LASER Maze 2012 - Halloween Haunted House

by **bkhurt** on October 16, 2012

Table of Contents

Intro: LASER Maze 2012 - Halloween Haunted House

A new version of the LASER Maze for 2012!
(find the old version here: LASER Maze 2011)

What is it?
The LASER Maze 2012 is a movie-style laser "security system" that is set up in the front half of my garage haunted house. Trick-or-treaters must dodge a web of green laser beams to avoid setting off the lights and sirens that are protecting the candy in the back of the garage.

How does it work?
The LASER Maze uses an Arduino microcontroller and a PC running Processing. The Arduino is wired to 4 light sensors. Green handheld laser pointers are fixed to the garage wall and pointed at the light sensors on the opposite wall. When the laser beam is broken, the Arduino activates a flashing red light and sends a signal to the PC to play an alarm sound. The Arduino is also constantly sending light level readings to the PC, which are displayed on screen as bar graphs. This helps the LASER Maze operator monitor and calibrate the sensors.

How is the LASER Maze 2012 different from last years version?
The old 2011 version of the LASER Maze was designed as a game where players watch an intro video clip, hit a start button, navigate the maze, then hit the stop button to see their total time and score. The game aspect didn't work very well for a Halloween setting, because there was a constant flow of trick-or-treaters running through the maze. Also, the old 2011 version did not include any way to monitor your light sensors, which made troubleshooting difficult if things weren't working as expected.

The LASER Maze 2012 fixes some of the problems I experienced with the old version. It is now a continuously running system with no start or stop button. Time and number of lasers broken are no longer tracked, and there is no video clip (which makes setting it up easier!). Since there is no intro video, the PC monitor is now used to display data from the light sensors, which is extremely helpful for calibrating and troubleshooting the system.

One last note on safety before we get started. Please BE SMART if you build your own LASER Maze. Use low power lasers and keep them low to the ground to avoid eye injuries. Check out the Wikipedia entry on laser safety .

Step 1: Materials Needed

Here is everything you will need:
a garage
a laptop PC (with speakers or attached to an external sound system)
Arduino Duemilanove with USB cord
4 green Lasers ($10.87 each on Amazon)
4 CDS cells - these are the light sensors ($0.50 each at Electronic Goldmine)
4 10K resistors
1 breadboard
1 servo
1 surge protector with switch
1 red beacon light
1 1000W fog machine
phone cord long enough to run to your sensors (cheap at the thrift store)

You will also need electrical tape, cardboard TP tubes, soldering iron with soldier, velum paper or some other semi transparent paper, something to mount lasers and sensors to your wall, and possibly some other small items.

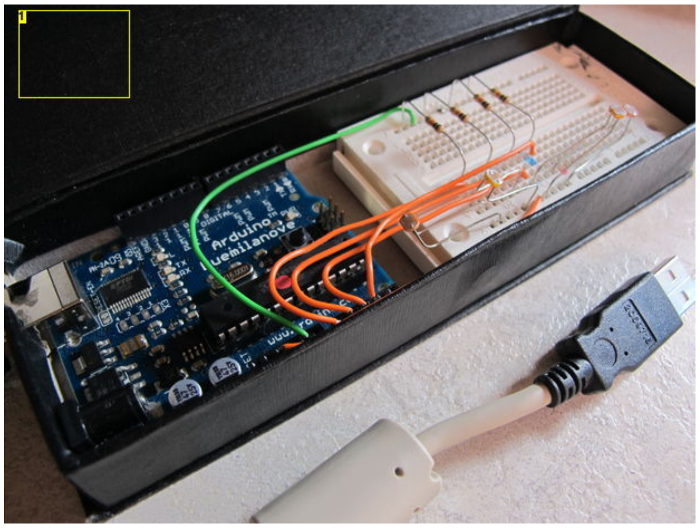


Image Notes
1. The Arduino and breadboard fit perfectly in one of the laser boxes.

