to port 8 of the Stem Base Shield.



Checkpoint!

Take a look at the connections and compare it with the one on the right. Are they correct? By now we have completed all of our connections to the Stem Base Shield. It is time again to check all connections and make sure they are not loose.

Good job with the hardware!

We will now make the buzzer notify us if we would enter a wrong key combination with 2 long beeps. If we are correct then we get 3 short beeps.

Our code will look like below:
Code available online at:
https://github.com/wendellinfinity/GroveBarkingDog/blob/master/_04_Final_Code/_04_Final_Code.pde

```
#include <Wire.h> // include I2C library
#include <GroveMultiTouch.h> // include our Grove I2C touch sensor library
#include <GroveSoundRecorder.h> // include our sound recorder library

// define sensor pins
#define SOUNDRECSEL1 2 // Grove pin 2 for sound recorder
#define TOUCHINT 7     // arduino pin 7 for I2C touch interrupt
#define BUZZER 8       // Grove pin 8 for buzzer signal
#define PIR 10         // pin 10 for PIR signal
#define PIRIND 11      // recycle N/C pin from PIR to led test indicator
#define PASSCODELEN 3  // length of passcode

// initialize a recorder
GroveSoundRecorder recorder(SOUNDRECSEL1);
// initialize the Grove I2C touch sensor
GroveMultiTouch feelers(TOUCHINT);
// keep track of 4 pads' states
boolean padTouched[4];

// keep states
boolean isMoveDetected; // if PIR triggered
boolean isCodeCorrect; // if code is good
byte inputcode[PASSCODELEN]; // user input code
byte password[PASSCODELEN] = { // secret code
     3,1,2};
int inputcounter;

void setup() {
     Wire.begin(); // needed by the GroveMultiTouch lib
     // initialize the containers
```

```
    for(int i=0; i<=3; i++) {
        padTouched[i]=false;
    }
    // initialize the touch sensors
    feelers.initialize();
    inputcounter=0;
    for(int i=0; i<PASSCODELEN; i++) {
        inputcode[i]=0;
    }
    isCodeCorrect=false;
    // initialize the sound recorder
    recorder.initialize();
    // initialize buzzer pin
    pinMode(BUZZER,OUTPUT);
    // initialize motion sensor pin
    isMoveDetected = false;
    pinMode(PIR, INPUT);
    pinMode(PIRIND, OUTPUT);
}

void loop() {
    if(digitalRead(PIR) && !isMoveDetected) {
        //Serial.println("Something moved");
        digitalWrite(PIRIND,HIGH);
        if(!isMoveDetected) {
            recorder.beginPlaybackLoop(TRACK2);
            isMoveDetected = true;
        }
    }
    if(isMoveDetected) {
        feelers.readTouchInputs(); // test read the touch sensors
        // loop through our touch sensors 1 to 4
        for(int i=0; i<=3; i++) {
            // get the touch state based on pin #
            if(feelers.getTouchState(i)) {
                if(!padTouched[i]) {
                    // sound the buzzer
                    digitalWrite(BUZZER,HIGH);
                }
                // flag the touch sensor state
                padTouched[i]=true;
            }
            else {
                if(padTouched[i]) {
                    // turn buzzer off
                    digitalWrite(BUZZER,LOW);
                    if(inputcounter==PASSCODELEN) {
                        inputcounter=0;
                    }
                    switch(i) {
                    case 0:
                        inputcode[inputcounter]=1;
                        inputcounter++;
                        break;
                    case 1:
                        inputcode[inputcounter]=2;
                        inputcounter++;
                        break;
                    case 2:
                        inputcode[inputcounter]=3;
                        inputcounter++;
                        break;
                    case 3:
                        delay(500);
                        // check if input code is good
```

```
                                isCodeCorrect=true;
                                for(int i=0; i<PASSCODELEN; i++) {
                                     if(inputcode[i]==password[i]) {
                                          isCodeCorrect=true;
                                     }
                                }
                                // check if shutdown code is correct
                                if(isCodeCorrect) {
                                     // turn off alarm
                                     isMoveDetected = false;
                                     digitalWrite(PIRIND,LOW);
                                     recorder.stopPlayback();
                                     // sound buzzer 3 times for correct
                                     for(int b=0;b<3;b++) {
                                          digitalWrite(BUZZER,HIGH);
                                          delay(100);
                                          digitalWrite(BUZZER,LOW);
                                          delay(100);
                                     }
                                     delay(5000); // delay to settle down
                                }
                                else {
                                     // sound buzzer 2 long beeps, means wrong
                                     for(int b=0;b<2;b++) {
                                          digitalWrite(BUZZER,HIGH);
                                          delay(250);
                                          digitalWrite(BUZZER,LOW);
                                          delay(200);
                                     }
                                }
                                // reset user input code
                                inputcounter=0;
                                for(int i=0; i<PASSCODELEN; i++) {
                                     inputcode[i]=false;
                                }
                                break;
                           default:
                                break;
                           }
                           delay(300);
                      }
                      // reset the touch sensor state
                      padTouched[i]=false;
                 }
            }
       }
}
```

