A Bewitched Book

by JohnShute

Bewitched Book

Intro:

Hello, this is my first Instructable. My project is a book that appears to be haunted. It will shake, laugh and glow green. The cover will try to open and sometimes the inside will glow eerily blue. I was inspired to make this project after watching a tutorial about how to make books look very old. I thought to myself, what if it was also haunted and did scary things every once in a while.

Please note; I was unhappy with the video so I have replaced it. Please be sure to turn up your volume to hear the evil laughter. Also the book didn't move or shake as much as I would like so I glued some plastic screw caps on the bottom. See last step for photo.

I get a lot of my parts from old electronics (printers, computers, radios, etc) that I take apart and scavenge the usable items from. Also, they can be found in electronic stores and surplus stores. The Arduino parts can be found on Amazon.

https://www.youtube.com/watch?v=QzN4UL-X4vA

Step 1: Gather the Parts

One old book...................Please don't use a classic. I found a 1929 Poetry reference book. It's great because I didn't need to age the pages.

One Arduino Nano

One ISD1820 Sound Module

One 8 ohm speaker...........The one that comes with the ISD1820 is not very good.

One Servo Motor

One Relay

One small 5 to 10 Volt DC motor

One weight......................I found mine in an old foot massager. Any small heavy piece of metal attached to the DC motor should work.

Three LEDs,.....................Two green and one blue.

One popsicle stick

Two small screws

Three resistors..................Three 220 ohm and One 1 meg ohm.

A peizo disk

One small on/off switch

Some Arduino wires

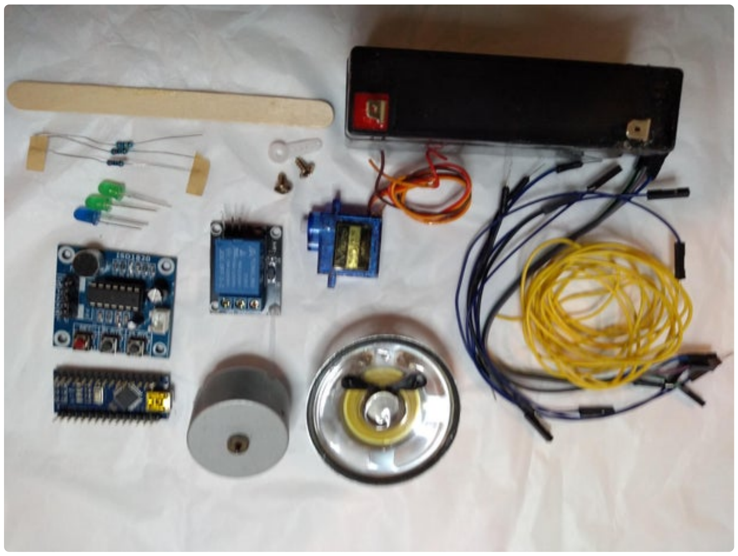
Some 14 guage wire

A small piece of 1/4" (6 mm) thick wood

A piece of clear plastic about 1/16" (2mm) thick..........I found in an inexpensive picture frame

Glue

Heat shrink and/or electrical tape



Step 2: Preparation

1. Hollow out your book. Leave about 25 pages at the top. The first 25 pages must remain uncut and unglued. Also leave about 1/2 inch (1.5cm) border or wall. I used a utility knife and a ruler to cut a rectangular hole. Go all the way to the back cover. My back cover wasn't very sturdy. I cut the piece of wood to the same size as my hole, drilled a hole in the wood the same size as the DC motor and glued the wood inside the back cover to fit in my rectangular hole.

2. The green glow is achieved with the clear plastic. Cut it into strips a wide as or wider than the sides of the book. Glue them in various places between the pages. Make sure they are flush on the outside. Now, paint a coat of clear glue over the outside of pages.