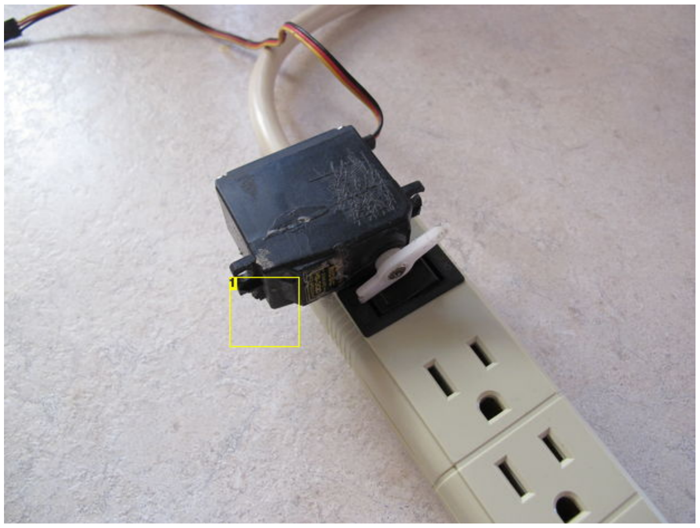


Image Notes
1. Servo hot glued to surge protector to power the spinning red beacon light

Step 2: Setting Up Your PC

I am including a lot of detail here for people that are not familiar with Arduino or Processing. If you already use an Arduino or Processing, these steps will be pretty easy.

1. Download Processing Here - I use version 1.5.1 Standard found under "Stable Releases"
2. Download Arduino IDE Here - I use version 1.0.1 for Windows
3. Download the zip file attached to this step (LASER_Maze_2012.zip at the bottom of this page). The zip contains the Processing code, Arduino Code, and 4 different alarm mp3 files.
4. Unzip the file and copy the 4 mp3 files to your c:\ drive.
   - The Processing program is hard coded to look for the mp3 files in c:\
5. Copy the LASER_Maze_2012_Arduino folder to your Arduino sketchbook folder
   - By default, the sketchbook folder is My Documents > Arduino
6. Copy LASER_Maze_2012_Processing folder to your Processing sketchbook folder
   - By default, the sketchbook folder is My Documents > Processing
7. Connect the Arduino to your PC with a USB cord
8. Open the Arduino IDE and upload the LASER_Maze_2012_Arduino.ino to your Arduino
   - Detailed steps for setting up your Arduino and uploading a program can be found HERE.
9. Open up Processing and load LASER_Maze_2012_Processing.pde
   - Detailed steps for setting up Processing and running a program can be found HERE.
10. Modify the COM port in the Processing code to the COM port that your Arduino is using.
    - Check Tools > Serial Port in your Arduino IDE to find the COM port Arduino is using
    - The line in Processing that you need to modify looks like this:
      - myPort = new Serial(this, Serial.list()[1], 9600);
11. Disconnect the USB cord from your Arduino. We will be building the circuit in the next step.

File Downloads

LASER_Maze_2012.zip (106 KB)
[NOTE: When saving, if you see .tmp as the file ext, rename it to 'LASER_Maze_2012.zip']

Step 3: Setting Up Your Arduino And Breadboard

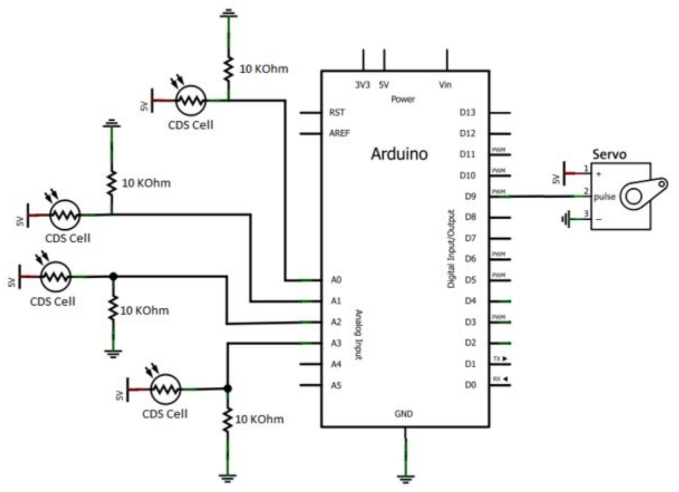
CDS Cells
The light sensors being used are actually called CDS cells or Light Dependant Resistors. They change resistance based on the amount of light hitting them. The sensors are very small, which makes it difficult to aim a laser at them from across the garage. To solve this problem, we put the CDS cell inside a carboard tube with velum paper or tissue paper covering the opening, and black tape covering the back side. The paper provides a larger target to aim the laser at, and it helps block out other ambient light and strobe lights in the haunted house.

