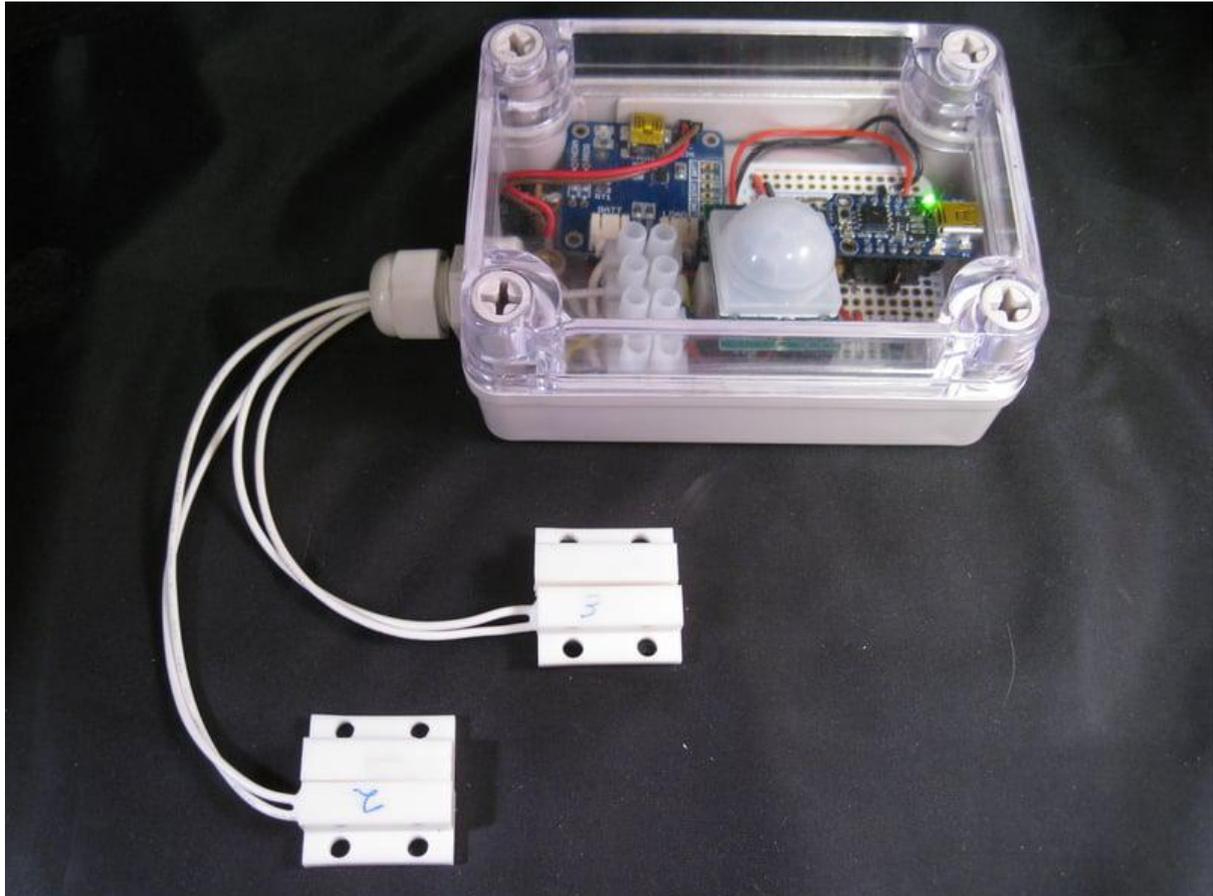




# Trinket Bluetooth Alarm System

Created by Anne Barela



<https://learn.adafruit.com/trinket-bluetooth-alarm-system>

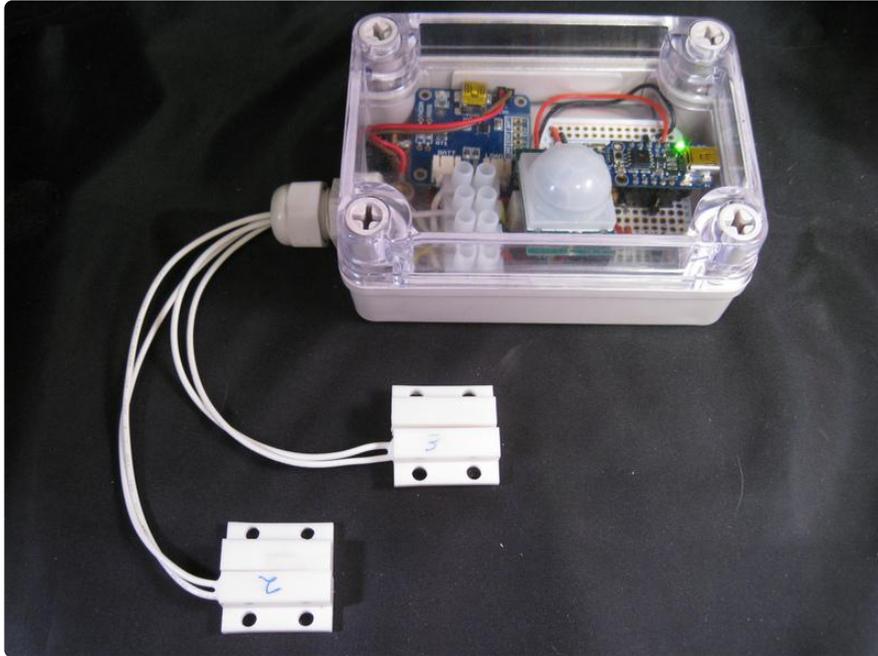
Last updated on 2022-12-01 02:05:12 PM EST

# Table of Contents

Overview	3
Alarm Basics	4
• Theory	
• Multiple Sensors, One Pin	
Design	6
• Annunciation Selections	
Build	8
• Populating the Board	
Code	13
Test	16
• Troubleshooting	
Installation and Use	18
• Walk Test	
• Use	
Going Further	20
• Fancier Annunciation	

---

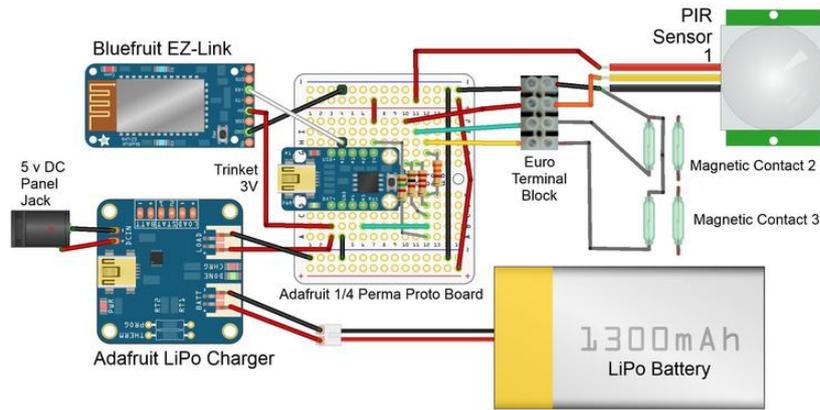
# Overview



Many alarm projects are rather simplistic: a single sensor protects a single object or field of view. Such projects are good for learning or practicing simple circuit techniques. But eventually you may wish to have a more advanced / sophisticated alarm system. More like those used by commercial alarm companies. Actually the circuits needed to accomplish this are still fairly simple and can be built around the Adafruit Trinket mini microcontroller.

