

## Halloween Hacks: Dracula Clyde

Friday, October 24, 2014



Clyde sits cloaked in the corner, and suddenly he comes to life! Add a motion sensor and a piezo speaker to Clyde to create spooky Halloween decor.

### Overview

#### Summary

In this tutorial you'll learn how to add piezo speaker and motion sensor to Clyde and how to program Clyde to play a song when the motion sensor is triggered.

This tutorial is presented in three steps (four, if you count costume-making). You'll start by adding the piezo speaker and then the motion sensor. Next, you'll program Clyde to play a scary tune, and turn on his lights when the motion sensor is triggered. As a finishing touch, you can make Clyde a little cloak and vampire teeth to complete the Dracula look. Let's get started!

#### The Steps

- Step 1: [Build the piezo speaker circuit](#)
- Step 2: [Connect the motion sensor to the board](#)
- Step 3: [Program Dracula Clyde behavior](#)
- Step 4: [Outfit Clyde in a cloak and fangs.](#)

#### What you'll need

- Clyde
- PIR Motion Sensor, <https://www.sparkfun.com/products/8630>
- Piezo Speaker, <https://www.sparkfun.com/products/7950>
- Mini breadboard, <https://www.sparkfun.com/products/12044>
- 1K ohm through hole resistors, brown-black-red-gold
- Female/male jumper wires, <http://www.adafruit.com/products/1954>
- Male/male jumper wires, <http://www.adafruit.com/products/1956>
- 22-24 AWG wire, 2 wires of about 17 inches each



Clyde is currently sold out, but the next improved litter is on the way!

**PRE-ORDER CLYDE V2 FROM \$135**

[Learn more about Clyde](#)

### Stay up to date.

We send occasional development news, approximately once a month. No spam, and we never share your info.

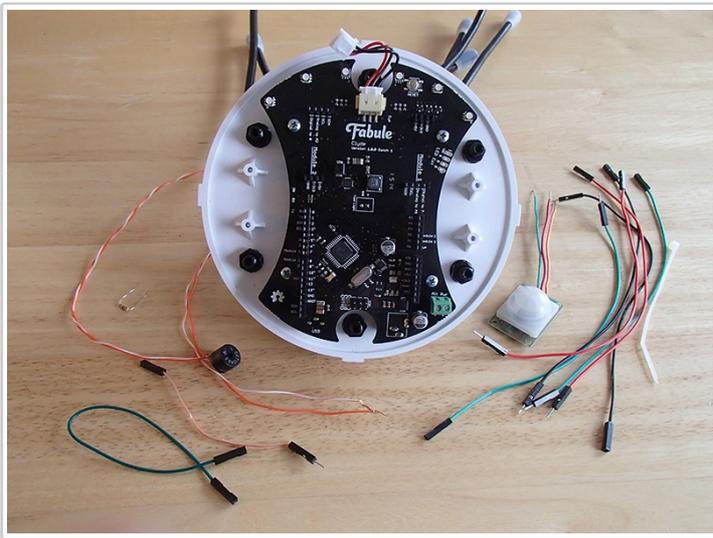
**SIGN UP**

Tags: [Clyde Hacks](#), [Halloween Hacks](#)

Related Posts:

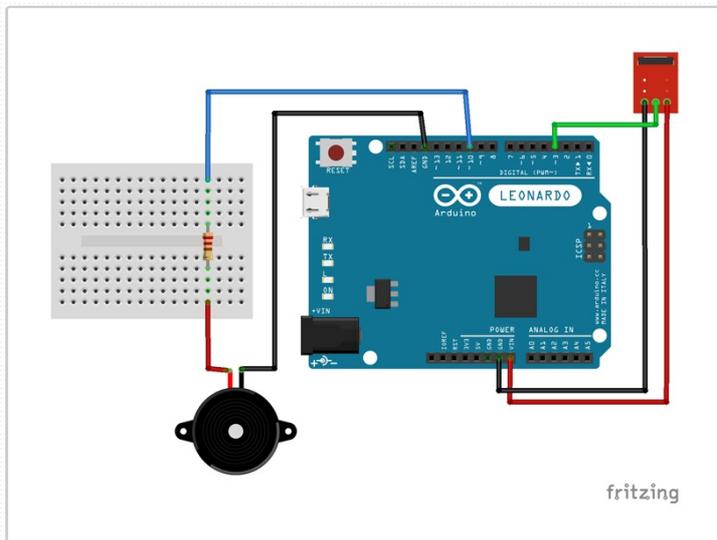
- [RobotGrrl Hack #1: Disco Clyde](#)
- [Clyde Halloween Costume Contest](#)
- [Halloween Hacks](#)
- [Halloween Hacks: Ballerina Clyde](#)
- [Halloween Hacks: Ghost Clyde](#)
- [RobotGrrl Hack #2: Clyde Halloween](#)
- [RobotGrrl Hack #3: Clyde Flower Mon...](#)
- [RobotGrrl Hack #4: Clyde the Pirate](#)



