

## 'Hide and Seek' Salt and Pepper Shakers

### Inspiration

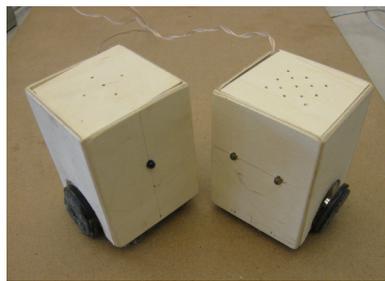
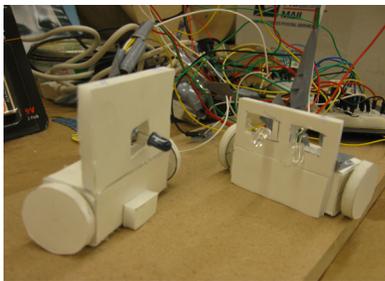
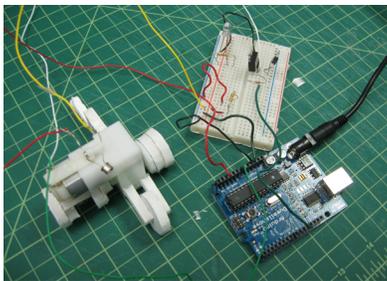
"If asked for the salt or pepper, pass both together, even if a table mate asks for only one of them. This is so dinner guests won't have to search for orphaned shakers. Set any passed item... directly on the table instead of passing hand-to-hand."

<http://whatscookingamerica.net/Menu/DiningEtiquetteGuide.htm>

### Summary

This project is for diners who want to be entertained while they eat. The 'hide and seek' salt and pepper shakers locate each other on the table. They are unlike ordinary shakers that are easily separated and lost from one another and they act like a discussion piece at dinner.

### Photographs

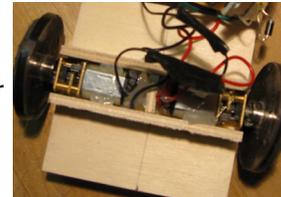


### Parts List

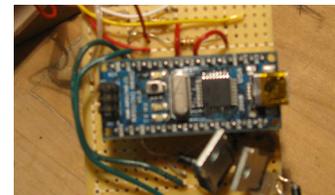
- 2 arduino nanos ([www.robotshop.com](http://www.robotshop.com))
- 4 56.8:1 Mini Metal Gear Motor ([www.solarbotics.com](http://www.solarbotics.com))
- 1 IR Emitter (Radio Shack)
- 2 IR Sensors (Radio Shack)
- 4 Tip 120s Transistors
- Resistors: 4 1k ohm, 2 10k ohm, 1 200 ohm
- Wire
- Solder
- Electrical Board
- 9 volt adapter
- Plywood
- 4 'wheels' approx. 1.25"
- 1 mini usb cable

### Details

motor



circuit board



### Goal

The salt shaker should find and follow the pepper shaker who slowly tires to run away.