This tutorial will provide you with:

- The basics of how alarm systems are function
- Alarm system design
- Building a practical alarm
- The code
- Installation and use
- Bluetooth wireless information transmission
- How to add additional capability



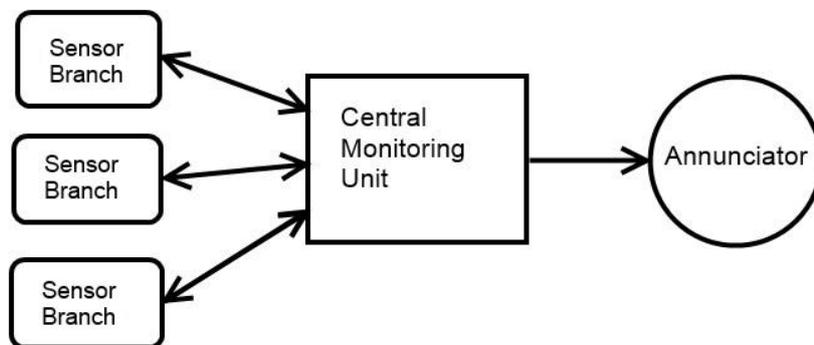
This project was demonstrated on Adafruit's Show and Tell on December 7, 2013

<https://www.youtube.com/watch?v=yw5I9pKP2Uo> ()

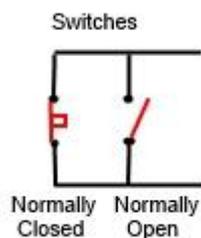
## Alarm Basics

### Theory

Most alarm systems follow a basic design. A switch closure (or opening) triggers action by a central monitor and the alarm generates some form of annunciation.



The sensor blocks may actually be multiple sensors in one monitoring branch. Designs can be made which give only one indication the sensor chain has been triggered or if you have a more smart design, multiple sensors will provide individual indications.

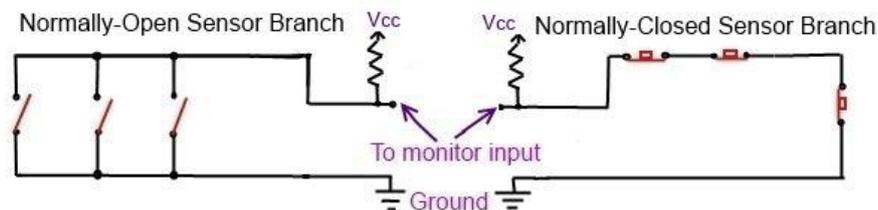


Switches can be normally closed or normally open. When they are triggered, they change state (closed goes open, open goes closed). The monitoring unit notices the change and if it meets criteria that it believes is an alarm, it annunciates.

Annunciations can take as many forms as sensors. The stereotype alarm system has a very large horn to make an ear-splitting sound. But it can be much more subtle. The central monitor can alert a "silent alarm": making the alarm known in some way, either locally or far away. An example would be a text or SMS message on your phone stating an alarm has been triggered. The simplest alarm may only trigger a local alarm not designed to scare the intruder - this can be a nearby notification or it could be a record of alarm events for later review.

## Multiple Sensors, One Pin

You can use nearly unlimited sensor switches if you place multiple switches in a branch. The branch can sense one switch out of all that may have changed state in a branch. In this manner you can only tell if any one of the multiple sensors in the branch has tripped. You can do this using normally open sensors in parallel or normally closed sensors in series. You can use both types of sensors putting the closed switches in line and the open switches in parallel. The resistor pulls the circuit high if the loop is opened otherwise the loop is grounded.

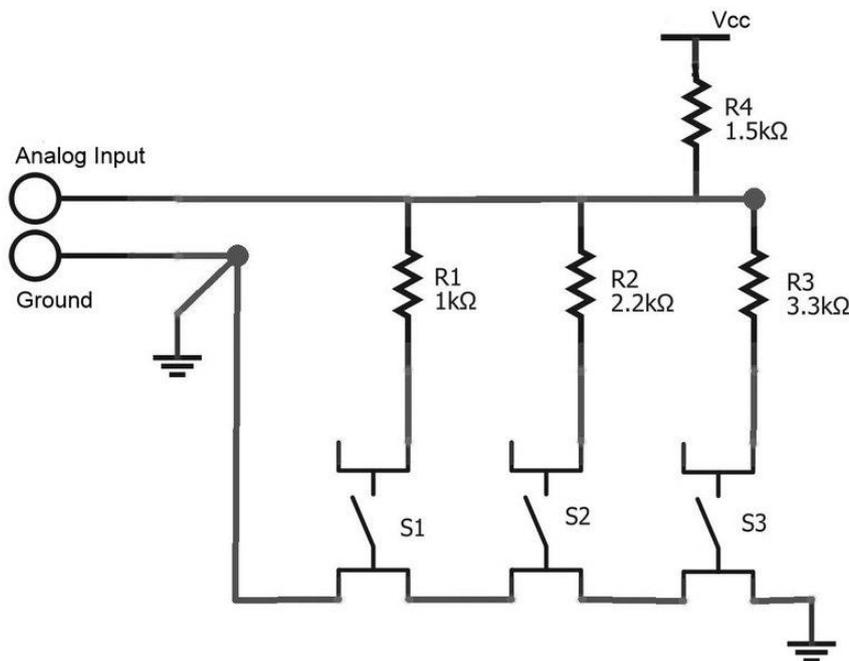


There are circuit methods that allow you to identify which switch in a branch was tripped if you use an analog pin. These use resistors to change the voltage values the analog pin reads. Typical "display shields" use this to read 4 to 6 switches to determine what was entered.

But many analog switch reading circuits have a problem: they cannot determine if two buttons are pressed at the same time. For a simple alarm system, this might not matter - an alarm happened but you only know it is on a specific branch. With a bit more circuit design, you can determine if multiple sensors have tripped on a single branch and which ones they were. The method most often in textbooks is the R-2R resistor ladder.