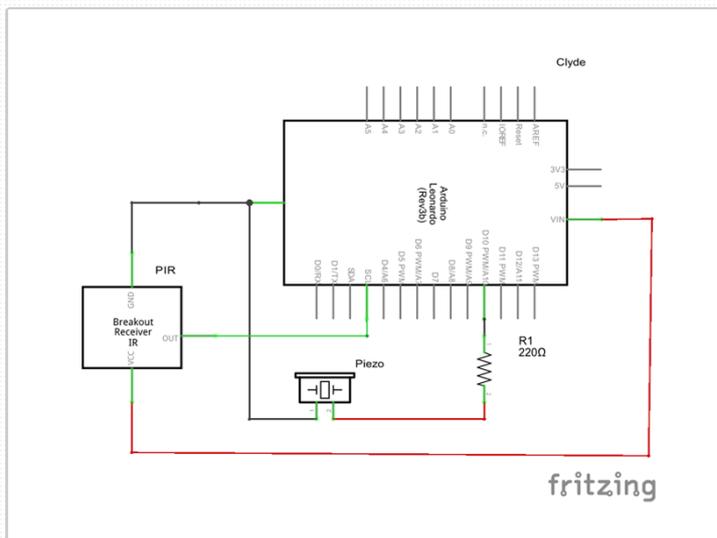


## Schematics and Diagrams

### Dracula Clyde Breadboard Diagram



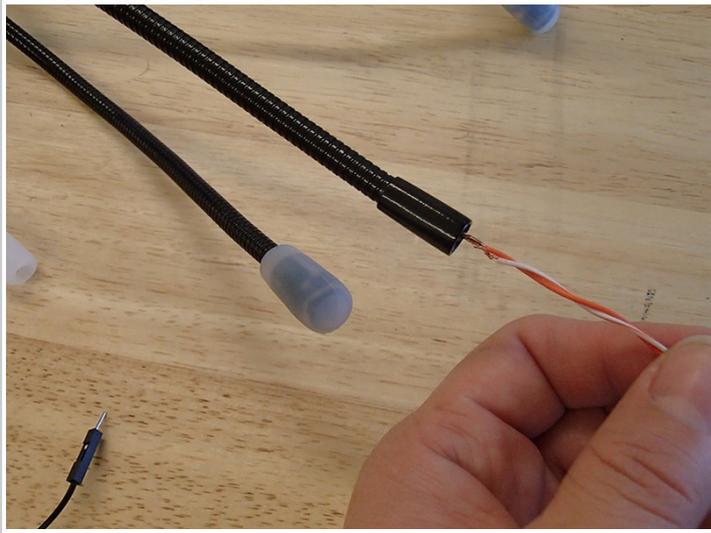
### Dracula Clyde Schematic Diagram



## Step 1: Build the piezo speaker circuit

In this step you will need the piezo speaker, the 22-24 AWG wires, female/male

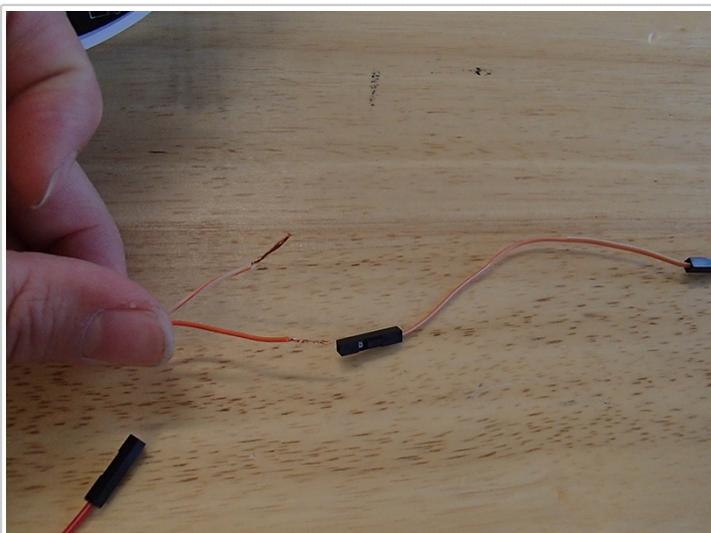
and male/male jumpers, the bread board and the resistor. Solder the two 17 inch wires to each of the piezo speaker leads. Wrap the 2 wires together and string the piezo speaker wires up Clyde's leg.



My piezo speaker sticks out of Clyde's leg a little bit.

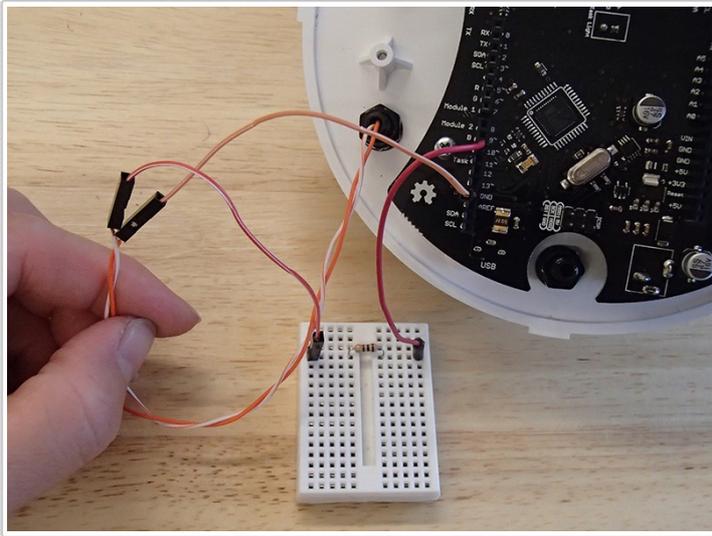


Connect the speaker leads to wire jumpers in order to have a nice stable connection with the breadboard.



Connect one of the leads to GND. The other connects to the breadboard, then to the resistor. The other lead of the resistor connects to pin 10. The piezo speaker

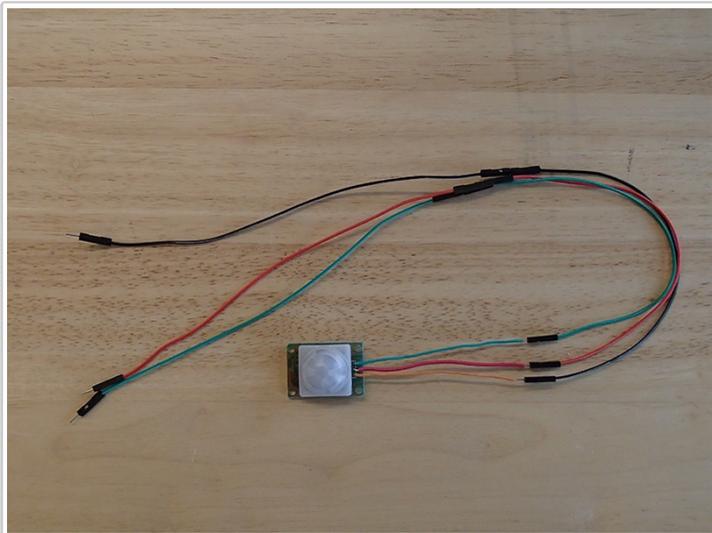
circuit is complete. Quite simple really.



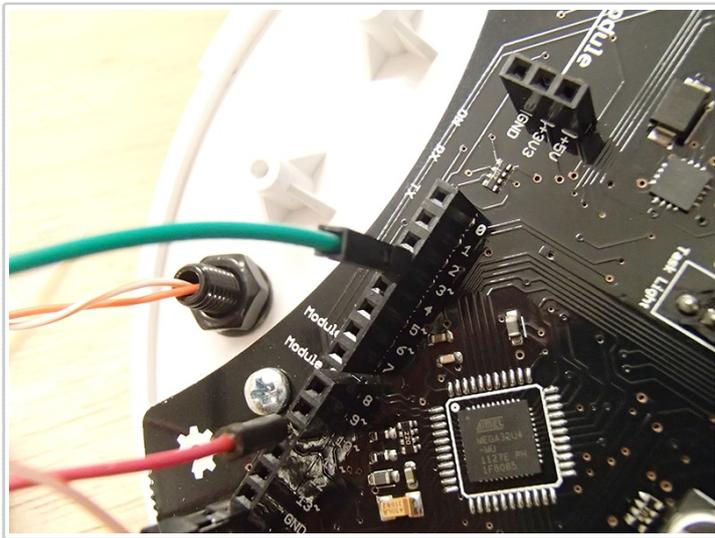
## Step 2: Connect the motion sensor to the board

The motion sensor does not require any additional electronic components to work with Clyde. It needs to be wired to Clyde's board. You'll need the PIR motion sensor and the female/male jumper wires.

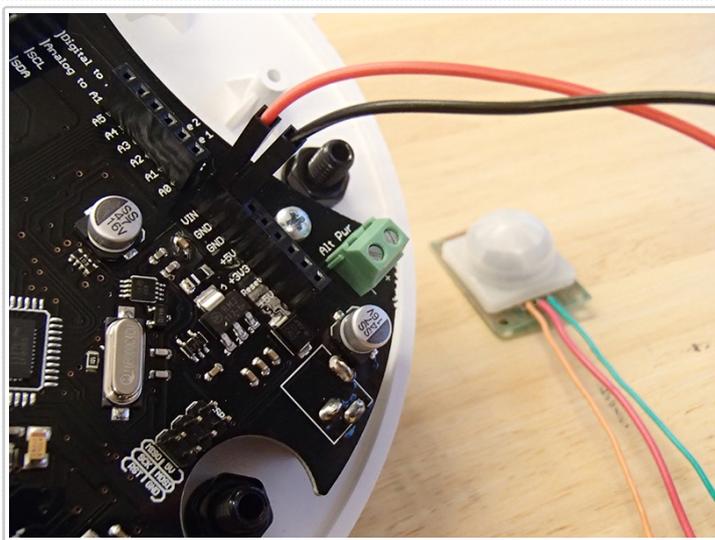
Connect the female/male jumper wires to the motion sensor.