Beacon Light
The Arduino turns on a 110V red police light when a laser beam is broken. To avoid interfacing the Arduino directly with 110V mains power, I hot glued a servo to a power strip. The Arduino activates the servo, which turns the switch on the power strip on or off.

Steps To Complete

1. Build the CDS cell enclosure as described above and shown in the picture
2. Hot glue your servo to your power strip so that the servo arms can turn the switch on or off
    - The amount the servo moves can be adjusted in the LASER_Maze_2012Arduino.ino code
3. Attach long wires(I use phone cord) to the CDS cell enclosure and servo so you can position them as needed in the garage
4. Assemble the circuit shown in the circuit diagram picture
5. Reconnect your Arduino to the PC via USB
6. Open up Processing on your PC and run the LASER_Maze_2012_Processing.pde program
7. If everything worked, you should see 4 bar graphs on the PC that show data from your 4 light sensors
8. If you press the reset button built in to the Arduino Duemilanove, the light sensors will take a new baseline reading and calculate new alarm trigger values. This is useful if your lighting conditions change or the sensors do not behave as expected.

The hard part is done! Now it's time to set everything up in your garage.



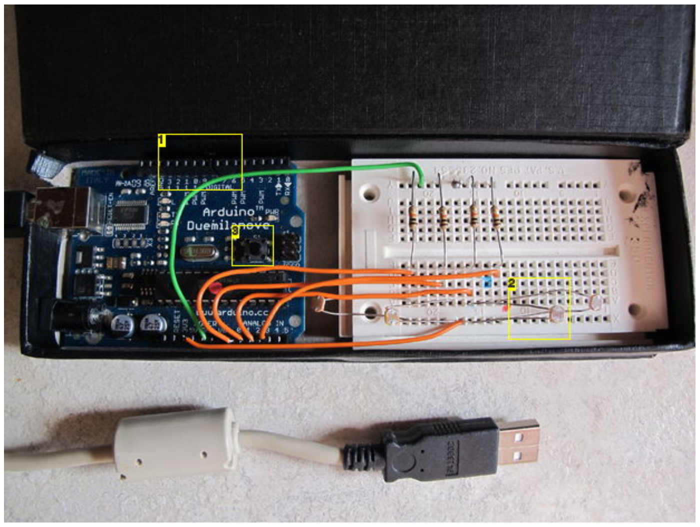
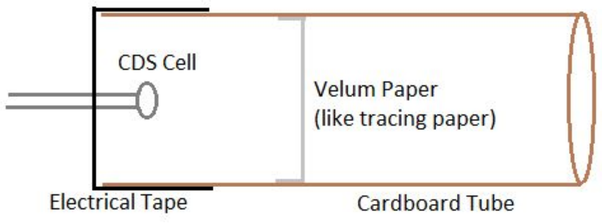
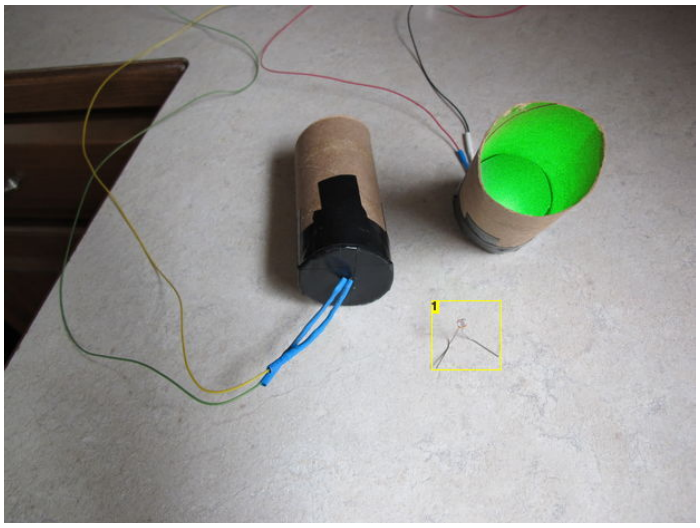


Image Notes
1. The servo isn't connected yet.
2. The CDS cells are out in the open during testing.
3. Reset button