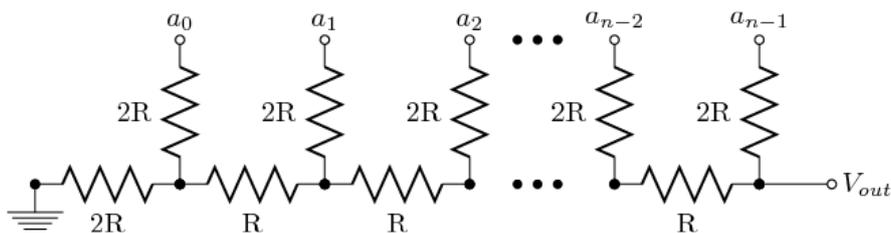
Here I will demonstrate a simplified design of my own, a parallel resistor system that uses less components and has good accuracy for Arduino-type analog inputs. The

complexity grows with the number of sensors, so I will show for three sensors on a branch.



Normally the Analog Input A1 is pulled high by resistor R4. If any of the switches S1, S2, or S3 is closed, the resistance changes in a predetermined way. Circuit-wise, each of the resistors R1, R2, and R3 would add to the total resistance using the familiar circuit formula  $1/R = 1/R1 + 1/R2 + 1/R3$ . All we have to do is measure the analog values read by the Trinket and add them to our code. This method also works equally well for normally open or normally closed switches.

This method does take only four resistors whereas the pure [R-2R ladder method](#) () requires five to six resistors for 3 switches).



R-2R ladder (courtesy [Wikipedia](#) ())

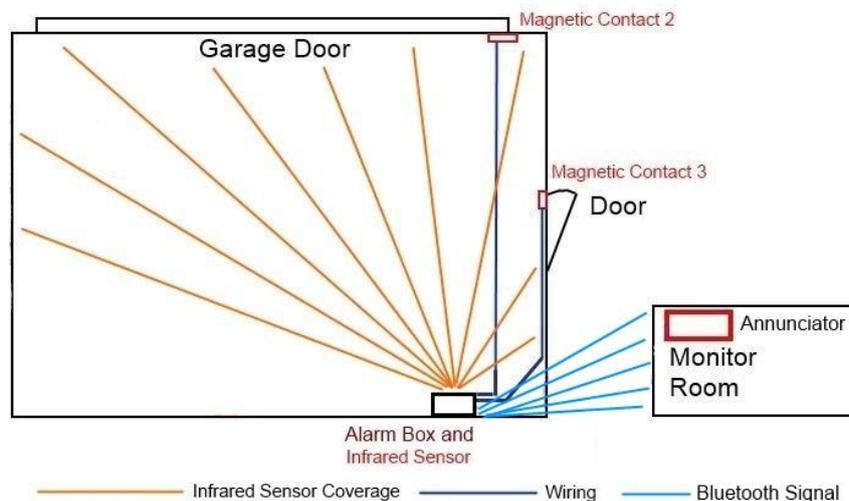
## Design

This tutorial will take sensors, add central monitoring code, and output alarm events to an annunciation system (another computer connected via Bluetooth). Several configurations will be shown to allow you to configure your own alarm system.

The Trinket has the capability to design an excellent alarm at a price point much lower than other alarm platforms. There are 5 general use pins, three of which may be digital or analog. At a minimum, you would need one sensor (or sensor branch) in, and one annunciation path out. That uses two of our five pins, thus you can have an annunciator that uses up to 4 pins or several sensor branches and more modest annunciator.

We will use a generic monitored area in the presented design. A single area alarmed and annunciated either locally or to a remote site. You can use the technique to wire multiple rooms (doors, windows, movement) but there are limitations to the number of sensors if you want to know the exact sensor number. If you only need to know "something" happened in a sensor chain, a nearly unlimited number of sensors may be used.

A typical scenario is monitoring a place like a room or a garage. The garage has a main garage door, a side door and the space within the room (a regular room would substitute a window for the garage door). To cover the area well, a magnetic contact will be placed on each of the areas that can open/close plus a volumetric (area) sensor in case one of the other sensors fails or someone breaks in without tripping a magnetic contact. You can always go to extremes covering an area but it only costs resources (money and processor pins) without providing much more security.



This design will use one analog sensor branch per the design diagram. The Infrared sensor and the magnetic contacts will be wired together into an analog ladder per alarm basics. These will be transmitted via Bluetooth to a monitor room inside the house. A good design has the monitor away from the area to not allow the intruder to disable the alarm easily.

This leaves many pins for additional sensors or other annunciation.

# Annunciation Selections

Again there are many ways to alert you if the alarm has tripped. Some of my favorite ways:

1) Speaker: the simplest method is to use a loud noise. The piezo speaker does this well - you can output several tones or just an ear piercing tone. You can use an audio amplifier to make a louder sound. You will need to match the wattage of your speaker to that of the amplifier. Between the Trinket and amplifier ensure you do not input too much power or voltage.

2) Display: if you leave Pins #0 and #2 (the I2C pins) free, a character display with backpack could display the status. You may also use an LED as a simple indicator. The Trinket Pin 1 built-in LED is perfect for this although Neopixels or other LEDs could also be used.

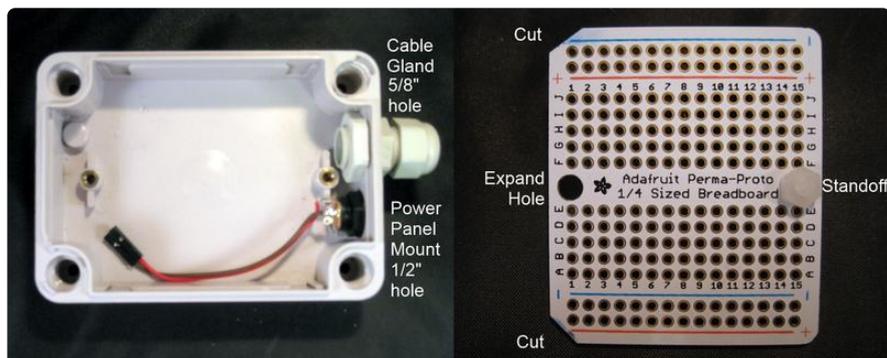
3) Communication: My favorite - leverage the power of your home network to get a signal out. Trinket does not have enough memory to use a Wifi shield directly. But a serial link like new [Bluefruit-EZ -Link \(http://adafru.it/1588\)](http://adafru.it/1588) or an [FTDI Friend \(http://adafru.it/284\)](http://adafru.it/284) can be sent to another computer to get the word out - leverage your cloud to spread the alarm message.

or The Combination: If the infrared sensor and all the magnetic switches are on a combined branch on an analog input, then multiple annunciation methods could be available.

The final project built uses the Pin 1 LED and a Bluefruit.