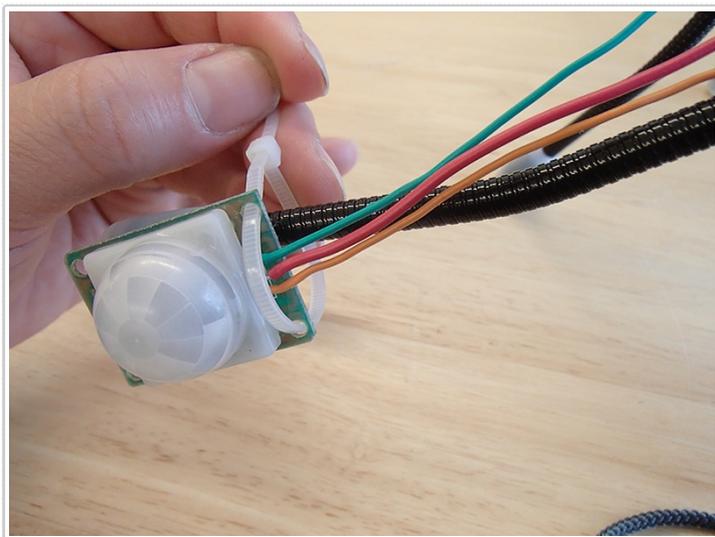
Connect the motion sensor's signal wire to pin 3.



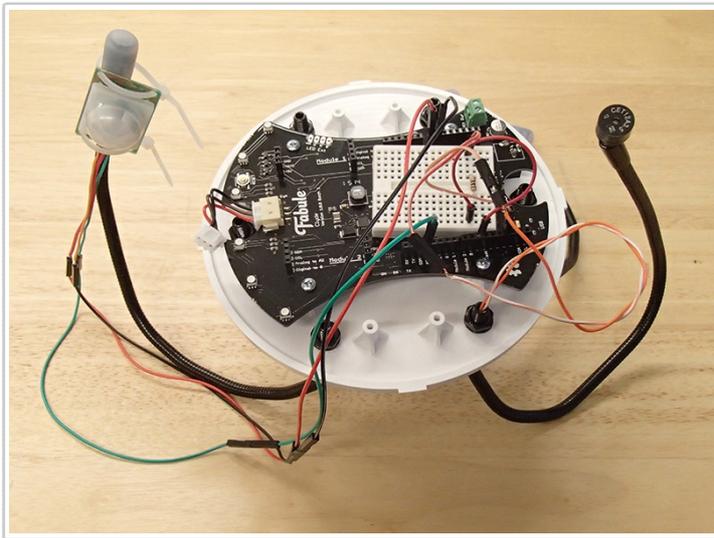
Connect the motion sensor's V+ wire to VIN pin, GND to GND



Attach the motion sensor to one of Clyde's feet with the zip tie.



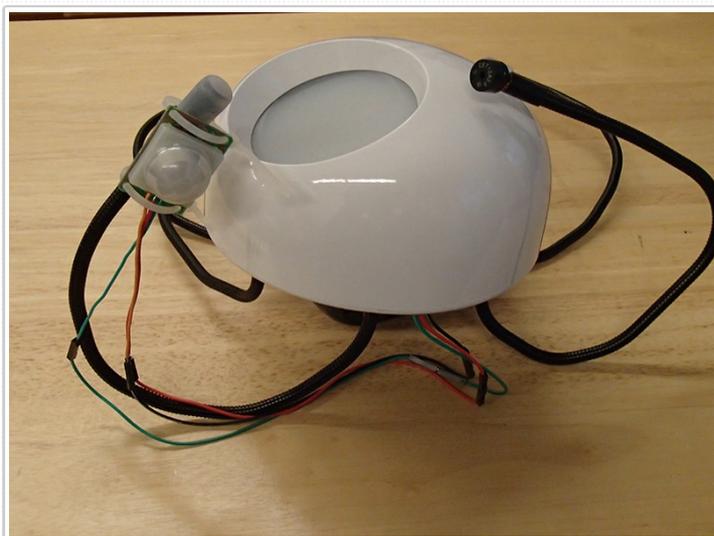
Ta-da! Time to for the finishing touches.



Carefully place Clyde's lid back onto the base with the wires for the motion sensor sticking out.



That's it for the electronics. You are now ready to move onto programming Clyde's behavior.