### Additional Information

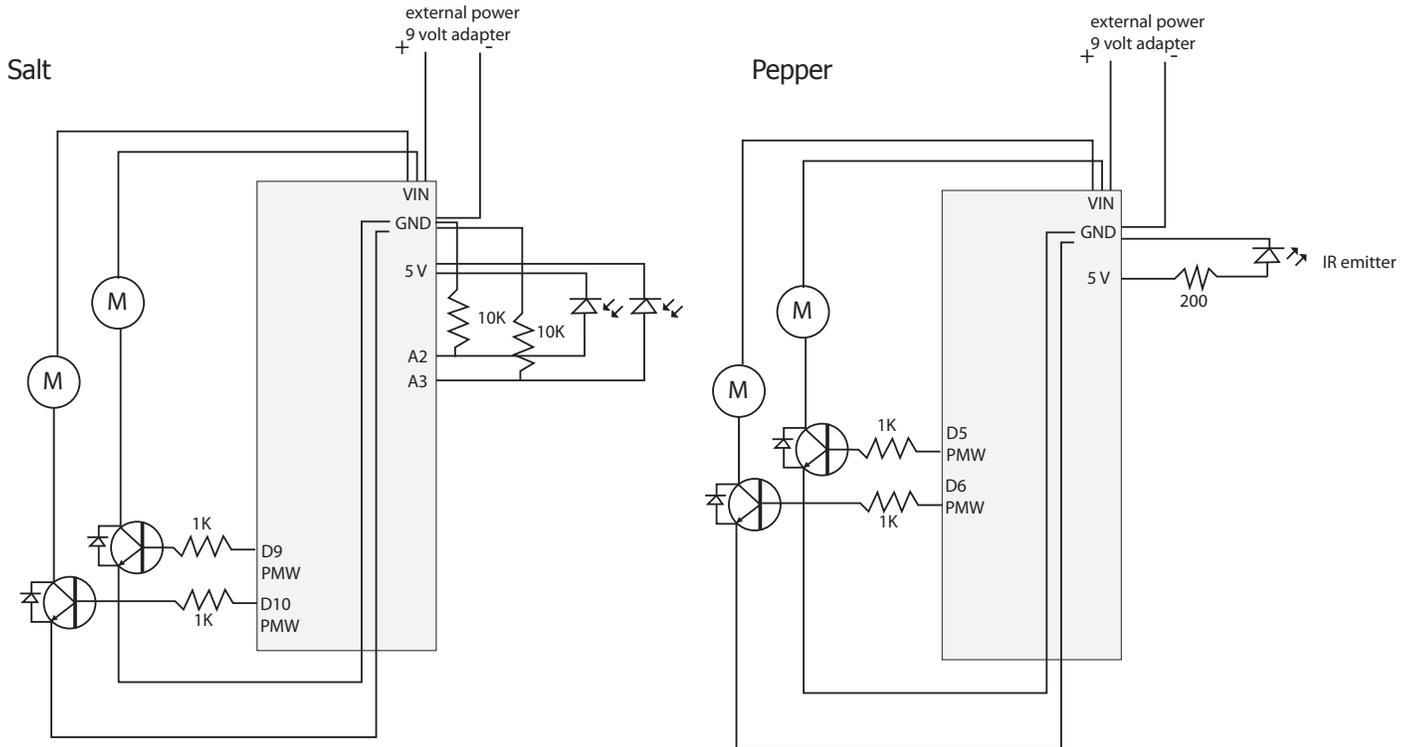
To repeat this project, it is necessary to make two units that locate each other with infrared light and travel on wheels. The general functionality is made of 4 parts. First the infrared emitter must independently emit light therefore, it should be hooked directly to electricity as it will be constantly on. Second, the emitter's motors should create motion at their own pace (based on a random number). The third part is the infrared sensors that are placed in a pair. These independently read the infrared signal and control their corresponding motor (part four). Therefore, in theory the shaker will turn towards the light.

To make this, the boxes must be suspended off the ground however, I found that using felt or some soft material was a good buffer to rest on the floor without preventing the motion. Each shaker has its own arduino so as to prevent a much connection as possible. Ideally, the boxes should have openings for servicing like a whole for the usb port on the arduinos.