3. Setup the ISD1820. I used a laugh I found on line but you can do your own.

There is a great tutorial on how to set up and use the ISD1820 right here at Instructables.

https://www.instructables.com/id/HOW-TO-USE-ISD1820-VOICE-RECORDER-AND-PLAYER/-----

A hint to make it easier:

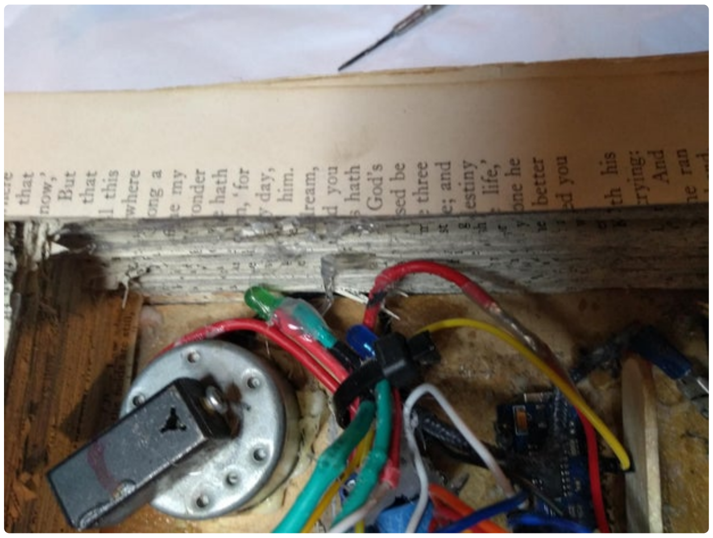
Attached the ISD1820 to the your Nano 5V and ground. Plug the Nano into your USB.Now play the sound you want to record close to the microphone on the ISD1820. It is now loaded in the memory of the recorder. When you load the Bewitched Book program it will m play the sound when the corresponding Nano pin goes high.

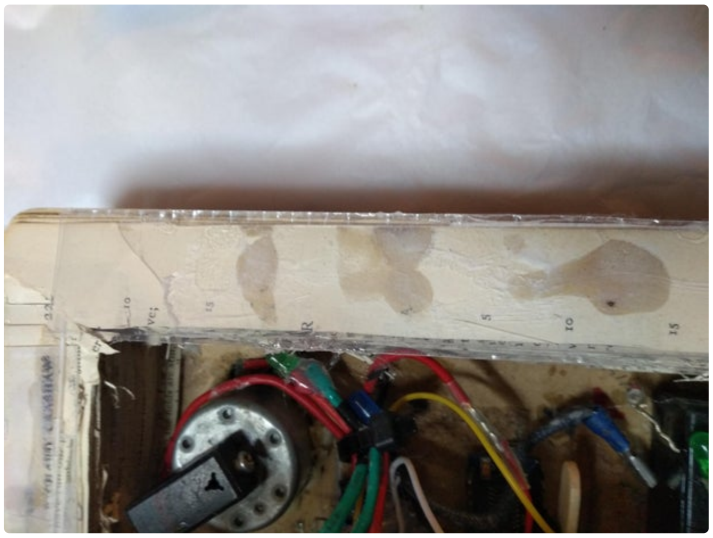
4. The Servo Motor comes with several small white arms. Drill two holes in the popsicle stick and screwed it to one of the arms. The white arm that only protrudes in one direction is best. Gently turn the servo motor arm clockwise until it stops. You could alternatively write a short program to set the servo at 0 degrees.

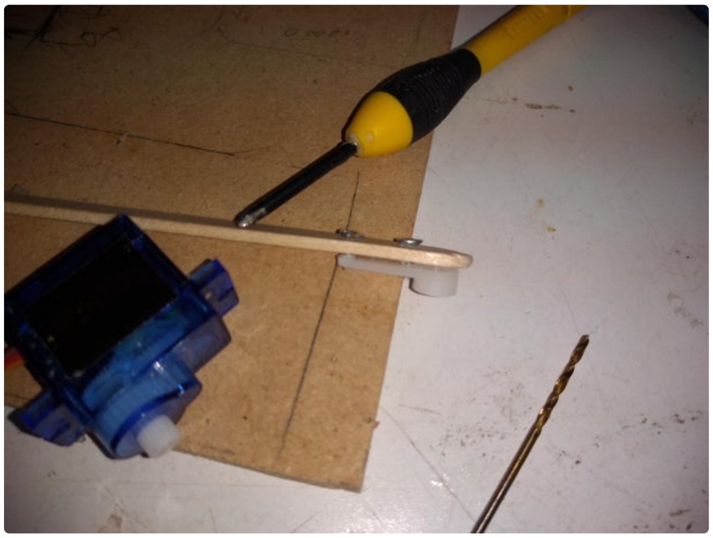
5. The Relay is for the DC motor. Connect the main power wire to the common connector and the DC motor positive wire to the N.O. (normally open) connector