### Step 3: Program Dracula Clyde behavior

---

Create a new Arduino sketch, save it, call it DraculaClyde. Copy and paste the

following code into your Arduino IDE.

```
1  /*
2  Ghost Clyde
3  Halloween Hacks
4  Fabule Fabrications
5  by Angela Gabereau
6  Oct 21, 2014
7
8  Motion triggers the haunting, Clyde's eye light turns on, and the task light
9  and fan fade in and out several times, then the ghost goes silent again.
10
11  Arduino's basic fade example was the starting points of this code.
12  */
13  int taskLight = 11;          // the pin that Clyde's task light is attached to
14  int fan = 13;               // the pin that the fan is attached to
15  int R = 5, G = 6, B = 9;    // the pins for the red, green and blue of the eye light
16  int rColor = 255, gColor = 255, bColor = 255; // The color values for the eye light
17
18  int pirPin = 10;           //the digital pin connected to the PIR sensor's output
19  int calibrationTime = 30;  //the time we give the sensor to calibrate (1 second)
20  boolean haunting = false;  // should the task light and fan fade in and out
21  int spookCount = 0;        //The number of times the spooky light/fan cycle has run
22  int spookLimit = 3;       //The number of times the spooky light/fan cycle runs
23  int brightness = 0;       // how bright the LED is
24  int fadeAmount = 5;       // how many points to fade the LED by
25
26  // the setup routine runs once when you press reset:
27  void setup() {
28
29      //Initialize pins.
30
31      // Declare Clyde's task light pin to be an output:
32      pinMode(taskLight, OUTPUT);
33      // Turn off Clyde's task light by setting pin high.
34      digitalWrite(taskLight, HIGH);
35
36      // Declare fan pin to be an output:
37      pinMode(fan, OUTPUT);
38      // Turn off fan by setting pin low.
39      digitalWrite(fan, LOW);
40
41      // Declare R,G and B pins to be outputs, for Clyde's eye light.
42      pinMode(R, OUTPUT);
43      pinMode(G, OUTPUT);
44      pinMode(B, OUTPUT);
45      // Turn Clyde's eye light red to show that it is calibrating.
46      digitalWrite(R, 255);
47      digitalWrite(G, 0);
48      digitalWrite(B, 0);
49
50      calibrateMotionSensor();
51
52      // turn Clyde's eye light off to show that calibration is complete
53      digitalWrite(R, 0);
54      digitalWrite(G, 0);
55      digitalWrite(B, 0);
56  }
57
58  // The loop routine runs over and over again forever:
59  void loop() {
60
61      // Does Ghost Clyde perform his haunting behaviour?
62      if(haunting){
63
64          // Fade light and fan.
65          spookyClyde();
66
67          // Check if the spooky light and fan fading cycle has run enough times
68          if(spookCount>spookLimit){
69              // Who you gonna call?!
70              ghostbusters();
71          }
72      }else{
73          checkMotionSensor();
74      }
75  }
76
77
78
79  // Fade Clyde's task light and fan, turn on his eye light.
80  void spookyClyde(){
81
82      // Set Clyde's eye light to color
83      digitalWrite(R, rColor);
84      digitalWrite(G, gColor);
85      digitalWrite(B, bColor);
86
87      // Fade task light and fan
88      fade(taskLight, 0);
89      fade(fan, 255);
90
91      // Fade eye light
92      fade(R, 255);
93      fade(G, 0);
94      fade(B, 0);
95
96      // Wait for the fading to complete
97      delay(1000);
98
99      // Turn task light and fan on
100     digitalWrite(taskLight, HIGH);
101     digitalWrite(fan, LOW);
102
103     // Turn eye light on
104     digitalWrite(R, 255);
105     digitalWrite(G, 0);
106     digitalWrite(B, 0);
107
108     // Increment the spook count
109     spookCount++;
110
111     // Wait for the spook count to reach the limit
112     delay(1000);
113
114     // Turn haunting on
115     haunting = true;
116
117     // Wait for the haunting to complete
118     delay(1000);
119
120     // Turn haunting off
121     haunting = false;
122
123     // Wait for the haunting to complete
124     delay(1000);
125
126     // Turn task light and fan off
127     digitalWrite(taskLight, LOW);
128     digitalWrite(fan, HIGH);
129
130     // Turn eye light off
131     digitalWrite(R, 0);
132     digitalWrite(G, 0);
133     digitalWrite(B, 0);
134
135     // Wait for the task light and fan to fade
136     delay(1000);
137
138     // Turn task light and fan on
139     digitalWrite(taskLight, HIGH);
140     digitalWrite(fan, LOW);
141
142     // Turn eye light on
143     digitalWrite(R, 255);
144     digitalWrite(G, 0);
145     digitalWrite(B, 0);
146
147     // Increment the spook count
148     spookCount++;
149
150     // Wait for the spook count to reach the limit
151     delay(1000);
152
153     // Turn haunting on
154     haunting = true;
155
156     // Wait for the haunting to complete
157     delay(1000);
158
159     // Turn haunting off
160     haunting = false;
161
162     // Wait for the haunting to complete
163     delay(1000);
164
165     // Turn task light and fan off
166     digitalWrite(taskLight, LOW);
167     digitalWrite(fan, HIGH);
168
169     // Turn eye light off
170     digitalWrite(R, 0);
171     digitalWrite(G, 0);
172     digitalWrite(B, 0);
173
174     // Wait for the task light and fan to fade
175     delay(1000);
176
177     // Turn task light and fan on
178     digitalWrite(taskLight, HIGH);
179     digitalWrite(fan, LOW);
180
181     // Turn eye light on
182     digitalWrite(R, 255);
183     digitalWrite(G, 0);
184     digitalWrite(B, 0);
185
186     // Increment the spook count
187     spookCount++;
188
189     // Wait for the spook count to reach the limit
190     delay(1000);
191
192     // Turn haunting on
193     haunting = true;
194
195     // Wait for the haunting to complete
196     delay(1000);
197
198     // Turn haunting off
199     haunting = false;
200
201     // Wait for the haunting to complete
202     delay(1000);
203
204     // Turn task light and fan off
205     digitalWrite(taskLight, LOW);
206     digitalWrite(fan, HIGH);
207
208     // Turn eye light off
209     digitalWrite(R, 0);
210     digitalWrite(G, 0);
211     digitalWrite(B, 0);
212
213     // Wait for the task light and fan to fade
214     delay(1000);
215
216     // Turn task light and fan on
217     digitalWrite(taskLight, HIGH);
218     digitalWrite(fan, LOW);
219
220     // Turn eye light on
221     digitalWrite(R, 255);
222     digitalWrite(G, 0);
223     digitalWrite(B, 0);
224
225     // Increment the spook count
226     spookCount++;
227
228     // Wait for the spook count to reach the limit
229     delay(1000);
230
231     // Turn haunting on
232     haunting = true;
233
234     // Wait for the haunting to complete
235     delay(1000);
236
237     // Turn haunting off
238     haunting = false;
239
240     // Wait for the haunting to complete
241     delay(1000);
242
243     // Turn task light and fan off
244     digitalWrite(taskLight, LOW);
245     digitalWrite(fan, HIGH);
246
247     // Turn eye light off
248     digitalWrite(R, 0);
249     digitalWrite(G, 0);
250     digitalWrite(B, 0);
251
252     // Wait for the task light and fan to fade
253     delay(1000);
254
255     // Turn task light and fan on
256     digitalWrite(taskLight, HIGH);
257     digitalWrite(fan, LOW);
258
259     // Turn eye light on
260     digitalWrite(R, 255);
261     digitalWrite(G, 0);
262     digitalWrite(B, 0);
263
264     // Increment the spook count
265     spookCount++;
266
267     // Wait for the spook count to reach the limit
268     delay(1000);
269
270     // Turn haunting on
271     haunting = true;
272
273     // Wait for the haunting to complete
274     delay(1000);
275
276     // Turn haunting off
277     haunting = false;
278
279     // Wait for the haunting to complete
280     delay(1000);
281
282     // Turn task light and fan off
283     digitalWrite(taskLight, LOW);
284     digitalWrite(fan, HIGH);
285
286     // Turn eye light off
287     digitalWrite(R, 0);
288     digitalWrite(G, 0);
289     digitalWrite(B, 0);
290
291     // Wait for the task light and fan to fade
292     delay(1000);
293
294     // Turn task light and fan on
295     digitalWrite(taskLight, HIGH);
296     digitalWrite(fan, LOW);
297
298     // Turn eye light on
299     digitalWrite(R, 255);
300     digitalWrite(G, 0);
301     digitalWrite(B, 0);
302
303     // Increment the spook count
304     spookCount++;
305
306     // Wait for the spook count to reach the limit