Try to load this up to our Seeeduino and look at how it works.

Do the LED light up and the sound recorder play TRACK 2 when it detects your movement?

When we touch 1, 2, 3 and 4(enter), does it do 2 long beeps and the barking goes on?

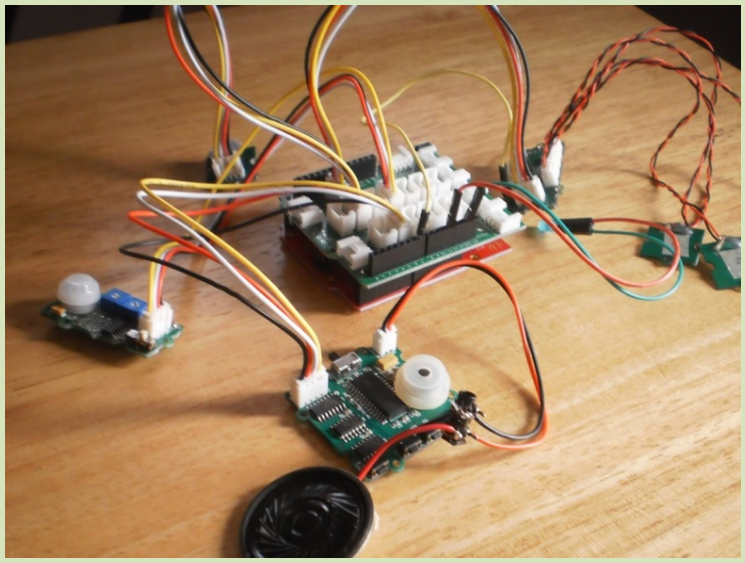
When we touch 3, 1, 2 and 4(enter), which is the correct code, does it beep 3 times, then the barking stops?

When we touch 1 and 4(enter), does it do 2 long beeps and the barking goes on?

When we touch continuously (more than 4 touches without 4(enter)), does it do 2 long beeps and it still barks?

When we touch 4(enter) without any touches with the code numbers, does it do 2 long beeps and it still barks?

Finished Circuit

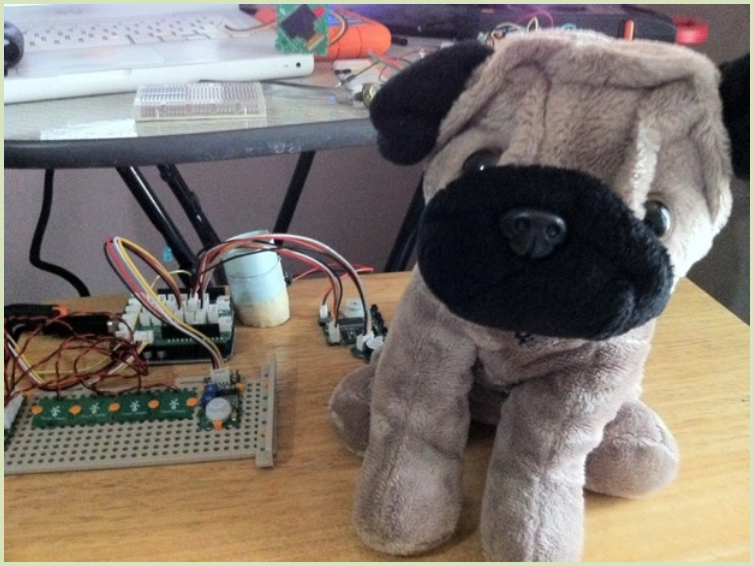


Our setup will now look like this.

We can say at this point that: Good job for finishing the circuit!

We will now integrate it with our stuffed toy dog.

Integration Time!



We would have to consider the size of the dog against the components that we would put in.