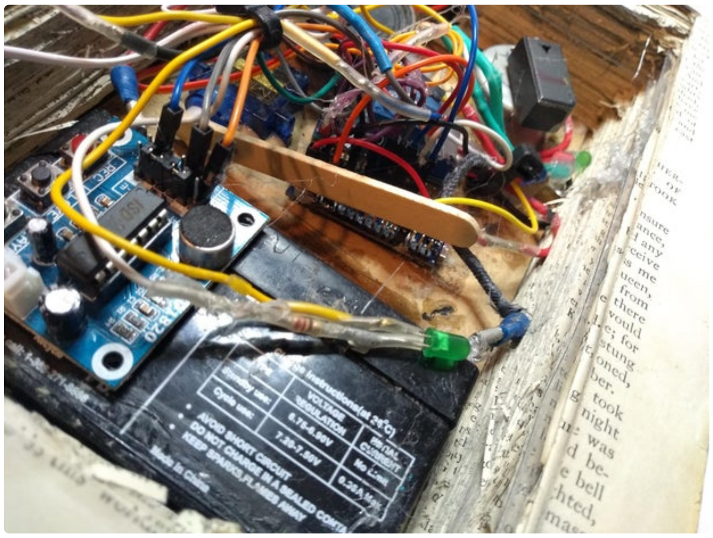
6. Solder the 220 ohm resistors and wiring to the LEDs as shown in the picture. Remember to cover any exposed wires with heat shrink or electrical tape.

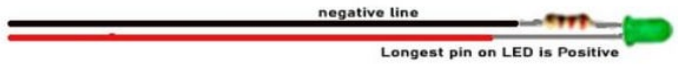
7. The Piezo disk should have a wire soldered to the brass colour and a wire soldered to the white center portion. One wire is attached to ground and the other to an analog pin of the Nano. The wires need to be long enough to go through the pages to the inside cover of the book. Also, the one Meg resistor must be soldered between the wires somewhere along the lines.











Step 3: Programming

. Due to the tight space, you may want to load the program onto the Nano before placing the Nano board into the book. You can check the comments in the code to change how much delays are between actions.

The code is Here:

In the video I have set the delay between actions to one second (1000 milliseconds) so that the video didn't run too long. Normally, it should be set around 30 to 60 seconds (30000 to 60000 milliseconds) to make it's actions more unexpected. You can see this in the code comments.

```
[code]
 /* Bewitched Book Sketch by John Shute (not a pro by any stretch of the imagination).  I declare
    this code to be public domain and I except no responsibility for anything.
    The code will control several functions intended to make a book appear to be haunted.
*/
#include <Servo.h>             //Library
Servo servo;                   //declare the servo motor
int pos = 0;                   //position variable
int i = 0;                     // count variable
int val = 0;                   //value for knock sensor
const int KnockSensor = A0;   //knock sensor
const int THRESHOLD = 350;    //How hard a Knock to start  Cases  (o to 1023)
int pwmLED = 5;                //Green Fading Leds
int pwmLED2 = 6;               //   "     "     "

void setup() {

  Serial.begin  (9600);       //for debugging
  pinMode(A0, INPUT);         // Knocksensor Pin
  pinMode(3, OUTPUT);         // Open book
  pinMode(4, OUTPUT);         // Rumble Motor
  pinMode(7, OUTPUT);         // Blue light
  pinMode(8, OUTPUT);         // Evil Laugh
  pinMode(pwmLED, OUTPUT);    // Green Fade (pin 5)
  pinMode(pwmLED2, OUTPUT);   // Green fade (pin 6)

}
```

```
}

void loop() {
  sensor();                          // check the sensor
  haunt();                           // start the program
}
void sensor() {
  val = analogRead(KnockSensor);     // read the knock sensor
  if (val < THRESHOLD) {             // if sensor not activated,
    sensor();                        // keep checking
  }
  else  {                            // if the sensor is disturbed
    haunt();                         // go to haunting actions
  }
}
void haunt() {                           //begin actions

  delay(5000);                           //wait 5 seconds

  //   Rumble Motor,  Relay to rumble motor turned on, book vibrates
  digitalWrite(4, HIGH);    //turn on the rumble motor
  delay(700);               //run 7 milliseconds
  digitalWrite(4, LOW);     // stop

  delay(20000);              //wait-----CHANGE THIS VALUE TO ADJUST THE DELAY BETWEEN ACTIONS.
                          //(1000); IS 1 SECOND.  (60000); IS 1 MINUTE).

  //  Evil Laugh,  recording plays
  digitalWrite(8, HIGH);                  // recording plays
  delay(3000);                            // time for recording to play
  digitalWrite(8, LOW);                   // stop

  delay(17000);                           //wait...The delay between actions.

  //   Pages Glow Green   Fade in, then out.
  for (int i = 0; i < 255; i++) {          //set the rate of fade in-increas i by 1
    analogWrite(pwmLED, i);                 //write the i value to pin 5
    analogWrite(pwmLED2, i);                //and 6
    delay(5);                              //time between iterations
    analogWrite (pwmLED,  HIGH);            //after 255 write high
    analogWrite (pwmLED2,  HIGH);           //
  }

  for (int i = 255; i > 0; i--) {           //decrease i by 1
    analogWrite(pwmLED, i);                  // see above
    analogWrite(pwmLED2, i);
    delay(5);
    analogWrite (pwmLED, LOW);
    analogWrite (pwmLED2,  LOW);

  }
  delay(20000);                              //wait...The delay between actions.

  //   Book Open and Close, blue LED on
  servo.attach(3);                         //turn the servo on
  for (pos = 0; pos <= 35; pos += 1) {     // goes from 0 degrees to 35 degrees in steps of 1 degree
    digitalWrite (7, HIGH);                // bue led on
    servo.write(pos);                      // tell servo to go to position in variable 'pos'
    delay(100);                            // set the speed it moves, relatively slow
  }
  for (pos = 35; pos >= 0; pos -= 1) {     // return to 0 degrees
    servo.write(pos);
    digitalWrite (7, LOW);                  // tell servo to go to position in variable 'pos'
    delay(15);                             // close quickly
  }
  servo.detach();                           // servo off

  delay(14000);                              //wait...The delay between actions.

  //   Rumble With Green light,
  digitalWrite(4, HIGH);   //turn on the rumble motor
  digitalWrite (pwmLED, HIGH);  //green light on too
  digitalWrite(pwmLED2, HIGH);
  delay(500);
  digitalWrite(4, LOW);      //turn them off
  digitalWrite(pwmLED, LOW);
  digitalWrite(pwmLED2, LOW);

  delay(21000);                         //wait...The delay between actions.


  //   Evil Laugh
  digitalWrite(8, HIGH);             // recording plays
  delay(3000);                       // time for recording to play
  digitalWrite(82, LOW);              // stop

  delay(20000);                          //wait...The delay between actions.

  //  Book Opens with no light
  servo.attach(3);
  for (pos = 0; pos <= 35; pos += 1) {   // goes from 0 degrees to 35 degrees  Steps in steps of 1 degree
    servo.write(pos);                    // tell servo to go to position in variable 'pos'
    delay(100);                          // go slowly in this direction

  }
  delay(500);
  for (pos = 35; pos >= 0; pos -= 1) {     // return to 0 degrees
    servo.write(pos);

    delay(15);                            // waits 15ms for the servo to reach the position

  }
  servo.detach();

  delay(19000);                            //wait...The delay between actions.

  //   Green Fade
  for (int i = 0; i < 255; i++) {
    analogWrite(pwmLED, i);                //write the i value to pin 5 and 8
    analogWrite(pwmLED2, i);
    delay(5);                            //wait 5 ms then do the for loop again
    analogWrite (pwmLED,  HIGH);
    analogWrite (pwmLED2,  HIGH);

  }
  for (int i = 255; i > 0; i--) {           //decrease i by 1
    analogWrite(pwmLED, i);
    analogWrite(pwmLED2, i);
    delay(5);
    analogWrite (pwmLED, LOW);
    analogWrite (pwmLED2,  LOW);
  }
  delay (20000);                           //wait...The delay between actions.

  //Rumble
  digitalWrite(4, HIGH);   //turn on the rumble motor
  delay(500);
  digitalWrite(4, LOW);

  delay(20000);                       //wait...The delay between actions.

  //  Evil Laugh
  digitalWrite(8, HIGH);             // recording plays
  delay(3000);                       // time for recording to play
  digitalWrite(8, LOW);              // stop

  delay (15000);                      //wait...The delay between actions.

  //  Book Open
  servo.attach(3);
  for (pos = 0; pos <= 35; pos += 1) {   // goes from 0 degrees to 35 degrees  Steps in steps of 1 degree
    servo.write(pos);                    // tell servo to go to position in variable 'pos'

    delay(100);                          // go slowly in this direction
  }
  delay(500);
  for (pos = 35; pos >= 0; pos -= 1) {     // return to 0 degrees
    servo.write(pos);

    delay(15);                            // waits 15ms for the servo to reach the position
  }
```

```
  servo.detach();

  delay(20000);                              //wait...The delay between actions.

  //Book Open with Blue Light and Laugh
  servo.attach(3);               //turn the servo on
  for (pos = 0; pos <= 35; pos += 1) {     // goes from 0 degrees to 35 degrees in steps of 1 degree
    digitalWrite (7, HIGH);
    digitalWrite (8, HIGH);
    servo.write(pos);                     // tell servo to go to position in variable 'pos'
    delay(100);
  }
  for (pos = 35; pos >= 0; pos -= 1) {     // return to 0 degrees
    servo.write(pos);
    digitalWrite (7, LOW);
    digitalWrite (8, LOW);                  // tell servo to go to position in variable 'pos'
    delay(15);
  }
  servo.detach();                         // turn the servo off

  delay(140000);                          //wait...The delay between actions.

  haunt();     //Go back to the Haunt loop
}
[/code]<br>
```