307     delay(1000);
308
309     // Turn haunting on
310     haunting = true;
311
312     // Wait for the haunting to complete
313     delay(1000);
314
315     // Turn haunting off
316     haunting = false;
317
318     // Wait for the haunting to complete
319     delay(1000);
320
321     // Turn task light and fan off
322     digitalWrite(taskLight, LOW);
323     digitalWrite(fan, HIGH);
324
325     // Turn eye light off
326     digitalWrite(R, 0);
327     digitalWrite(G, 0);
328     digitalWrite(B, 0);
329
330     // Wait for the task light and fan to fade
331     delay(1000);
332
333     // Turn task light and fan on
334     digitalWrite(taskLight, HIGH);
335     digitalWrite(fan, LOW);
336
337     // Turn eye light on
338     digitalWrite(R, 255);
339     digitalWrite(G, 0);
340     digitalWrite(B, 0);
341
342     // Increment the spook count
343     spookCount++;
344
345     // Wait for the spook count to reach the limit
346     delay(1000);
347
348     // Turn haunting on
349     haunting = true;
350
351     // Wait for the haunting to complete
352     delay(1000);
353
354     // Turn haunting off
355     haunting = false;
356
357     // Wait for the haunting to complete
358     delay(1000);
359
360     // Turn task light and fan off
361     digitalWrite(taskLight, LOW);
362     digitalWrite(fan, HIGH);
363
364     // Turn eye light off
365     digitalWrite(R, 0);
366     digitalWrite(G, 0);
367     digitalWrite(B, 0);
368
369     // Wait for the task light and fan to fade
370     delay(1000);
371
372     // Turn task light and fan on
373     digitalWrite(taskLight, HIGH);
374     digitalWrite(fan, LOW);
375
376     // Turn eye light on
377     digitalWrite(R, 255);
378     digitalWrite(G, 0);
379     digitalWrite(B, 0);
380
381     // Increment the spook count
382     spookCount++;
383
384     // Wait for the spook count to reach the limit
385     delay(1000);
386
387     // Turn haunting on
388     haunting = true;
389
390     // Wait for the haunting to complete
391     delay(1000);
392
393     // Turn haunting off
394     haunting = false;
395
396     // Wait for the haunting to complete
397     delay(1000);
398
399     // Turn task light and fan off
400     digitalWrite(taskLight, LOW);
401     digitalWrite(fan, HIGH);
402
403     // Turn eye light off
404     digitalWrite(R, 0);
405     digitalWrite(G, 0);
406     digitalWrite(B, 0);
407
408     // Wait for the task light and fan to fade
409     delay(1000);
410
411     // Turn task light and fan on
412     digitalWrite(taskLight, HIGH);
413     digitalWrite(fan, LOW);
414
415     // Turn eye light on
416     digitalWrite(R, 255);
417     digitalWrite(G, 0);
418     digitalWrite(B, 0);
419
420     // Increment the spook count
421     spookCount++;
422
423     // Wait for the spook count to reach the limit
424     delay(1000);
425
426     // Turn haunting on
427     haunting = true;
428
429     // Wait for the haunting to complete
430     delay(1000);
431
432     // Turn haunting off
433     haunting = false;
434
435     // Wait for the haunting to complete
436     delay(1000);
437
438     // Turn task light and fan off
439     digitalWrite(taskLight, LOW);
440     digitalWrite(fan, HIGH);
441
442     // Turn eye light off
443     digitalWrite(R, 0);
444     digitalWrite(G, 0);
445     digitalWrite(B, 0);
446
447     // Wait for the task light and fan to fade
448     delay(1000);
449
450     // Turn task light and fan on
451     digitalWrite(taskLight, HIGH);
452     digitalWrite(fan, LOW);
453
454     // Turn eye light on
455     digitalWrite(R, 255);
456     digitalWrite(G, 0);
457     digitalWrite(B, 0);
458
459     // Increment the spook count
460     spookCount++;
461
462     // Wait for the spook count to reach the limit
463     delay(1000);
464
465     // Turn haunting on
466     haunting = true;
467
468     // Wait for the haunting to complete
469     delay(1000);
470
471     // Turn haunting off
472     haunting = false;
473
474     // Wait for the haunting to complete
475     delay(1000);
476
477     // Turn task light and fan off
478     digitalWrite(taskLight, LOW);
479     digitalWrite(fan, HIGH);
480
481     // Turn eye light off
482     digitalWrite(R, 0);
483     digitalWrite(G, 0);
484     digitalWrite(B, 0);
485
486     // Wait for the task light and fan to fade
487     delay(1000);
488
489     // Turn task light and fan on
490     digitalWrite(taskLight, HIGH);
491     digitalWrite(fan, LOW);
492
493     // Turn eye light on
494     digitalWrite(R, 255);
495     digitalWrite(G, 0);
496     digitalWrite(B, 0);
497
498     // Increment the spook count
499     spookCount++;
500
501     // Wait for the spook count to reach the limit
502     delay(1000);
503
504     // Turn haunting on
505     haunting = true;
506
507     // Wait for the haunting to complete
508     delay(1000);
509
510     // Turn haunting off
511     haunting = false;
512
513     // Wait for the haunting to complete
514     delay(1000);
515
516     // Turn task light and fan off
517     digitalWrite(taskLight, LOW);
518     digitalWrite(fan, HIGH);
519
520     // Turn eye light off
521     digitalWrite(R, 0);
522     digitalWrite(G, 0);
523     digitalWrite(B, 0);
524
525     // Wait for the task light and fan to fade
526     delay(1000);
527
528     // Turn task light and fan on
529     digitalWrite(taskLight, HIGH);
530     digitalWrite(fan, LOW);
531
532     // Turn eye light on
533     digitalWrite(R, 255);
534     digitalWrite(G, 0);
535     digitalWrite(B, 0);
536
537     // Increment the spook count
538     spookCount++;
539
540     // Wait for the spook count to reach the limit
541     delay(1000);
542
543     // Turn haunting on
544     haunting = true;
545
546     // Wait for the haunting to complete
547     delay(1000);
548
549     // Turn haunting off
550     haunting = false;
551
552     // Wait for the haunting to complete
553     delay(1000);
554
555     // Turn task light and fan off
556     digitalWrite(taskLight, LOW);
557     digitalWrite(fan, HIGH);
558
559     // Turn eye light off
560     digitalWrite(R, 0);
561     digitalWrite(G, 0);
562     digitalWrite(B, 0);
563
564     // Wait for the task light and fan to fade
565     delay(1000);
566
567     // Turn task light and fan on
568     digitalWrite(taskLight, HIGH);
569     digitalWrite(fan, LOW);
570
571     // Turn eye light on
572     digitalWrite(R, 255);
573     digitalWrite(G, 0);
574     digitalWrite(B, 0);
575
576     // Increment the spook count
577     spookCount++;
578
579     // Wait for the spook count to reach the limit
580     delay(1000);
581
582     // Turn haunting on
583     haunting = true;
584
585     // Wait for the haunting to complete
586     delay(1000);
587
588     // Turn haunting off
589     haunting = false;
590
591     // Wait for the haunting to complete
592     delay(1000);
593
594     // Turn task light and fan off
595     digitalWrite(taskLight, LOW);
596     digitalWrite(fan, HIGH);
597
598     // Turn eye light off
599     digitalWrite(R, 0);
600     digitalWrite(G, 0);
601     digitalWrite(B, 0);
602
603     // Wait for the task light and fan to fade
604     delay(1000);
605
606     // Turn task light and fan on
607     digitalWrite(taskLight, HIGH);
608     digitalWrite(fan, LOW);
609
610     // Turn eye light on
611     digitalWrite(R, 255);
612     digitalWrite(G, 0);
613     digitalWrite(B, 0);
614
615     // Increment the spook count
616     spookCount++;
617
618     // Wait for the spook count to reach the limit
619     delay(1000);
620
621     // Turn haunting on
622     haunting = true;
623
624     // Wait for the haunting to complete
625     delay(1000);
626
627     // Turn haunting off
628     haunting = false;
629
630     // Wait for the haunting to complete
631     delay(1000);
632
633     // Turn task light and fan off
634     digitalWrite(taskLight, LOW);
635     digitalWrite(fan, HIGH);
636
637     // Turn eye light off
638     digitalWrite(R, 0);
639     digitalWrite(G, 0);
640     digitalWrite(B, 0);
641
642     // Wait for the task light and fan to fade
643     delay(1000);
644
645     // Turn task light and fan on
646     digitalWrite(taskLight, HIGH);
647     digitalWrite(fan, LOW);
648
649     // Turn eye light on
650     digitalWrite(R, 255);
651     digitalWrite(G, 0);
652     digitalWrite(B, 0);
653
654     // Increment the spook count
655     spookCount++;
656
657     // Wait for the spook count to reach the limit