---

## Build



The [small weatherproof enclosure \(http://adafru.it/903\)](http://adafru.it/903) is an economical enclosure for the project. The layout will be a bit tight (things are layered in the enclosure). If you need additional room, the large enclosure or an Otter Box is plenty big. The [1/4 sized Perma Proto Board \(http://adafru.it/589\)](http://adafru.it/589) is a great way to lay out circuit components after breadboarding.

The [2.1mm panel mount barrel jack \(http://adafru.it/610\)](http://adafru.it/610) is used for power - solder power wires on prior to installation (large lug is positive). Drill a 1/2 inch hole just past the cover hole. The plastic dimple was chisled to fit it in that location. Thread the jack in securing with the included plastic screw ring. A [large cable gland \(http://adafru.it/761\)](http://adafru.it/761) is used to run sensor/signal wires through the enclosure keeping things weathertight. Drill a 5/8 inch hole between the power jack and the other case cover hole. There is a small lip on the case - I dremeled it down a bit to have the gland fit snug.

Best to make mounting holes in the back of the case if needed to screw the case into the wall. This would probably be harder later in the build given the amount of items we are putting in.

The 1/4 Perma Proto is nearly perfect. The corners should be cut just a bit and the left hole expanded slightly to allow it to fit snug screwed on the left brass mounting hole. A small 8-32 screw is fitted on the left and I used an optional standoff through the right hole.

## Populating the Board

You can refer to the Fritzing diagram on the Overview Page for the parts placement.

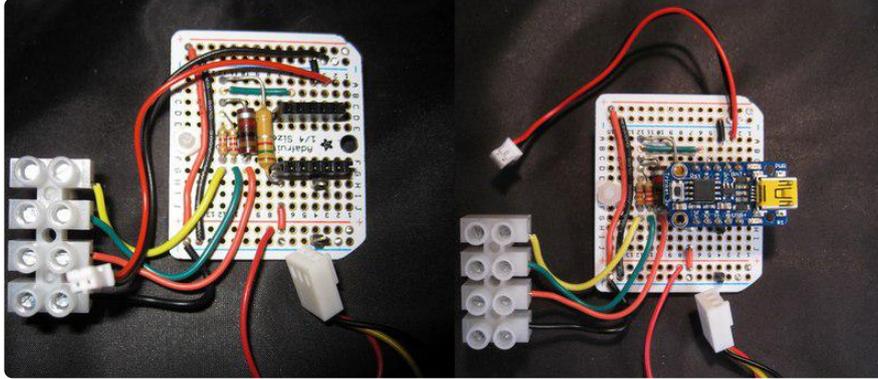
I used two pieces of female header, 5 pins each, to mount the Trinket to the board. Single male pins are placed on the - line, the Trinket BAT+, and Trinket Pin 0 for connecting Bluefruit later although you can use hook-up wire if you like.

Four resistors provide the analog divider to multiplex three sensors. The 1500 ohm resistor (the large one in the pictures, your size is probably smaller, I used what was on hand) goes between the trinket 3.3 volt regulated pin to the common resistor junction. Each of the other resistors is connected to this common area and connected to Trinket Pin 3 (Analog pin 3). Four wires are run off-board to connect the sensors.

The black and red wires connect the power ground. The board is powered via a JST connector with attached wires that comes with the Adafruit LiPo charger board (near the top of the photos). The red wire goes to the Trinket BAT+ pin, the other to the pin labeled Gnd. Other parts of the circuit draw regulated power from the pin marked 3V (which is regulated by the Trinket to 3.3 volts and provides up to 150 milliamps).

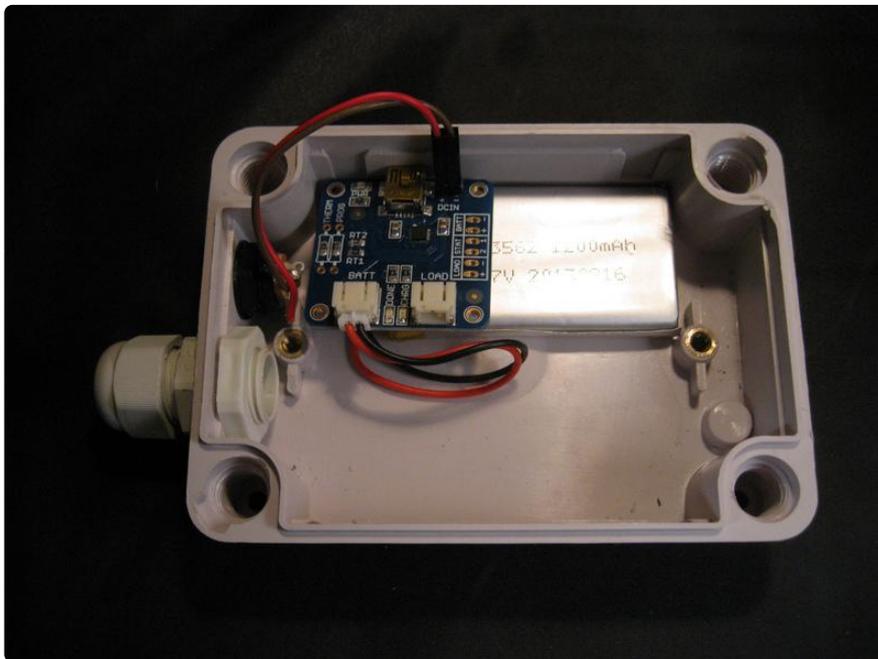
The connector for the PIR (infrared sensor) is connected to regulated power also (bottom of pictures).

Double check your connections.



The battery is placed in the case and the LiPo charger board placed on top near the power connector. Solder or plug in the wires from the power connector to the DC in connections on the charger (near the USB connector).

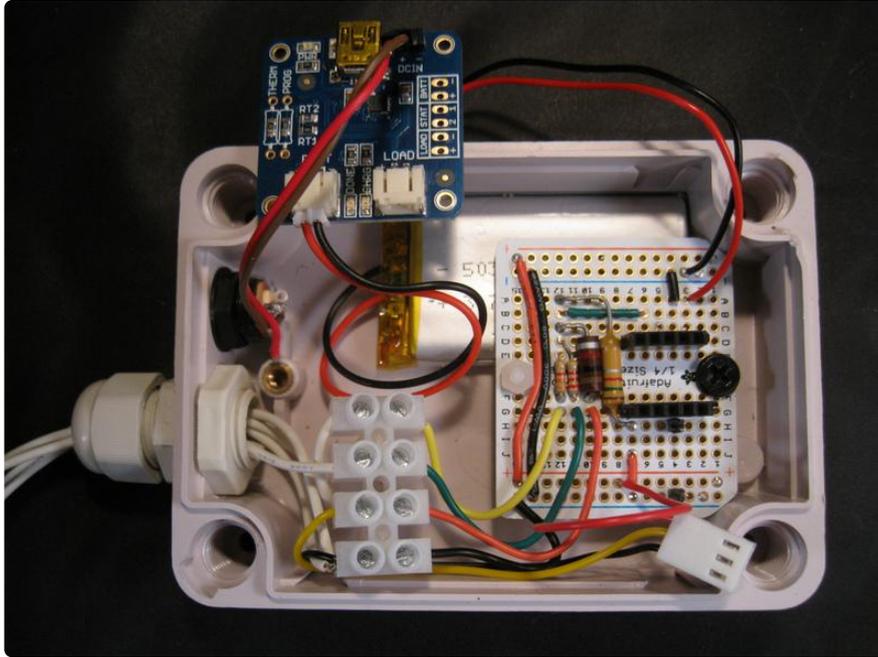
At this point it may be best to plug in the 5 volt power supply and charge the LiPo battery.