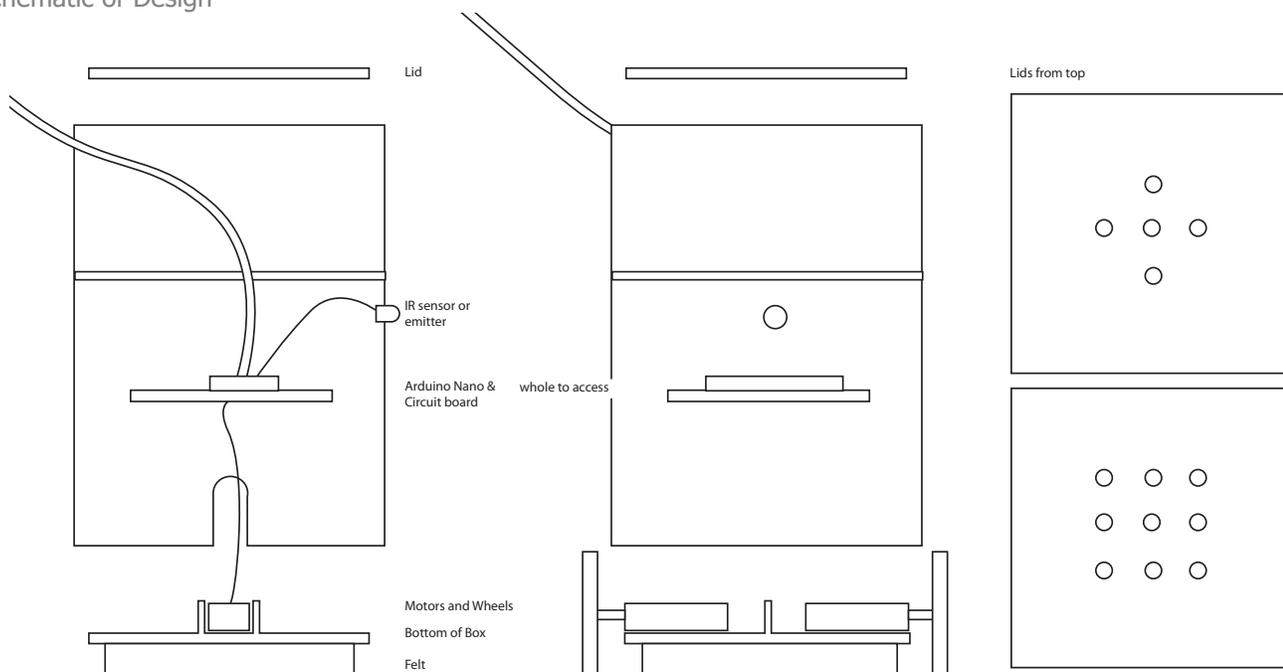
They can be further disconnected if a battery is placed in the upper cavity and a switch is used just to regulate the electricity.

To improve on this project, a pressure switch could be added to the bottom of the shaker so that when removed from the table top, it stops moving. Furthermore, a real impressive addition would be to include navigation around other object using distance sensors. However, the individual motion should be highly controlled prior to attempting this.

### Schematic of Electronics



### Schematic of Design



Code (two sperate files)

### Salt

```
int motorSensorA = 10; // the two motor pins
int motorSensorB = 9;

int sensorA = 2; // the two sensor pins
int sensorB = 3;

int sensorAval = 0; //starting sensor values
int sensorBval = 0;
//int light = ;

void setup()
{
  Serial.begin(9600);

  pinMode(motorSensorA,OUTPUT);
  pinMode(motorSensorB, OUTPUT);
  pinMode(sensorA, INPUT);
  pinMode(sensorB, INPUT);
  pinMode(switchPin, INPUT);
}

void loop()
{
  int sensorAval = analogRead(sensorA); //read the ir value
  int sensorBval = analogRead(sensorB);

  Serial.print("SensorA");
  Serial.print(sensorAval);
  Serial.print("SensorB");
  Serial.println(sensorBval);

  // if no light is seen only turn one wheel
  if(sensorAval == 0 && sensorBval== 0){
    motorSensorA = 0;
    motorSensorB = 30;
    Serial.print("turning");
  }
  // if too much ir, it is close so stop
  else if ((sensorAval + sensorBval > 1400) && (sensorAval > 995 || sensorBval > 1000)){
    motorSensorA = 0;
    motorSensorB = 0;
    Serial.print("stopped");
  }
  // if in between other two turn motor based on sensor readings
  else{
    analogWrite(motorSensorA, constrain((int)(1000-sensorAval)/10, 0, 75));
    analogWrite(motorSensorB, constrain((int)(1000-sensorBval)/10, 0, 75));
    Serial.println(constrain((1000-sensorAval)/10, 0, 75));
    Serial.println(constrain((1000-sensorBval)/10, 0, 75));
  }
}
```

### Pepper

```
int motor = 5; //two motor pins
int motor2 = 6;

void setup()
{
  Serial.begin(9600);

  pinMode(motor,OUTPUT);
  pinMode(motor2, OUTPUT);
}

void loop()
{
  //find a random number between 0 and 100
  long mutualSpeed = random(100);

  //set motors to the random number speed
  analogWrite(motor, mutualSpeed);
  analogWrite(motor2, mutualSpeed);

  Serial.println("speed " + mutualSpeed);
  delay(300); //wait

  //stop both motors
  analogWrite(motor, 0);
  analogWrite(motor2, 0);
  delay(1000); //wait
}
```