658     delay(1000);
659
660     // Turn haunting on
661     haunting = true;
662
663     // Wait for the haunting to complete
664     delay(1000);
665
666     // Turn haunting off
667     haunting = false;
668
669     // Wait for the haunting to complete
670     delay(1000);
671
672     // Turn task light and fan off
673     digitalWrite(taskLight, LOW);
674     digitalWrite(fan, HIGH);
675
676     // Turn eye light off
677     digitalWrite(R, 0);
678     digitalWrite(G, 0);
679     digitalWrite(B, 0);
680
681     // Wait for the task light and fan to fade
682     delay(1000);
683
684     // Turn task light and fan on
685     digitalWrite(taskLight, HIGH);
686     digitalWrite(fan, LOW);
687
688     // Turn eye light on
689     digitalWrite(R, 255);
690     digitalWrite(G, 0);
691     digitalWrite(B, 0);
692
693     // Increment the spook count
694     spookCount++;
695
696     // Wait for the spook count to reach the limit
697     delay(1000);
698
699     // Turn haunting on
700     haunting = true;
701
702     // Wait for the haunting to complete
703     delay(1000);
704
705     // Turn haunting off
706     haunting = false;
707
708     // Wait for the haunting to complete
709     delay(1000);
710
711     // Turn task light and fan off
712     digitalWrite(taskLight, LOW);
713     digitalWrite(fan, HIGH);
714
715     // Turn eye light off
716     digitalWrite(R, 0);
717     digitalWrite(G, 0);
718     digitalWrite(B, 0);
719
720     // Wait for the task light and fan to fade
721     delay(1000);
722
723     // Turn task light and fan on
724     digitalWrite(taskLight, HIGH);
725     digitalWrite(fan, LOW);
726
727     // Turn eye light on
728     digitalWrite(R, 255);
729     digitalWrite(G, 0);
730     digitalWrite(B, 0);
731
732     // Increment the spook count
733     spookCount++;
734
735     // Wait for the spook count to reach the limit
736     delay(1000);
737
738     // Turn haunting on
739     haunting = true;
740
741     // Wait for the haunting to complete
742     delay(1000);
743
744     // Turn haunting off
745     haunting = false;
746
747     // Wait for the haunting to complete
748     delay(1000);
749
750     // Turn task light and fan off
751     digitalWrite(taskLight, LOW);
752     digitalWrite(fan, HIGH);
753
754     // Turn eye light off
755     digitalWrite(R, 0);
756     digitalWrite(G, 0);
757     digitalWrite(B, 0);
758
759     // Wait for the task light and fan to fade
760     delay(1000);
761
762     // Turn task light and fan on
763     digitalWrite(taskLight, HIGH);
764     digitalWrite(fan, LOW);
765
766     // Turn eye light on
767     digitalWrite(R, 255);
768     digitalWrite(G, 0);
769     digitalWrite(B, 0);
770
771     // Increment the spook count
772     spookCount++;
773
774     // Wait for the spook count to reach the limit
775     delay(1000);
776
777     // Turn haunting on
778     haunting = true;
779
780     // Wait for the haunting to complete
781     delay(1000);
782
783     // Turn haunting off
784     haunting = false;
785
786     // Wait for the haunting to complete
787     delay(1000);
788
789     // Turn task light and fan off
790     digitalWrite(taskLight, LOW);
791     digitalWrite(fan, HIGH);
792
793     // Turn eye light off
794     digitalWrite(R, 0);
795     digitalWrite(G, 0);
796     digitalWrite(B, 0);
797
798     // Wait for the task light and fan to fade
799     delay(1000);
800
801     // Turn task light and fan on
802     digitalWrite(taskLight, HIGH);
803     digitalWrite(fan, LOW);
804
805     // Turn eye light on
806     digitalWrite(R, 255);
807     digitalWrite(G, 0);
808     digitalWrite(B, 0);
809
810     // Increment the spook count
811     spookCount++;
812
813     // Wait for the spook count to reach the limit
814     delay(1000);
815
816     // Turn haunting on
817     haunting = true;
818
819     // Wait for the haunting to complete
820     delay(1000);
821
822     // Turn haunting off
823     haunting = false;
824
825     // Wait for the haunting to complete
826     delay(1000);
827
828     // Turn task light and fan off
829     digitalWrite(taskLight, LOW);
830     digitalWrite(fan, HIGH);
831
832     // Turn eye light off
833     digitalWrite(R, 0);
834     digitalWrite(G, 0);
835     digitalWrite(B, 0);
836
837     // Wait for the task light and fan to fade
838     delay(1000);
839
840     // Turn task light and fan on
841     digitalWrite(taskLight, HIGH);
842     digitalWrite(fan, LOW);
843
844     // Turn eye light on
845     digitalWrite(R, 255);
846     digitalWrite(G, 0);
847     digitalWrite(B, 0);
848
849     // Increment the spook count
850     spookCount++;
851
852     // Wait for the spook count to reach the limit
853     delay(1000);
854
855     // Turn haunting on
856     haunting = true;
857
858     // Wait for the haunting to complete
859     delay(1000);
860
861     // Turn haunting off
862     haunting = false;
863
864     // Wait for the haunting to complete
865     delay(1000);
866
867     // Turn task light and fan off
868     digitalWrite(taskLight, LOW);
869     digitalWrite(fan, HIGH);
870
871     // Turn eye light off
872     digitalWrite(R, 0);
873     digitalWrite(G, 0);
874     digitalWrite(B, 0);
875
876     // Wait for the task light and fan to fade
877     delay(1000);
878
879     // Turn task light and fan on
880     digitalWrite(taskLight, HIGH);
881     digitalWrite(fan, LOW);
882
883     // Turn eye light on
884     digitalWrite(R, 255);
885     digitalWrite(G, 0);
886     digitalWrite(B, 0);
887
888     // Increment the spook count
889     spookCount++;
890
891     // Wait for the spook count to reach the limit
892     delay(1000);
893
894     // Turn haunting on
895     haunting = true;
896
897     // Wait for the haunting to complete
898     delay(1000);
899
900     // Turn haunting off
901     haunting = false;
902
903     // Wait for the haunting to complete
904     delay(1000);
905
906     // Turn task light and fan off
907     digitalWrite(taskLight, LOW);
908     digitalWrite(fan, HIGH);
909
910     // Turn eye light off
911     digitalWrite(R, 0);
912     digitalWrite(G, 0);
913     digitalWrite(B, 0);
914
915     // Wait for the task light and fan to fade
916     delay(1000);
917
918     // Turn task light and fan on
919     digitalWrite(taskLight, HIGH);
920     digitalWrite(fan, LOW);
921
922     // Turn eye light on
923     digitalWrite(R, 255);
924     digitalWrite(G, 0);
925     digitalWrite(B, 0);
926
927     // Increment the spook count
928     spookCount++;
929
930     // Wait for the spook count to reach the limit
931     delay(1000);
932
933     // Turn haunting on
934     haunting = true;
935
936     // Wait for the haunting to complete
937     delay(1000);
938
939     // Turn haunting off
940     haunting = false;
941
942     // Wait for the haunting to complete
943     delay(1000);
944
945     // Turn task light and fan off
946     digitalWrite(taskLight, LOW);
947     digitalWrite(fan, HIGH);
948
949     // Turn eye light off
950     digitalWrite(R, 0);
951     digitalWrite(G, 0);
952     digitalWrite(B, 0);
953
954     // Wait for the task light and fan to fade
955     delay(1000);
956
957     // Turn task light and fan on
958     digitalWrite(taskLight, HIGH);
959     digitalWrite(fan, LOW);
960
961     // Turn eye light on
962     digitalWrite(R, 255);
963     digitalWrite(G, 0);
964     digitalWrite(B, 0);
965
966     // Increment the spook count
967     spookCount++;
968
969     // Wait for the spook count to reach the limit
970     delay(1000);
971
972     // Turn haunting on
973     haunting = true;
974
975     // Wait for the haunting to complete
976     delay(1000);
977
978     // Turn haunting off
979     haunting = false;
980
981     // Wait for the haunting to complete
982     delay(1000);
983
984     // Turn task light and fan off
985     digitalWrite(taskLight, LOW);
986     digitalWrite(fan, HIGH);
987
988     // Turn eye light off
989     digitalWrite(R, 0);
990     digitalWrite(G, 0);
991     digitalWrite(B, 0);
992
993     // Wait for the task light and fan to fade
994     delay(1000);
995
996     // Turn task light and fan on
997     digitalWrite(taskLight, HIGH);
998     digitalWrite(fan, LOW);
999
1000    // Turn eye light on
1001    digitalWrite(R, 255);
1002    digitalWrite(G, 0);
1003    digitalWrite(B, 0);
1004
1005    // Increment the spook count
1006    spookCount++;
1007