Screw the wired Perma Proto board into the threaded connector with an 8-32 screw (not provided) or otherwise secure. A standoff on the free end is helpful as a counterpoint.

Wire the three sensor wires to a terminal strip for ease of connection. Below shows

the bottom common (ground) connection, then the PIR switch wire (yellow), then magnetic sensor #2 and magnetic sensor #3 towards the top of the picture. You can solder wires instead or use a [spring terminal block](http://adafru.it/1074) (<http://adafru.it/1074>) on the board itself although some of the wires may not fit well as the magnetic contacts have large stranded wires.

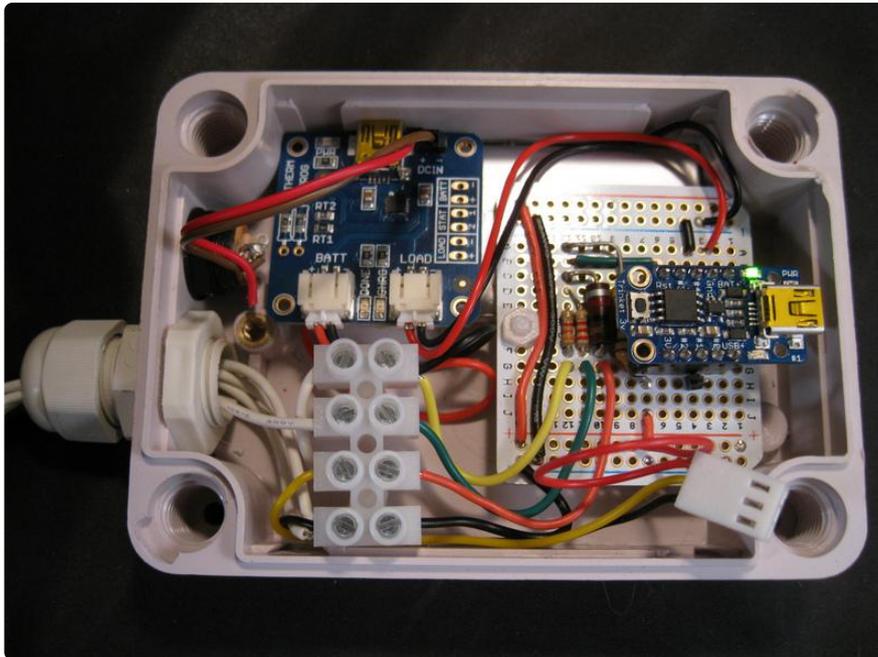


Ensure you have set up the Arduino IDE according to the Introduction to Trinket tutorial so the programming will work well. Programming Trinket is a bit different than other boards.

Program the Trinket 3V off board with the code in the next section or with your custom code.

Insert the Trinket into the pins.

Plug the circuit board power connector into the LOAD connector on the LiPo charger. Layer the LiPo charger board on top of the battery.



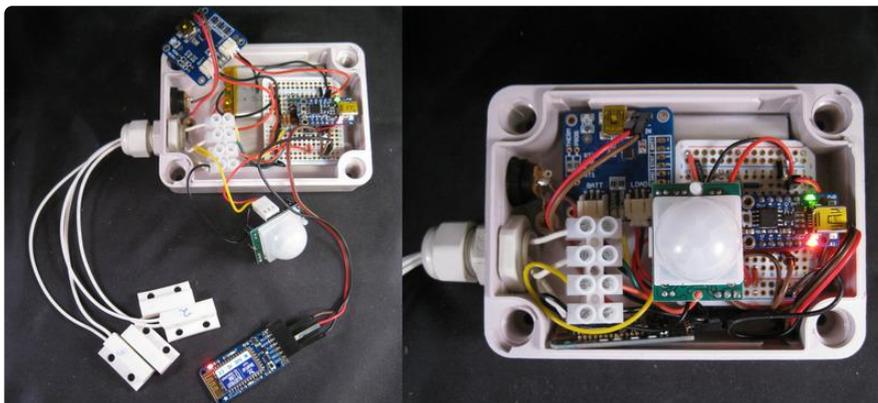
Next add the PIR sensor. Connect the PIR ensuring the red wire is to +5, black to GND. The PIR has two orange variable resistors, one to adjust sensitivity, one for latch time. The latch time can be fairly short (we are sensing any change). The sensitivity should be turned down at first and may be adjusted when making the final installation.

The PIR can be placed outside the enclosure but the nice clear case begged to inside mounting. There is enough clearance, but just. For best sensitivity, a round hole for the lens to come out (with some sealant) may be better.

The Bluefruit is connected to the circuit board as follows:

- BAT+ (line into Trinket) goes to Bluefruit Vin
- Ground (-) goes to Bluefruit GND
- Trinket Pin 0 goes to Bluefruit RX