Step 4: Assembly and Wiring.

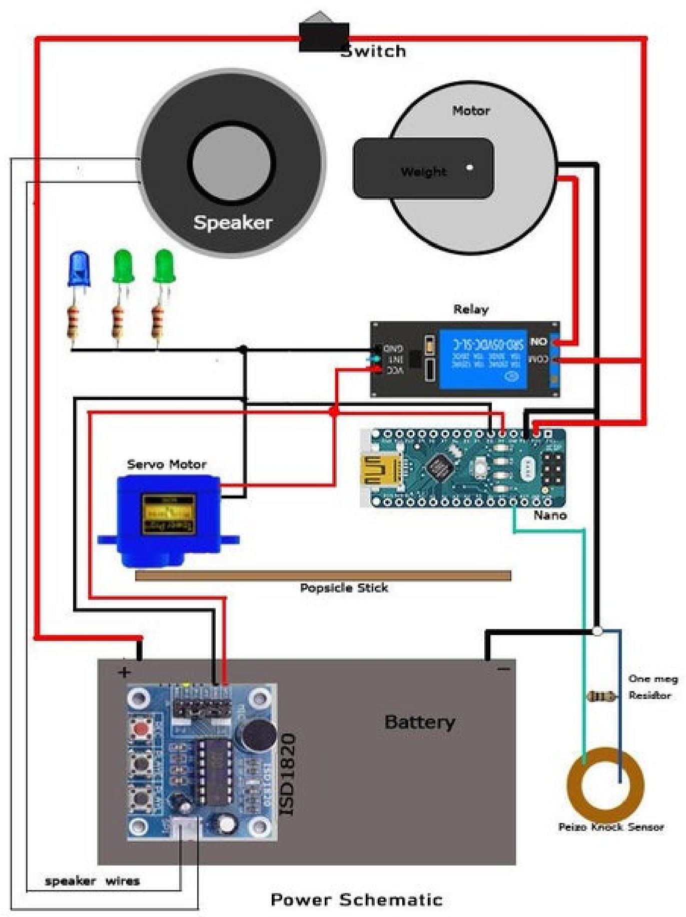
Step 4 Assembly and Wiring.

Glue the DC motor firmly into the hole drilled into the board. Insert the other parts. The wiring diagrams will illustrate how I placed the components. Glue them in, too. You need to cut a small slit in the top 25 pages to slide the Piezo disk through and tape the disk onto the back of the front cover. Cut a hole in the pages where it won't show and install the switch. Try to keep the LEDs close to the front of the book.