Materials:

Stuffed toy dog

We will now start to put all of our components inside the dog! Note that we will have to choose a stuffed toy that is big enough to hold all our components inside, specially the Seeeduino and Stem Base Shield.

Rest the toy dog belly up on a nice flat and well lighted surface

Be careful with handling sharp objects!

Materials:

1 Cutter or scissors

We will now open up the dog laterally and create an “incision” which is big enough for us to work on. Typically we can base our cut using the size of the Seeeduino with a little more than 1 inch at each end of the cut.

We will now see what is inside of the stuffed toy.

Let us take out all of the cotton from just the belly of the stuffed toy and put it aside in a bag. However, we will still leave the head and the legs with their stuffing on them.

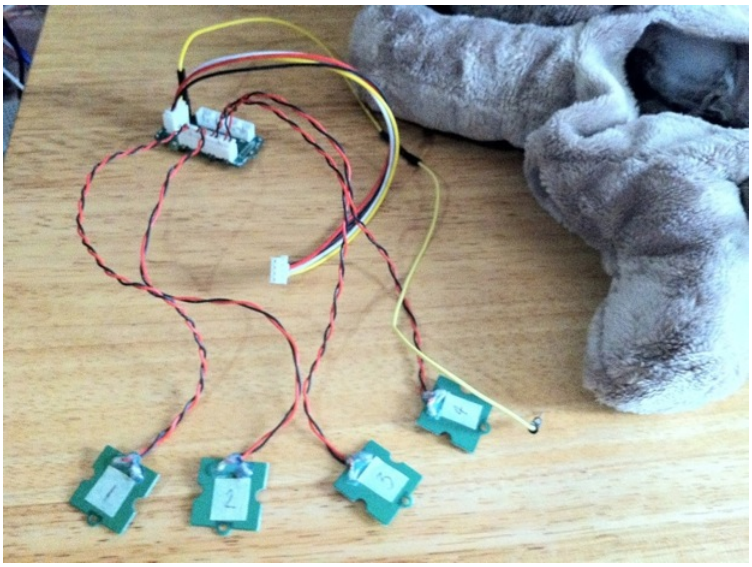
Note: most stuffed toys now contain a sack filled with small plastic granules or as they call it "beanie bags." If you have this, set it aside from the cotton stuffing as we might have a use for it later.

Materials:

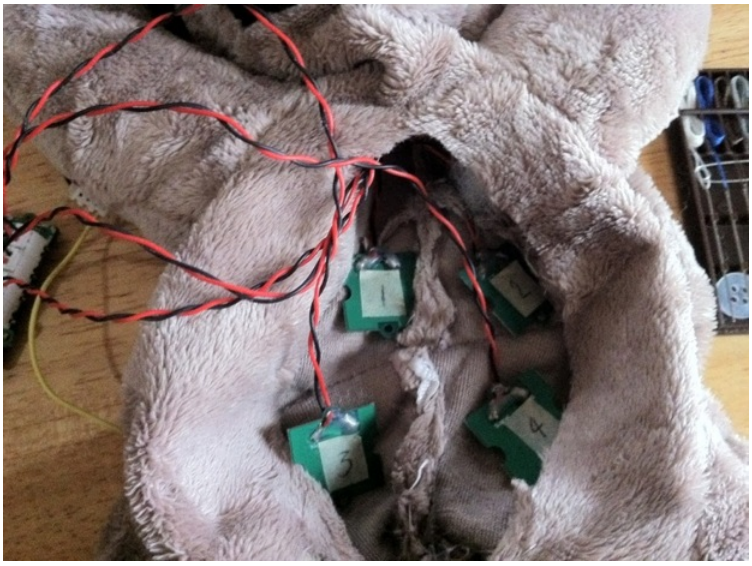
Small sewing kit; or just a sewing needle and a bobbin of thread.

If you can get a small sewing kit (the picture shown here is a small kit from a hotel) then we can expect that the thread has already been looped into the needle.

But a sizable needle and a bobbin of thread can be used anyway.



First stop; let us get our Twig I2C Touch Sensor first.