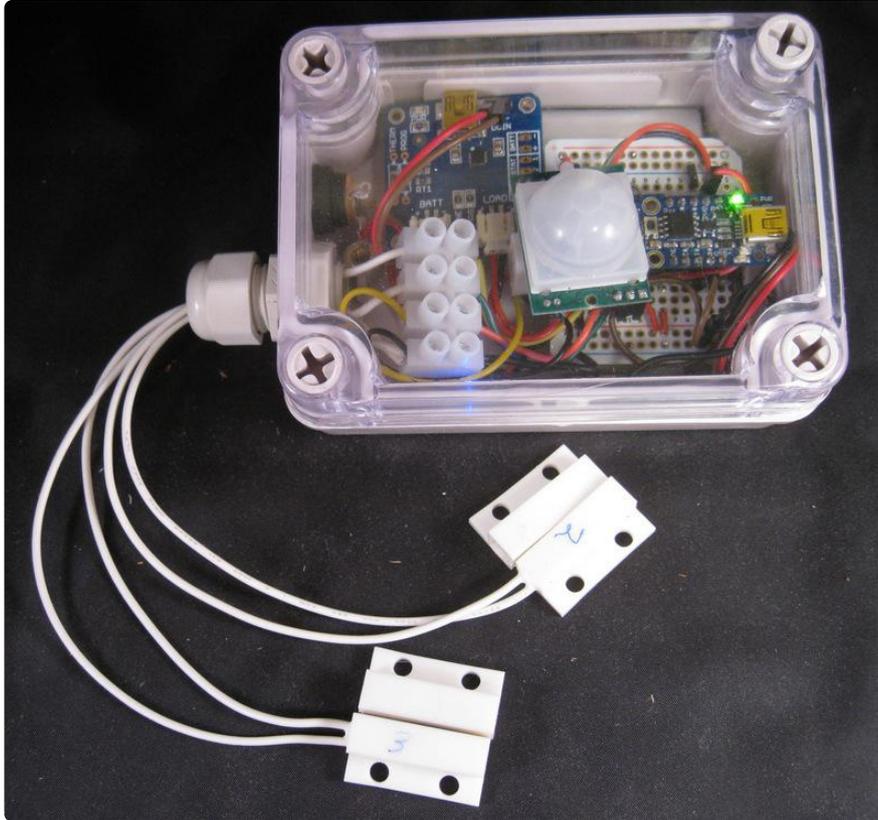
Tuck the Bluefruit in the case upright in what is shown as the side of the case at the bottom of the picture.



With the clear weatherproof case cover attached below.

Do a final test of the sensor function.

It is now ready to take the magnetic sensors off and perform the final install.



---

## Code

The code for a simple alarm with three sensors or branches is shown below.

This configuration has the sensors tied into resistors feeding Pin 3. The Trinket must be programmed out of the circuit as pins 3 and 4 are shared with the USB.

The program uses the standard SoftwareSerial library to talk via the Bluefruit EZ-Link on Pin 0 for transmit, Pin 2 for receive. You can eliminate the requirement to specify a receive pin (and shrink the code slightly) with the third party SendOnlySoftwareSerial library on arduino.cc via this forum thread: <http://forum.arduino.cc/index.php?topic=112013.0> (). This would also allow you to use Pin 2 (Analog 1) for alarms, freeing Pin 3 shared with USB.

Pin 1, which has the onboard red LED, is used as an indicator on which sensors are tripped. When the alarm is set (no alarms tripped), the LED does not blink. It blinks from 1 to 7 depending when sensors are tripped (1 for #1/PIR, 2 for #2, 3 for #3, 4 for 1

and 2, 5 for 1 and 3, 6 for 2 and 3, and 7 for all sensors tripped). If you decide to use Pin 1 for other purposes, you can do so but with the LED in-circuit it can be tricky. For example, the internal pullup resistor, if enabled, is too weak. You may use a fairly low value external pullup resistor like in the hundreds of ohms if you decide to use the pin for sensors.

In the code for the example build, enabling DEBUG (uncommenting the line `//#define DEBUG` by deleting the `//` characters) will output the alarm values for the analog pin to the serial connection. You should do this once your circuit is together to ensure the values read by your circuit gives values understood by the code as alarms are set off.

Trip each combination of sensors, record the value for the analog pin displayed on the serial line. Change the code line that has `uint16_t` values to the values you find. On the bench this process takes less than 5 minutes. If you have problems with the final install giving errors, try this process again as the resistance of the wiring may change the values.

Ensure your Arduino IDE is configured per ALL the steps in the Introducing Trinket tutorial to ensure the code compiles correctly and uploads.

```
/* Trinket Alarm Analog    An alarm system based on the Adafruit Trinket mini
microcontroller

    This version uses three sensors (or sensor branches) connected to one analog line
    Annunciation is performed via the onboard LED and a Adafruit Bluefruit EZ-
    Link

    A Trinket 3V is be used with a 3.7 volt LiPo battery and Adafruit charger
    connected to 5 volts
*/

#define SerialPin 0 // Serial debug via Bluefruit EZ-Link on this pin
#define LEDpin 1 // Use Trinket LED for displaying tripped sensors
#define SensorPin 3 // A3 which is GPIO #3 has resistor network to read 3
normally closed sensors

//#define DEBUG

// Serial code, use a Bluefruit EZ-Link with its RX pin connected to Pin 0 of
Trinket
// You will need a terminal program (such as freeware PuTTY for Windows) to get
the alerts
// but better would be a Processing or Python script looking for alarms.
#include <SoftwareSerial.h> // Software serial library (standard in Arduino
1.x +)
SoftwareSerial Serial(2,0); // Serial transmission on Trinket Pin 0, receive pin 2
(not used)

// Multiplex 3 normally closed sensors on one analog pin. If you have 2 sensors,
// you can leave the one resistor open and adjust the text values accordingly
const uint8_t numSensors = 3; // number of sensors on analog line
const uint8_t states = 8; // 2^numSensors
uint16_t values[8] = {541, 685, 661, 614, 840, 780, 776, 997};
char *textval[8] = {"Set","PIR", "2", "3", "PIR+2","PIR+3","2+3","All"};
```

```

void setup() {
  pinMode(LEDpin, OUTPUT);      // Set GPIO 1 to output to blink LED
  pinMode(SensorPin, INPUT);   // sets analog pin for input
  Serial.begin(9600);          // Send status information via serial
  Serial.println("Alarm System"); // Initialize message (can read to determine
reset)
}

void loop()
{
  int8_t contact;              // read alarm loops (returns -1 if a read
error)
  contact = readContact(SensorPin); //

  if(contact >= 1) {          // if any value greater than 0 (set),
  Blink(LEDpin, contact);    // we have an alarm! Blink LED
corresponding to
  Serial.print("Alarm! ");    // which sensor(s) and write to Bluetooth
  Serial.println(textval[contact]);
  }
  else if(contact <= 0) {    // a bad analog read was done. If you get
errors
  Serial.print("Error");      // set DEBUG, walk test, record values, and
  }                            // update code with analogread values
  else {
  Serial.println("Set");      // Alarm is set (no sensors tripped), all is
well
  }

  delay(500); // We do not need to poll the sensors very often although you can
change
}

int8_t readContact(uint8_t TrinketPin)
// returns the number corresponding to sensor values.
// TrinketPin is the analog pin on the Trinket (A1=#2, A2=#4, A3=#3)
{
  const int variance = 8; // Analog readings can vary, use this value for +-
variance
  int contact = 0;
  uint16_t readval = 0;
  readval = analogRead(TrinketPin); // Check the pin
#ifdef DEBUG
  Serial.print(": Sensor read value: ");
  Serial.println(readval);
#endif
  for(uint8_t i=0; i<states; i++) { // if reading is near state value, return
that state
    if(readval >= (values[i]-variance) && readval <= (values[i]
+variance) ) {
      return(i);
    }
  }
  return -1; // value not one of the alarm system values
}

// This routine toggles a pin the number you pass. Good to use on LED pin to
// output which sensors are triggered
void Blink(uint8_t pin, uint8_t times) {
  for(uint8_t i=1; i<=times; i++) {
    digitalWrite(pin, HIGH);
    delay(85);
    digitalWrite(pin, LOW);
    delay(85);
  }
}

```

Remember to comment out the DEBUG line for your final installation.