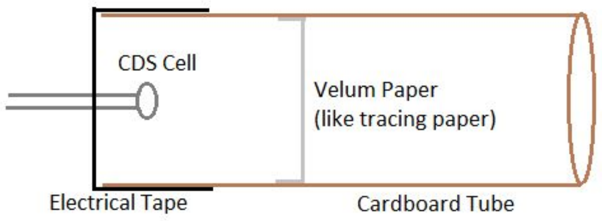


Image Notes
1. the CDS cell by itself would be too small to hit with the laser easily

Step 4: Putting It All Together

Set Up Your Garage
Build a path or a maze for trick-or-treaters to follow as they dodge the lasers. An easy and inexpensive way to make walls is to use cardboard or PVC tubes attached to the floor and ceiling, and wrap caution tape between them to form walls. The caution tape also leaves plenty of space for the lasers to pass through. After you have the walls built, set up the red light and power strip/servo you prepared. Make sure you have enough wire attached to the servo to reach your Arduino.

You will need a fog machine so the lasers beams are more visible. There are several Instructables for fog chillers if you want the fog to stay low to the ground, but it isn't necessary. It also helps keep fog in the garage if you block off parts of the garage doors with cardboard walls or some other kind of barrier to keep wind out and fog in.

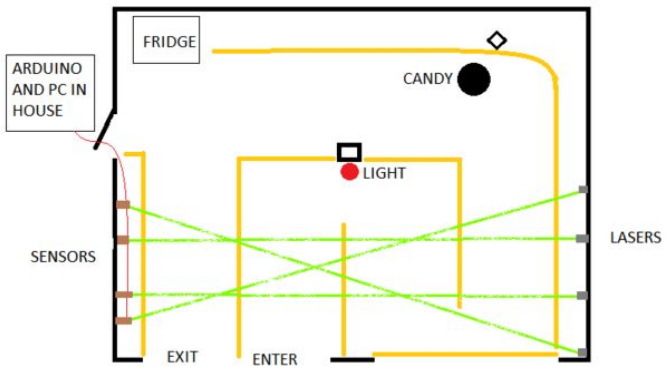
Set Up The Lasers
The lasers are not connected to the Arduino or PC in any way. The laser pointers come with AAA batteries by default. The batteries die quickly when you leave the laser on constantly. You can leave the battery cover off of the laser case, and power them with D-cell batteries instead. This will help extend the run time, and it will provide a more constant brightness. Use alligator clips or solder to connect wires to the spring(negative) and threaded part of the laser case(positive). Two D-cell batteries will power 2 laser pointers for over 6 hours. Each green laser pointer draws around 250-300mA, so keep that in mind if you come up with an alternative way to power them.

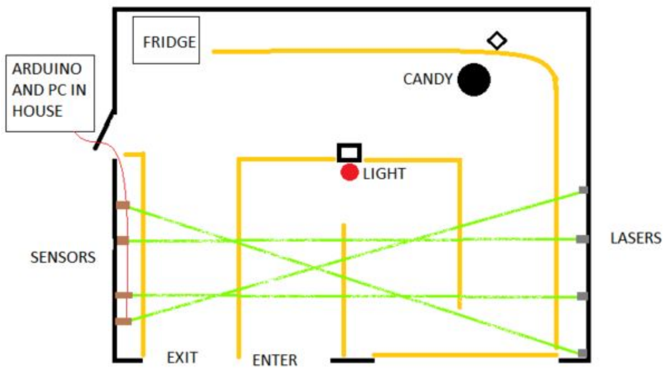
Once you get the lasers powered, position them in your garage. Put them 4-8 inches off the ground so that costumes with dresses or robes can make it over them. Use L brackets to fasten the lasers to the wall, or mount them on boxes or wood blocks. Use black tape to hold the power button on. The floorplan picture shows how I set up the lasers and walls in my garage. I put a caution tape or cardboard wall in front of the lasers and sensors to prevent people from accidentally bumping or kicking them.