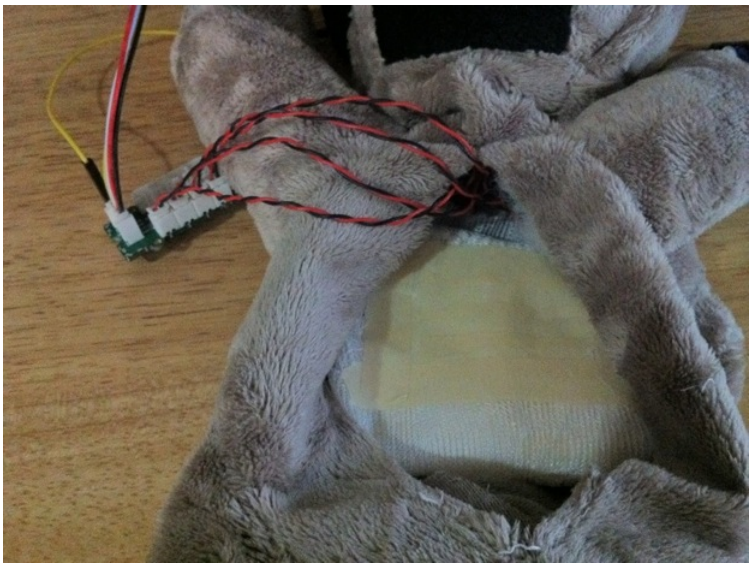
Place them flat inside of the dog with the feelers faced against its fur. From the illustration here, we can see that the backside of the feelers is the one that would be visible. You can go ahead and experiment on the positioning of the feelers.

Go ahead and sew it. Typically, you can use the hole on the feelers to anchor the sensor first and there after wrap it around with the thread to avoid it moving around.

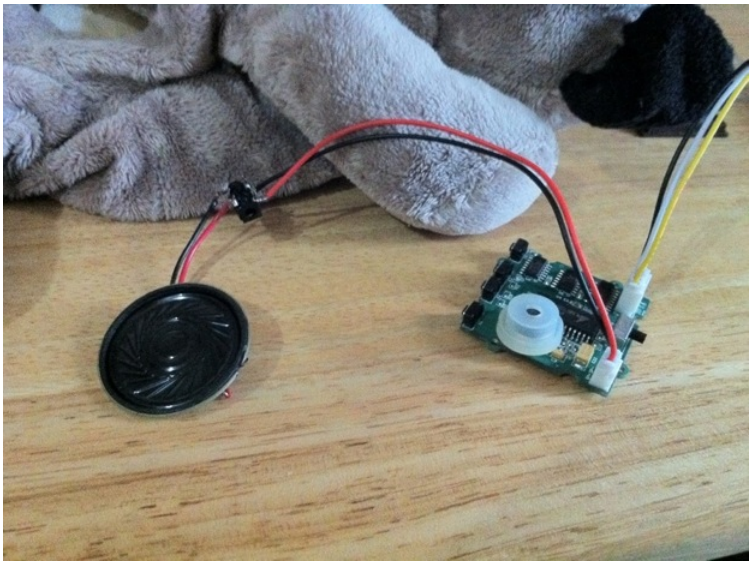
Checkpoint!

At this point, we can see that our touch feelers have been sewn already. From the illustration here, we can see that the feelers in a 2x2 grid pattern.

Then here is the beanie bag just in case you have it. We can use it to put on top of our touch sensors so we will get a more stable feedback when we press against the fur with the feelers beneath it.



We can place the beanie bag on top of the sensors inside the dog's belly. If you don't have this beanie bag then we can stuff in some **packaging bubble wrap** instead. But be sure not to pop them all before putting it in ☺



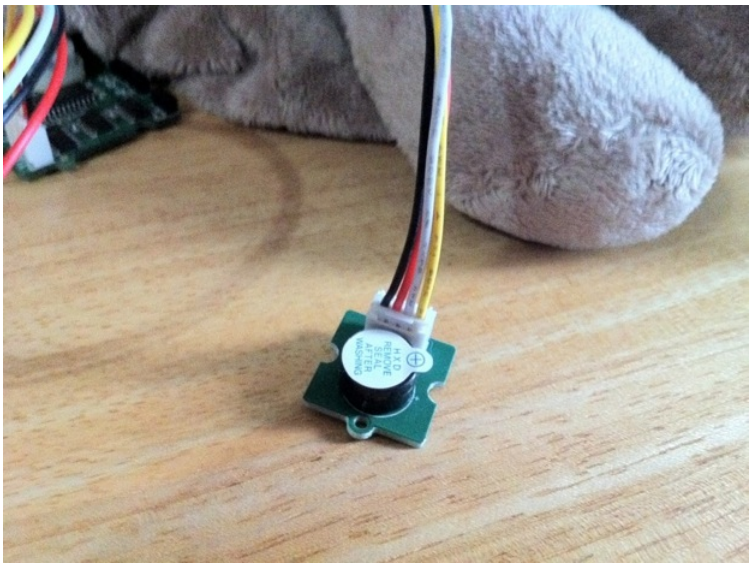
Next stop is our Twig Sound Recorder.