---

# Test

Ensure you have made all the connections noted in the Fritzing diagram on the Overview page.

Program the Trinket out of the circuit, then place in the headers on the circuit board. I recommend using headers over soldering directly in due to the connections to Pin 3.

Power up the circuit. The battery should probably charge with the CHRG light lit on the LiPo charger circuit board. The Trinket green LED should be lit. If not, unplug and check your connections.

When working right, the red LED on Trinket is probably blinking showing the sensors that are tripped - this is normal as the PIR sensor probably sees you moving and you might not have the magnets against the mag sensors. Place the magnets next to the mag sensors and aim the PIR away from you. The system should "set up" (no blinking). Then when you move in front of the sensor it should blink once a second or so with more blinks depending on which sensors tripped as follows:

- No blinks - secure
- One blink - PIR tripped
- Two blinks - Magnetic Sensor 2 tripped
- Three blinks - Magnetic Sensor 3 tripped
- Four blinks - PIR and Mag Sensor 2 tripped
- Five blinks - PIR and Mag Sensor 3 tripped
- Six blinks - both magnetic sensors tripped
- Seven blinks - all sensors tripped

If the PIR is not indicating correctly, move the orange potentiometers to reduce or increase sensitivity. If it will not "go off", reduce the latch time.

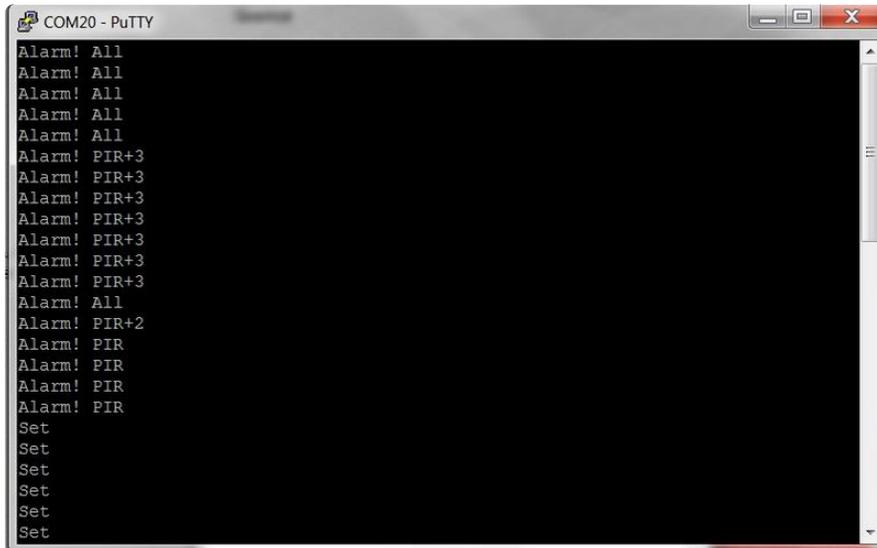
The system state is also broadcast via serial to an Adafruit Bluefruit EZ-Link at 9600 baud. It transmits up to 10 meters (33 feet).

With a laptop/PC with a Bluetooth Receiver, determine which serial port is connected to Bluetooth for your operating system (for Windows, go to Control Panel, Devices and Printers - if you do not see Adafruit Bluefruit listed, use the Bluetooth program in the systems tray (lower right, tiny B icon) to add it to your Bluetooth device list. You may have to press the pair button on the Bluefruit board to have it recognized.

To listen in on the serial stream, you should load a terminal program. Your operating system should have one, you might have to Google to determine it. For Windows, you

can download a good free terminal program called PuTTY [at this website](#) ().

Open your terminal window and set it to work on a Serial stream. Set your serial port (Windows COMxx where xx is the number of the port you found in Control Panel earlier). The baud rate is 9600. When you press the Open button, a black terminal screen should be displayed with white text output from the Bluetooth serial stream.



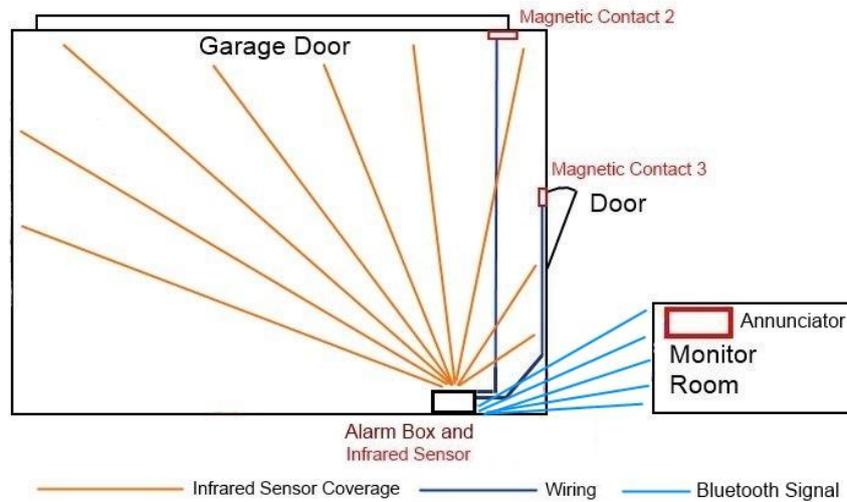
## Troubleshooting

The Bluefruit should have both red and blue flashing LEDs at this point. If not, double check that Trinket Pin 0 is connected to Bluefruit RX and you have GND and Vin connected to circuit ground and BAT+ respectively.

If PuTTY complains about no COM port found, double check the devices in Control Panel -> Devices and Printers and ensure the white "refrigerator" icon is there for Adafruit Bluefruit. If not, open the Bluetooth center by going to the right bottom of the screen, click the arrow to show all notification icons, select the oval with the blue B in it and pair your Bluefruit by pressing the pair button on Bluefruit. Note the COM port that the Bluetooth is using for Bluefruit and use that in PuTTY. Ensure the baud rate is 9600 (or if you changed that in the code you set it the same). The text output is so infrequent, 9600 baud does not slow the circuit down any.

---

# Installation and Use



You will want to install your alarm system box away from entry points. If you can, install the box where it takes a ladder or other pains to get at it so the intruder cannot just rip it off the wall.