1008    // Wait for the spook count to reach the limit
1009    delay(1000);
1010
1011    // Turn haunting on
1012    haunting = true;
1013
1014    // Wait for the haunting to complete
1015    delay(1000);
1016
1017    // Turn haunting off
1018    haunting = false;
1019
1020    // Wait for the haunting to complete
1021    delay(1000);
1022
1023    // Turn task light and fan off
1024    digitalWrite(taskLight, LOW);
1025    digitalWrite(fan, HIGH);
1026
1027    // Turn eye light off
1028    digitalWrite(R, 0);
1029    digitalWrite(G, 0);
1030    digitalWrite(B, 0);
1031
1032    // Wait for the task light and fan to fade
1033    delay(1000);
1034
1035    // Turn task light and fan on
1036    digitalWrite(taskLight, HIGH);
1037    digitalWrite(fan, LOW);
1038
1039    // Turn eye light on
1040    digitalWrite(R, 255);
1041    digitalWrite(G, 0);
1042    digitalWrite(B, 0);
1043
1044    // Increment the spook count
1045    spookCount++;
1046
1047    // Wait for the spook count to reach the limit
1048    delay(1000);
1049
1050    // Turn haunting on
1051    haunting = true;
1052
1053    // Wait for the haunting to complete
1054    delay(1000);
1055
1056    // Turn haunting off
1057    haunting = false;
1058
1059    // Wait for the haunting to complete
1060    delay(1000);
1061
1062    // Turn task light and fan off
1063    digitalWrite(taskLight, LOW);
1064    digitalWrite(fan, HIGH);
1065
1066    // Turn eye light off
1067    digitalWrite(R, 0);
1068    digitalWrite(G, 0);
1069    digitalWrite(B, 0);
1070
1071    // Wait for the task light and fan to fade
1072    delay(1000);
1073
1074    // Turn task light and fan on
1075    digitalWrite(taskLight, HIGH);
1076    digitalWrite(fan, LOW);
1077
1078    // Turn eye light on
1079    digitalWrite(R, 255);
1080    digitalWrite(G, 0);
1081    digitalWrite(B, 0);
1082
1083    // Increment the spook count
1084    spookCount++;
1085
1086    // Wait for the spook count to reach the limit
1087    delay(1000);
1088
1089    // Turn haunting on
1090    haunting = true;
1091
1092    // Wait for the haunting to complete
1093    delay(1000);
1094
1095    // Turn haunting off
1096    haunting = false;
1097
1098    // Wait for the haunting to complete
1099    delay(1000);
1100
1101    // Turn task light and fan off
1102    digitalWrite(taskLight, LOW);
1103    digitalWrite(fan, HIGH);
1104
1105    // Turn eye light off
1106    digitalWrite(R, 0);
1107    digitalWrite(G, 0);
1108    digitalWrite(B, 0);
1109
1110    // Wait for the task light and fan to fade
1111    delay(1000);
1112
1113    // Turn task light and fan on
1114    digitalWrite(taskLight, HIGH);
1115    digitalWrite(fan, LOW);
1116
1117    // Turn eye light on
1118    digitalWrite(R, 255);
1119    digitalWrite(G, 0);
1120    digitalWrite(B, 0);
1121
1122    // Increment the spook count
1123    spookCount++;
1124
1125    // Wait for the spook count to reach the limit
1126    delay(1000);
1127
1128    // Turn haunting on
1129    haunting = true;
1130
1131    // Wait for the haunting to complete
1132    delay(1000);
1133
1134    // Turn haunting off
1135    haunting = false;
1136
1137    // Wait for the haunting to complete
1138    delay(1000);
1139
1140    // Turn task light and fan off
1141    digitalWrite(taskLight, LOW);
1142    digitalWrite(fan, HIGH);
1143
1144    // Turn eye light off
1145    digitalWrite(R, 0);
1146    digitalWrite(G, 0);
1147    digitalWrite(B, 0);
1148
1149    // Wait for the task light and fan to fade
1150    delay(1000);
1151
1152    // Turn task light and fan on
1153    digitalWrite(taskLight, HIGH);
1154    digitalWrite(fan, LOW);
1155
1156    // Turn eye light on
1157    digitalWrite(R, 255);
1158    digitalWrite(G, 0);
1159    digitalWrite(B, 0);
1160
1161    // Increment the spook count
1162    spookCount++;
1163
1164    // Wait for the spook count to reach the limit
1165    delay(1000);
1166
1167    // Turn haunting on
1168    haunting = true;
1169
1170    // Wait for the haunting to complete
1171    delay(1000);
1172
1173    // Turn haunting off
1174    haunting = false;
1175
1176    // Wait for the haunting to complete
1177    delay(1000);
1178
1179    // Turn task light and fan off
1180    digitalWrite(taskLight, LOW);
1181    digitalWrite(fan, HIGH);
1182
1183    // Turn eye light off
1184    digitalWrite(R, 0);
1185    digitalWrite(G, 0);
1186    digitalWrite(B, 0);
1187
1188    // Wait for the task light and fan to fade
1189    delay(1000);
1190
1191    // Turn task light and fan on
1192    digitalWrite(taskLight, HIGH);
1193    digitalWrite(fan, LOW);
1194
1195    // Turn eye light on
1196    digitalWrite(R, 255);
1197    digitalWrite(G, 0);
1198    digitalWrite(B, 0);
1199
1200    // Increment the spook count
1201    spookCount++;
1202
1203    // Wait for the spook count to reach the limit
1204    delay(1000);
1205
1206    // Turn haunting on
1207    haunting = true;
1208
1209    // Wait for the haunting to complete
1210    delay(1000);
1211
1212    // Turn haunting off
1213    haunting = false;
1214
1215    // Wait for the haunting to complete
1216    delay(1000);
1217
1218    // Turn task light and fan off
1219    digitalWrite(taskLight, LOW);
1220    digitalWrite(fan, HIGH);
1221
1222    // Turn eye light off
1223    digitalWrite(R, 0);
1224    digitalWrite(G, 0);
1225    digitalWrite(B, 0);
1226
1227    // Wait for the task light and fan to fade
1228    delay(1000);
1229
1230    // Turn task light and fan on
1231    digitalWrite(taskLight, HIGH);
1232    digitalWrite(fan, LOW);
1233
1234    // Turn eye light on
1235    digitalWrite(R, 255);
1236    digitalWrite(G, 0);
1237    digitalWrite(B, 0);
1238
1239    // Increment the spook count
1240    spookCount++;
1241
1242    // Wait for the spook count to reach the limit
1243    delay(1000);
1244
1245    // Turn haunting on
1246    haunting = true;
1247
1248    // Wait for the haunting to complete
1249    delay(1000);
1250
1251    // Turn haunting off
1252    haunting = false;
1253
1254    // Wait for the haunting to complete
1255    delay(1000);
1256
1257    // Turn task light and fan off
1258    digitalWrite(taskLight, LOW);
1259    digitalWrite(fan, HIGH);
1260
1261    // Turn eye light off
1262    digitalWrite(R, 0);
1263    digitalWrite(G, 0);
1264    digitalWrite(B, 0);
1265
1266    // Wait for the task light and fan to fade
1267    delay(1000);
1268
1269    // Turn task light and fan on
1270    digitalWrite(taskLight, HIGH);
1271    digitalWrite(fan, LOW);
1272
1273    // Turn eye light on
1274    digitalWrite(R, 255);
1275    digitalWrite(G, 0);
1276    digitalWrite(B, 0);
1277
1278    // Increment the spook count
1279    spookCount++;
1280
1281    // Wait for the spook count to reach the limit
1282    delay(1000);
1283
1284    // Turn haunting on
1285    haunting = true;
1286
1287    // Wait for the haunting to complete
1288    delay(1000);
1289
1290    // Turn haunting off
1291    haunting = false;
```

```

84  digitalWrite(G, gColor);
85  digitalWrite(B, bColor);
86
87  // Set the brightness of the task light:
88  analogWrite(taskLight, 255-brightness);
89  // Set the brightness of the fan:
90  analogWrite(fan, brightness);
91
92  // Change the brightness for next time through the loop:
93  brightness = brightness + fadeAmount;
94
95  // Reverse the direction of the fading at the ends of the fade:
96  if (brightness == 0 || brightness == 255) {
97    fadeAmount = -fadeAmount ;
98  }
99
100 //A moment of darkness.
101 if (brightness == 0){
102   delay(1000);
103   spookCount++;
104 }
105
106 // Wait for 30 milliseconds to see the dimming effect
107 delay(30);
108 }
109
110 // Turn off all haunting behavior .
111 void ghostbusters(){
112
113   // Reset haunting values.
114   haunting = false;
115   spookCount = 0;
116
117   // Turn Clyde's eye light off
118   digitalWrite(R, 0);
119   digitalWrite(G, 0);
120   digitalWrite(B, 0);
121
122   // Turn off Clyde by setting pin high.
123   digitalWrite(taskLight, HIGH);
124
125   // Turn off fan by setting pin low.
126   digitalWrite(fan, LOW);
127 }
128
129 //Wait a little while so that the motion sensor can calibrate.
130 void calibrateMotionSensor(){
131
132   for(int i = 0; i < calibrationTime; i++){
133     delay(1000);
134   }
135   delay(50);
136 }
137
138 void checkMotionSensor(){
139   //If the PIR pin is high, trigger the haunting behavior .
140   if(digitalRead(pirPin) == HIGH){
141     haunting = true;
142   }
143 }