Detach the speaker first from the Sound Recorder and then stuff it in deep the dog's head, maybe near its mouth.

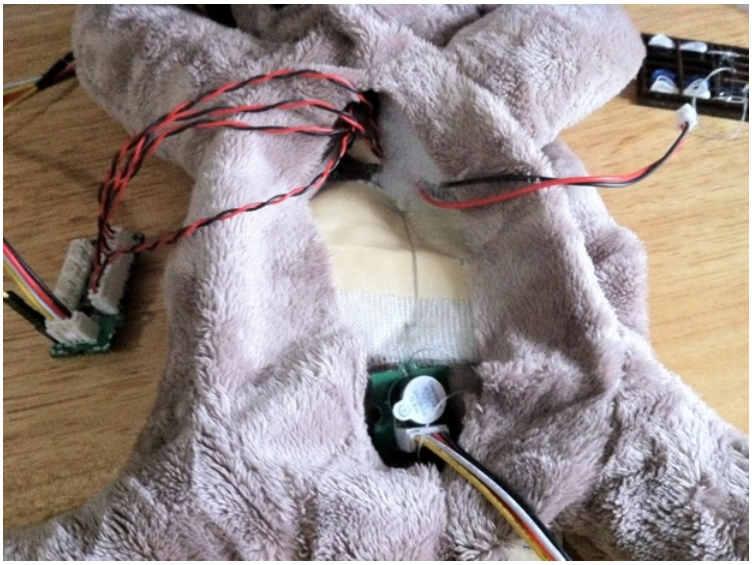
Checkpoint!

We will not see the speaker anymore here because it is now under the fur of the dog's mouth. The audio jack is also along with it and we can attach the audio cable extension.

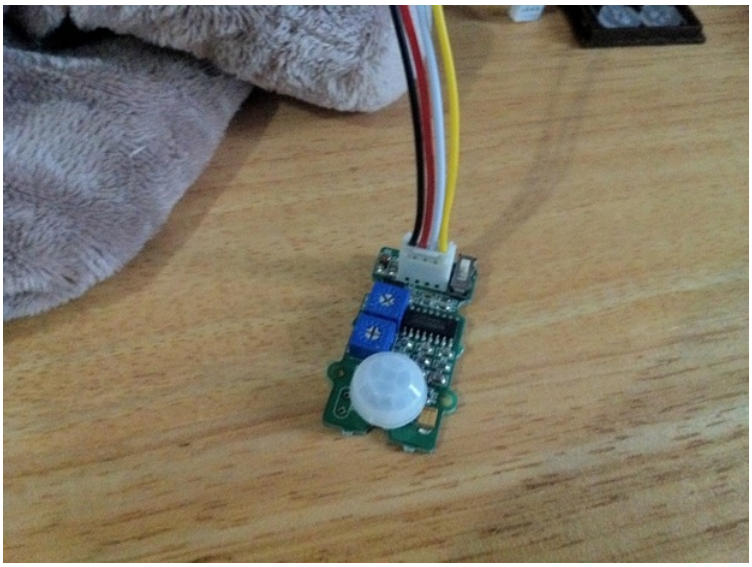
We can even sew the audio cable inside to avoid it getting pulled off easily.



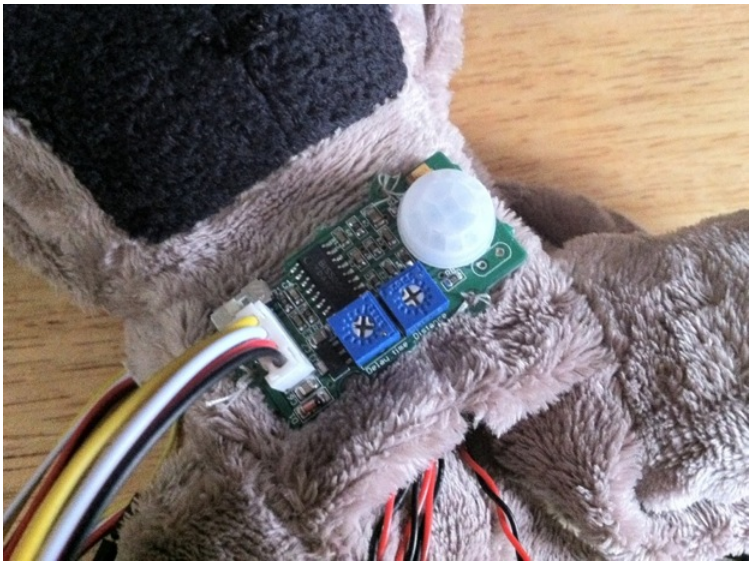
Our next component is the buzzer.