You will need to have wall power (mains) nearby for the 5 volt DC step-down power supply.

Even if the system is not installed outside, consider an environmental-rated case for the control unit to keep it dry.

Secure the case to the wall with screws. Best to have case holes predrilled before installation.

String your sensor wires: two conductor wire is straightforward but 3 or 4 conductor could be used for the magnetic contacts to use one run, multiple wires. If you use a fine gauge wire, the resistance will be higher but wires might be a bit easier to run.

You can use many methods to secure the wires: outdoors, heavy-duty staples may work. Plastic sticky tubing (Panduit or similar in the US) works to run cables along walls. Making them hidden or above reach is good. In a carpeted room, you can feed them in the space between the baseboard and the carpet. Ensure wires are firmly in place. You can leave a bit extra at each end to be sure when you hook things up you are not a bit short especially if you cut a wire while stripping it.

Mount your sensors. Mount magnetic sensors above/very near the door/window. Put a multimeter on the wires and you will register an open circuit. Now place the magnet on the closed door/window and move it until it registers as a closed circuit (near zero

ohms) That is where you want the magnet screwed in. You should probably tape the magnet in place and move the door/window open and closed a few times to make sure it consistently registers. When you are satisfied, permanently screw in the magnet and contact. Test to ensure it works the way you expect it to.

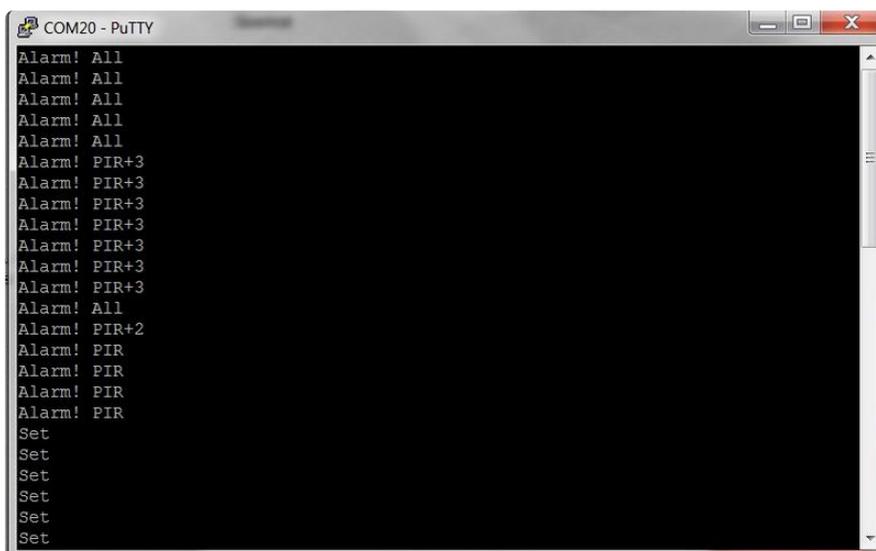
Solder connections between the sensor wires and the wiring to the control box. To keep connections from the elements, wrap them in electrical tape or heat shrink. You might consider a small box to place the connections in.

## Walk Test

The act of testing your system is called walk testing. First, the green LED of the Trinket should be on. If you use Bluetooth, ensure the computer is close by to monitor your work. A second person would be helpful at this point to avoid having to test then run to the PC to see what it did.

Close all the doors/windows. If you have a motion sensor, stand still so it stops registering movement. The red LED on the Trinket should not blink and the Bluetooth serial monitor will indicate "Set". Now "trip" each sensor in turn and check that the correct number of blinks are displayed on the LED and the Bluetooth link states the correct indications on the PC. If everything is working, each sensor trip (or combination) will work as intended. If not, check your connections.

Below is a terminal window with various sensors tripped then all sensors set up (no alarm).



```
COM20 - PuTTY
Alarm! All
Alarm! All
Alarm! All
Alarm! All
Alarm! All
Alarm! PIR+3
Alarm! All
Alarm! PIR+2
Alarm! PIR
Alarm! PIR
Alarm! PIR
Alarm! PIR
Set
Set
Set
Set
Set
Set
```

If you get the word "Error", then the circuit is reading analog values outside of those we found on the bench. Go back to loading the code with DEBUG to record the values for each open/close combination and you will probably find one or more values that changed due to resistance in the wire.

If you cannot get one or more sensors to "set up" (not alarm), for magnetic contacts adjust the magnet to trip when the door is closed. For the PIR, you must be still for the unit to set up.

Adjust the PIR's sensitivity and trip time again if necessary.

## Use

My circuit is monitored by my Bluetooth-enabled PC in the basement about 10 feet away (well within the 33 foot range of the Bluefruit). The PC monitors the stream for any text not "Set". You can write a Processing, Python or other program to connect through the Internet to alert if there is an alarm (left as an exercise for the Maker).

Be careful if you think about directly having your alarm alert the authorities. In many jurisdictions, alarm systems have to be registered and if there are nuisance alarms, you can be fined. You do not want the police coming for every small animal that might wander into your alarmed space.

In the next section, we'll discuss annunciating circuits that are more complex.

---

## Going Further

This tutorial provides many different possibilities in building an alarm system. It is intended for you to configure the project to your own needs.

### Fancier Annunciation

Internet: The Trinket cannot talk to the Internet directly, there are not enough pins and program memory. You can use the Trinket to communicate with another microcontroller or computer to provide Internet communications. For example, a [Processing \(\)](#) sketch on a computer running Windows/MacOS/Linux could read the alarm state via Bluetooth serial and communicate out to the Internet. There are also many programs demonstrating an Arduino Uno with Ethernet Shield talking to the Internet. A simple Uno sketch to get the Trinket alarm state via a Bluetooth (say an Adafruit Bluefruit EZ-Link shield) receiver and send it to a web page, or service such as xively.com or IFTTT.com via an Ethernet Shield.

Noise: A Piezo buzzer will alert you to an alarm. A loudspeaker would jar you out of bed. Note, if you do use sound, you will want some type of cutoff switch to mute the sound when you do not want it. Or you can modify the code to read from a serial connection and toggle a variable which allows / disallows the sound to be made. If you use a high current horn, connect to the Trinket pin via a transistor to switch a relay to control the horn line. If you connect high voltage or current directly to the

Trinket pin it will destroy the Trinket.

If you prefer not to use a Bluetooth connection, you can consider a hardwired connection to the monitor room. This could be in one of several forms - TTL serial direct from Pin 0 as configured. Or you could have a piezo speaker in the home to alert you like a smoke detector. Even a flashing light (preferable for those hearing impaired) via a PowerTail switch. Once you understand the basics, modifying to suit your own needs is relatively straightforward.