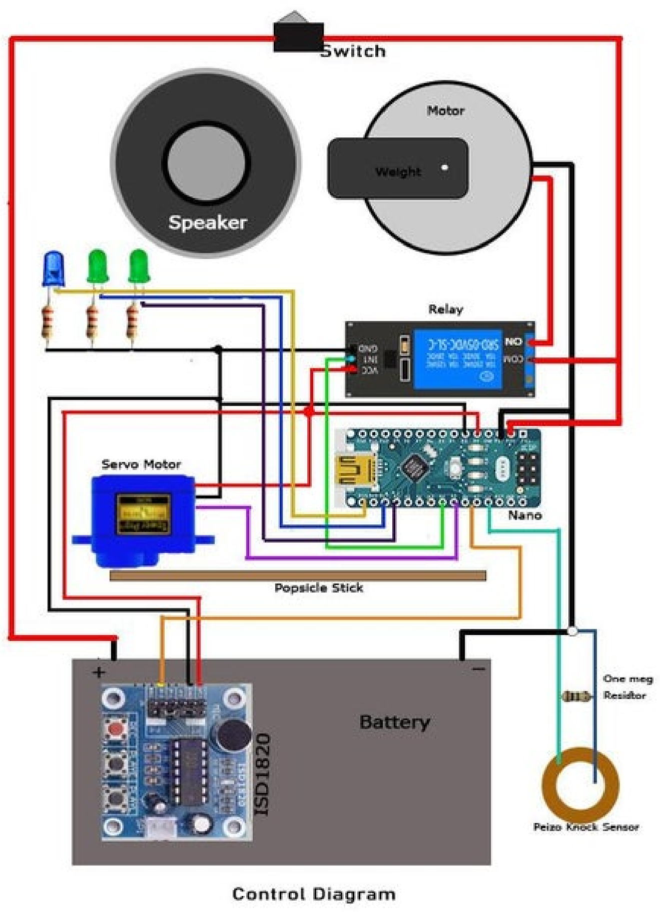
Wiring is a lot of work. I uploaded two diagrams

When wiring use the power line diagram first, then add the control lines.

The Nano board has a 5 volt regulator on the Vin pin so it and the DC motor are connected directly to the battery, through the on/off switch. All other components get 5 volts from The 5V pin on the Nano and are grounded to the Nano Gnd as well.



Power Schematic



Control Diagram

Step 5: Test and Adjustments

Once you turn the switch on the book may do a few actions such as the DC motor rumbles and the recorder plays. It takes a second or two for the Nano to start and control things. After this nothing should happen until you knock on the front cover. Then in five seconds the actions should begin.

You only want the servo motor to open the book slightly, otherwise people could see into your book. If it doesn't open the book at all you could adjust the code or, an easier way is as follows. Turn the book off, take the popsicle stick off the gears and place it back on at a higher point in its cycle.

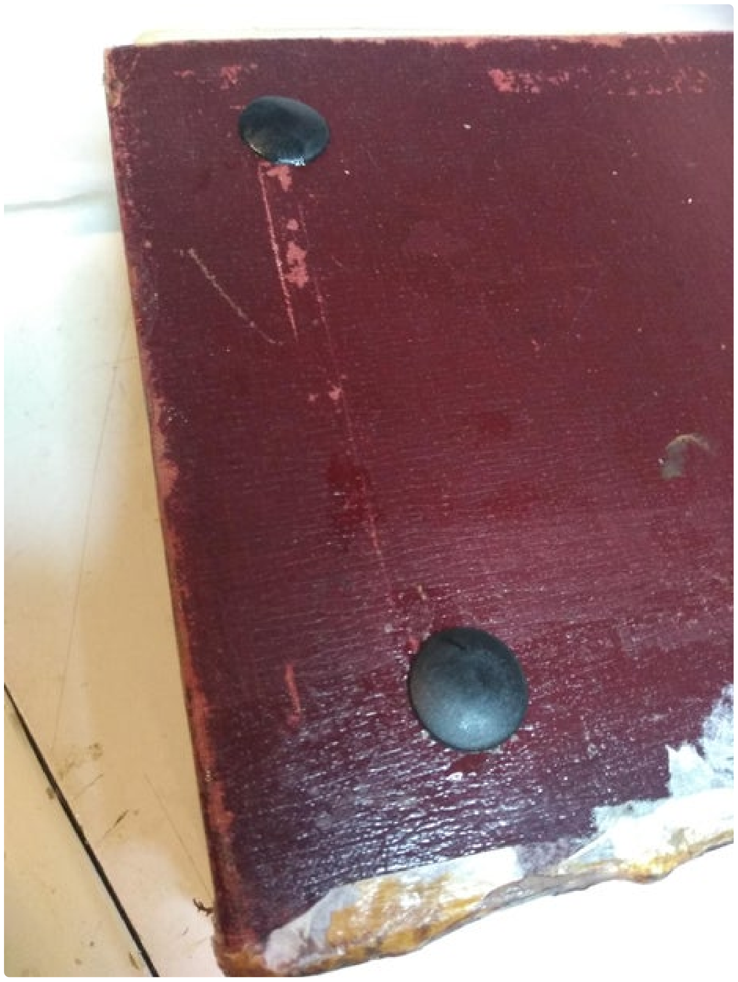
Step 6: Making It Look Old.