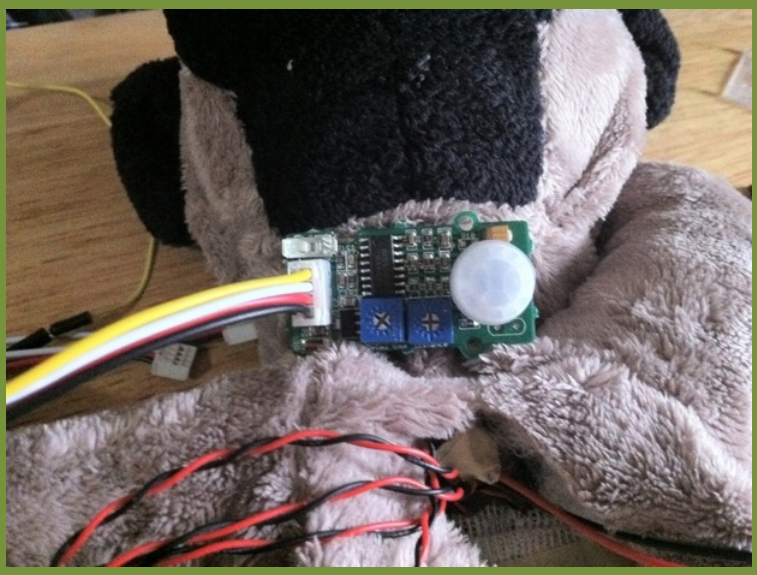
We can place the buzzer anywhere we want, and on the illustration here, the buzzer is sewn near the back of the dog.



Then let us now sew the Twig Motion Sensor on the dog's neck. This sensor has 3 holes on it and we can use them as the anchoring points.

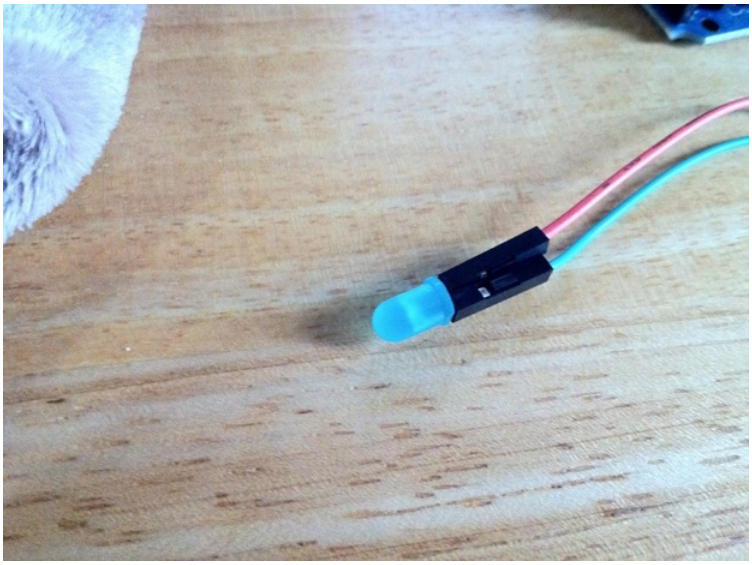


The Motion Sensor is now sewn onto the dog's neck.