```

ClydeGhost hosted with ♥ by GitHub

[view raw](#)

The sketch uses Arduino's [Tone](#) example sketch as a base. This requires the file *pitches.h*. Open a new tab by clicking the arrow on upper right-hand of Arduino IDE, select "New Tab". Name it "pitches.h". Copy and paste the following:

```

/*****
 * Public Constants
 *****/

#define NOTE_B0 31
#define NOTE_C1 33
#define NOTE_CS1 35
#define NOTE_D1 37
#define NOTE_DS1 39
#define NOTE_E1 41
#define NOTE_F1 44

```

```
#define NOTE_FS1 46
#define NOTE_G1 49
#define NOTE_GS1 52
#define NOTE_A1 55
#define NOTE_AS1 58
#define NOTE_B1 62
#define NOTE_C2 65
#define NOTE_CS2 69
#define NOTE_D2 73
#define NOTE_DS2 78
#define NOTE_E2 82
#define NOTE_F2 87
#define NOTE_FS2 93
#define NOTE_G2 98
#define NOTE_GS2 104
#define NOTE_A2 110
#define NOTE_AS2 117
#define NOTE_B2 123
#define NOTE_C3 131
#define NOTE_CS3 139
#define NOTE_D3 147
#define NOTE_DS3 156
#define NOTE_E3 165
#define NOTE_F3 175
#define NOTE_FS3 185
#define NOTE_G3 196
#define NOTE_GS3 208
#define NOTE_A3 220
#define NOTE_AS3 233
#define NOTE_B3 247
#define NOTE_C4 262
#define NOTE_CS4 277
#define NOTE_D4 294
#define NOTE_DS4 311
#define NOTE_E4 330
#define NOTE_F4 349
#define NOTE_FS4 370
#define NOTE_G4 392
#define NOTE_GS4 415
#define NOTE_A4 440
#define NOTE_AS4 466
#define NOTE_B4 494
#define NOTE_C5 523
#define NOTE_CS5 554
#define NOTE_D5 587
#define NOTE_DS5 622
#define NOTE_E5 659
#define NOTE_F5 698
#define NOTE_FS5 740
#define NOTE_G5 784
#define NOTE_GS5 831
#define NOTE_A5 880
#define NOTE_AS5 932
#define NOTE_B5 988
#define NOTE_C6 1047
#define NOTE_CS6 1109
#define NOTE_D6 1175
#define NOTE_DS6 1245
#define NOTE_E6 1319
#define NOTE_F6 1397
#define NOTE_FS6 1480
#define NOTE_G6 1568
#define NOTE_GS6 1661
#define NOTE_A6 1760
#define NOTE_AS6 1865
#define NOTE_B6 1976
#define NOTE_C7 2093
#define NOTE_CS7 2217
```

```
#define NOTE_D7 2349
#define NOTE_DS7 2489
#define NOTE_E7 2637
#define NOTE_F7 2794
#define NOTE_FS7 2960
#define NOTE_G7 3136
#define NOTE_GS7 3322
#define NOTE_A7 3520
#define NOTE_AS7 3729
#define NOTE_B7 3951
#define NOTE_C8 4186
#define NOTE_CS8 4435
#define NOTE_D8 4699
#define NOTE_DS8 4978
```

Upload to Clyde and allow him to frighten you with a terrifying tune!

## Step 4: Outfit Clyde in a cloak and fangs.

Dracula Clyde is not complete without a cloak and fangs. I sewed a simple cloak using black and red fabric, and a bit of decorative string as a finishing touch. Here is a tutorial on how to make a cloak: Fast and Easy DIY Vampire Cape, <http://www.thebottletree.net/2012/10/30/fast-easy-diy-vampire-cape/>

I had to get creative for the fangs. Initially I was planning on drawing some fangs on sticker paper, and then sticking them to Clyde's head. But then I discover something even better! I found some bloody-tipped fake nails at the Halloween store. I cut the fake nails into points, filled them with removable adhesive putty, and then stuck them to Clyde's head. Viola! Terror! If you cannot find these fake nails, then you could make some yourself with either clear or white fake nails and a bit of red nail polish.

The final finishing touch is a bowl of candies placed inside Clyde's cloak. When the motion sensor is triggered, Clyde's task light turns on and reveals the bowl of candy. I chose candies that look like eyeballs!

Don't forget to share your Clyde Halloween Hacks with us! We have a Clyde Halloween Costume Contest until Nov 10th. Learn more about it here: <https://fabule.com/eng/blog/clyde-halloween-costume-contest>

Happy Halloween from the Fabule Team!



< Prev post: Halloween Hacks: Ghost Cl ...      Next post: RobotGrrl Hack #2: Clyde ... >

There are no comments on this post. [Add a comment](#)

## About Fabule

Founded by designers, Fabule makes unique domestic devices with a lot of personality. We are committed to creating smart products that make you feel smart. You can easily open, upgrade, tinker with or repair anything we design, and make it truly yours.

## Browse

[Home](#)  
[Blog](#)  
[Forum](#)  
[Press](#)

## Contact Us

[info@fabule.com](mailto:info@fabule.com)  
or use our [online form](#)

Fabule Fabrications Inc  
201-642 Rue de Courcelle,  
Montreal, Qc, Canada H4C  
3C5