I glued some plastic screw caps on the back of the book so it would move better. See photo

Now we have to make our book look ancient. The best way I have found, and it's not to difficult, is to follow this lady's instructions, here.

rel="nofollow">https://www.youtube.com/watch?v=zPrkBor5TxM

So, I hope this project was as fun for you as it was for me. Also, I hope you get to startle a few friends and family this Halloween.



In your code, you are calling "sensor" recursively. This will lead to stack overflows. I would go for something like

```
void loop()
{
if sensor() // check the sensor
haunt(); // start the program
}

int sensor()
{
return (analogRead(KnockSensor) >= THRESHOLD) ;
}
```

Great instructable, BTW :-)

Hello again alfon1917
I tried the code you suggested. Verification returned: expected '(' before 'sensor'
I tried reading some tutorials, but I'm finding it hard to see what is wrong.

Well, you and your compiler are right, of course. The code should be:

```
void loop()
{
if (sensor()) // check the sensor
haunt(); // start the program
}

int sensor()
{
return (analogRead(KnockSensor) >= THRESHOLD) ;
}
```

Seems I've been away from the C language for quite some time...

This is what was wrong in my code. Regarding the issue in your code, a more detailed explanation might be useful:

Every time you call a function (in C and most procedural languages), the processor continues execution with that function. It needs to know where to return, however, when this function ends. To do this, it places the "return address" (i.e. the address of the first instruction after the function call) in a specified section of memory called the stack. When the function ends, the address is removed from the stack, the processor continues executing the code in this address and all is well. If before returning, the code in the called function calls another one, the new return address is "pushed" onto the stack, in the position just under the previous one. Again no problem: when the second called function ends, the second return address will be removed, then when the first called function ends the first return address will be removed. The problem is that the stack has a finite size, which in microcontrollers like the one used in Arduino is rather small. When you have recursive code (a function that calls itself, then calls itself again and again and again, this space will overflow. Although recursion is a legitimate coding practice (e.g. you could define the factorial of a number as this number multiplied by the factorial of the next lowest number), it must be used in a controlled way, which avoids its being used endlessly (e.g. in the factorial example, factorial(1) should return 1, not 1*factorial(0))otherwise the stack will be exhausted, and the "stack pointer" (the variable that shows the next position used for the stack) will either move outside the area designated for the stack and start overwriting program variables, or it will "wrap around" and start overwriting the first values pushed into the stack itself.

I hope the above makes sense. What it means in short is that you should avoid having a function call itself unless you are sure you are in full control of the situation.

Best regards,
Alf

You are a genius

Thanks, I'm glad you like it. I am no genius, though. BTW What did you mean by your father in the cone of shame? Who is your father?

That puts my father in the cone of shame

I wonder instead of the cover raising up the same level each time - albeit slowly - have a crankshaft with various levels the cover can open - unpredictable! I love the lights and the roughness of the Ol' Book!

Hello. What you ask could be done by changing the numbers in the servo part of the program. Under the the "Book open and closed" parts. The number 35 (two for each opening and closing) tells it how far to open. You could change these numbers, but if the book opens too much you will see the electronics inside. Have fun

Ah, yah, that could work too. Go with a random amount within the range of the opening!

A very spooky book!
Congratulations on your first instructable!
Welcome to the community!

Thanks

Any chance you still have the link to making a book look old?

Here you go.
https://www.youtube.com/watch?v=zPrkBor5TxM

Any possibility of making it voice activated, so that you can say a "magic word" and then the book starts trying to open?

That would me amazing if you could do it. Making it react to a voice command is beyond my knowledge. I can show you how to make it sound activated. That would be any sound at a specific volume would star it.

Well, you're whole project is light years beyond my ability, but I love what you did.

Really nice job putting this together :)

Thank you. I found it easier to do it in word, then copy and paste. I'm glad I stuck with it.

The effect is awesome! Well done.

Thanks. It was fun to make

My first instructable. Just in time for Halloween.