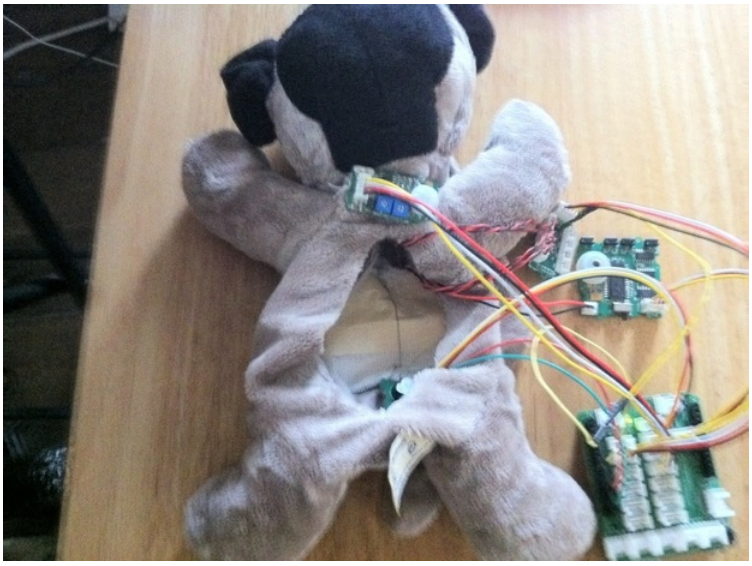
Checkpoint!

At this point, most of the bigger sensors have been attached to the dog. Check that all the wires does not twist with each other and also the all connections are not loose.

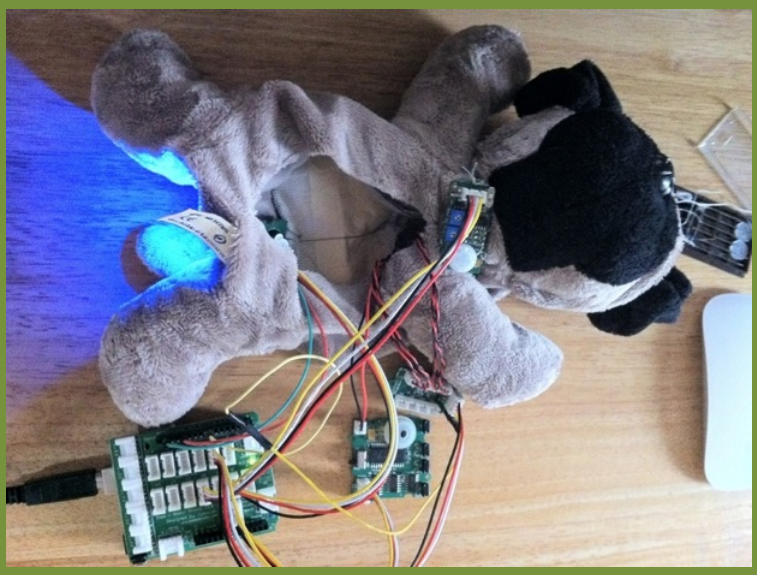


Finally, our LED indicator.

We can choose to put this anywhere by just punching a hole in any part of the dog. Shown in the illustration is that it is placed near the dog's tail, to make it a bit funny and seem robotic (just like Astroboy).

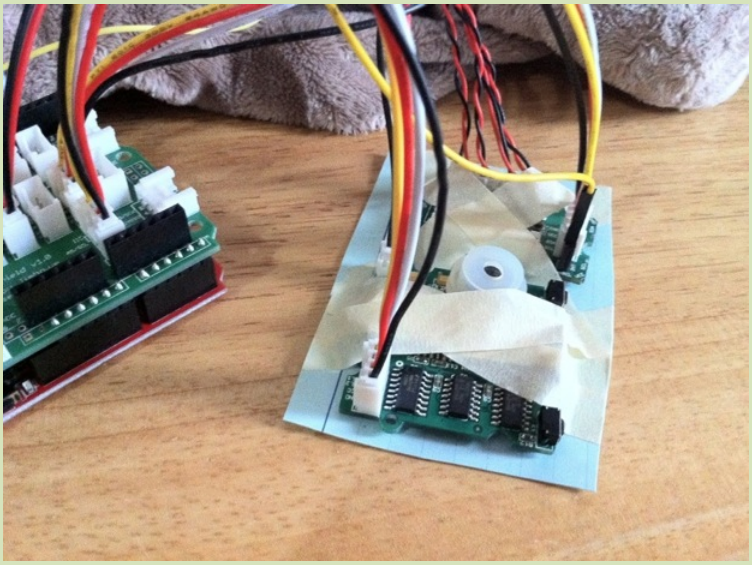


Then here it is! All of our sensors are now in place and attached to the Stem Base Shield.