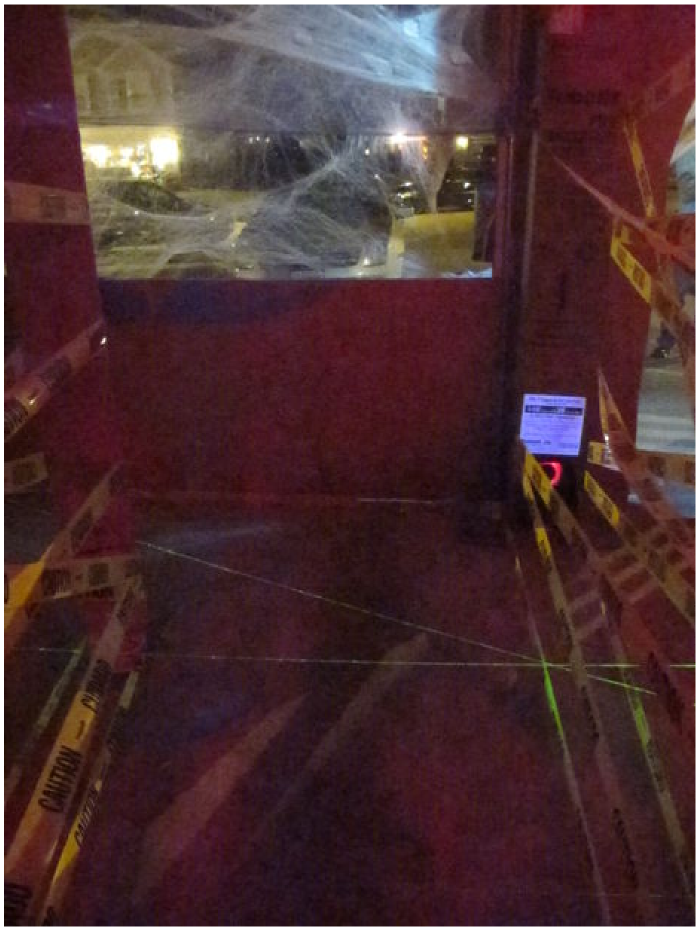
Set up the Arduino and PC
With the lasers in place and powered on, it is easy to position your light sensors. Mount the light sensors on the opposite wall so the lasers shine directly on them. Make sure you have enough wire on your light sensors to reach your Arduino. Plug in your light sensors and servo to the breadboard according to the circuit diagram on step 3. Plug the USB cord in to your Arduino and PC, and start up the Processing program.

Remember that you can use the Arduino's built in reset button to get a new average light reading and alarm trigger values as the ambient light and fog in your garage change. The bar graph display from Processing should help you get everything tuned and working perfectly.

You should have everything up and running at this point! The next few steps provide information on how the Processing and Arduino programs work, along with pictures and videos from Halloween 2012.







Step 5: Behind The Scenes - Arduino Code Detail

The Arduino code is in the LASER_Maze_2012.zip file attached to step 2 of this Instructable.

This step is just here to provide more information on how the Arduino code works.

The Arduino uses the analog inputs to monitor the value of 4 light sensors that are set up as voltage dividers . As soon as the Arduino is powered on, it averages together 3 readings for each light sensor. The average is the normal or expected amount of light when a laser is hitting the light sensor. If the light sensor value falls too far below this average, the Arduino knows the laser beam has been broken and sets off the alarms (activates the servo and sends the alarm signal to Processing).

You can calculate a new average and new alarm level by pressing the built in reset button on the Arduino. This is useful if the amount of ambient light or fog in the air changes enough to interfere with the sensors.

The Arduino also communicates with Processing using the Serial communication library. Each time a sensor value is read by the Arduino, it is sent to Processing to be displayed as a bar graph. When the alarm value for each sensor is calculated, it is sent to Processing to be displayed as a red line on that graph. The Arduino also sends a signal to Processing when the alarm servo is turned on or off. The signals being sent from Arduino to Processing are just numbers. The ones digit of the number lets Processing know what type of data has been sent, and the rest of the number is the actual data.

- Serial messages sent to processing. The digit in the ones place tells us what kind of data we have.
- ####0 - data from A0, where #### is the data and 0 is the code telling where the data belongs
- ####1 - data from A1
- ####2 - data from A2
- ####3 - data from A3
- ####4 - alarm trigger value for A0, where #### is the value and 4 is the code telling where the data belongs
- ####5 - alarm trigger value for A1
- ####6 - alarm trigger value for A2
- ####7 - alarm trigger value for A3
- 8 - alarm servo on
- 9 - alarm servo off

There are some values in the Arduino code that you may need to customize. They are all marked with a comment that says "###MODIFY IF NEEDED###". The customizable values include the number of light sensors you are using, the sensitivity (alarm trigger level), the amount of time the flashing red light/servo should stay active, and the on/off positions for the servo.

Step 6: Behind The Scenes - Processing Code Details

The Processing code is in the LASER_Maze_2012.zip file attached to step 2 of this Instructable.

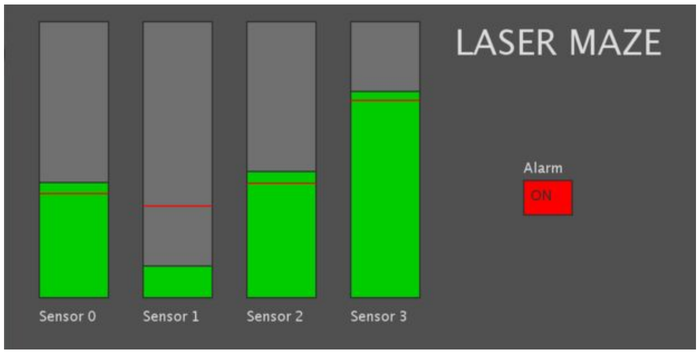
This step is just here to provide more information on how the Processing code works.

When the Processing program begins, it sets several variables that arrange the window layout, enables serial communication with the Arduino, and loads the 4 alarm sound clips from c:\. Every time the Arduino sends a serial message to Processing, the serialEvent() function is called.

The serialEvent() function does all of the real work. First it reads in the serial data, then it uses the last character in the serial string to determine what kind of data the Arduino has sent. The data will be a light sensor reading from one of the analog ports, an alarm trigger value for one of the sensors, or the servo turning on or off. The switch statement decides what action to take based on the type of data received.

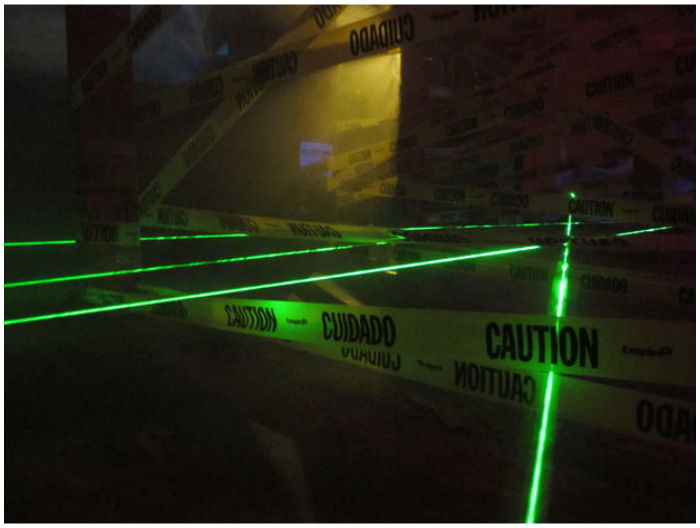
- If a sensor value is received, it is used to draw the green bar graph.
- If an alarm trigger value is received, it is used to draw the red line on the graph.
- If a servo ON signal is received, we know the laser has been broken. Processing will play one of the alarm sound clips and the alarm servo indicator turns green. There is a 3 second delay from the start of one alarm sound to the start of the next sound. This prevents constant triggering of the alarm sounds.
- If a servo OFF signal is received, the alarm servo indicator turns red.

The Processing code also writes a lot of the raw data to the Processing console area to help with troubleshooting and debugging.

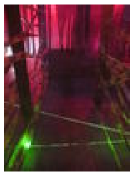


Step 7: The LASER Maze In Action - Videos and Pics

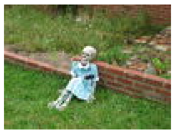
Pics and videos of the LASER Maze in action will be added here after Halloween!



Related Instructables



LASER Maze - Halloween Haunted House by bkhurt



Tunnel of Terror Halloween 2007 (Photos) by penandsword

Maze Solving Robot by patrickmccb

Make a Haunted House using PVC Pipe and Plastic Backdrops (video) by Andy Beck

Arduino powered Haunted Mansion Singing Busts by TheNewHobbyist

2011 Haunted House with help from instructables posters by l8nite