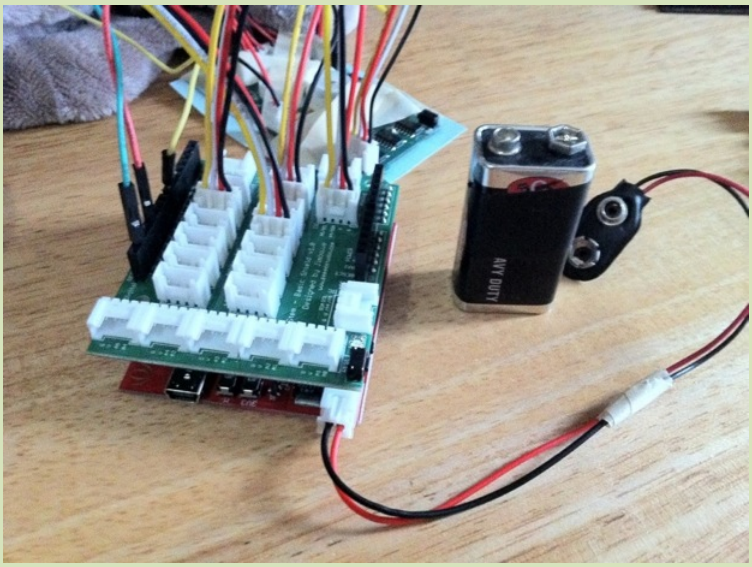
Checkpoint!

We can now try to do a test run of the circuit being active and check out all of the functionality (the feelers, motion detection, sound and buzzer) if they are all working.



Materials:
Small piece of cardboard
Masking tape
Scissors

Cut out a small piece of cardboard, which is about the size of the Sound Recorder and Touch Sensor placed side by side. Then tape these two components together on the cardboard. Set aside for a while.

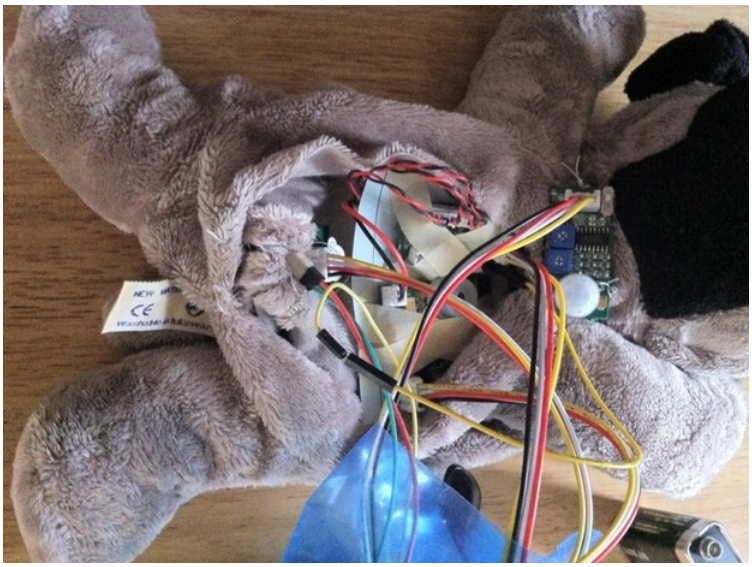


Materials:
1 9v connector
1 9v battery

Then let us get our power source. Shown here attached to the Seeeduino is a 9v battery clip. But do not attach the battery yet!

Materials:
1 anti-ESD bag

For sure you have some ESD (electro static discharge) protection bags lying around most specially when you by electronics. Get one and put in the Seeeduino with attached Stem Base Shield. We would not want our electronic components be damaged when the dog produces static electricity because of its material.



Put in our I2C Touch Sensor board/Sound Recorder board inside the dog, with the cardboard resting on top of the beanie bag/bubble wrap.

Then put our Seeeduino/Stem Base shield on top of the other sensor boards, with the Stem Base Shield facing down.

From the illustration, we will see the bottom of the Seeeduino.

Woof woof!
I am guard dog!

Sew in the tummy of the dog up to 3/4 towards its tail and then attach the battery. You can even sew in zippers or buttons! It will be yours to continue.

From the illustration here, we can easily remove the battery from the 3/4 opening.

Now I have left some problems/exercises for you to work on and extend:

- How will I arm/disarm the dog when it is not in use?
- How will I integrate an on/off switch?
- Can I change the dog's barking? How?

If you have your answers or suggestions, email me at:
wendell.hack@gmail.com

And that's it! We now have our barking desk guard dog that we can use and scare off people who are trying to take something, lets say your chocolate bar, from your desk!

I hope you enjoyed this guide!

For questions, contact me:
Wendell
wendell.hack@gmail.com
http://wendellinfinity.wordpress.com

Thank you Seeedstudio for these great products!