


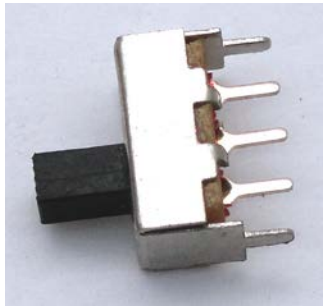




EG Robotics LDR/IR Beginner Robot Instruction Manual

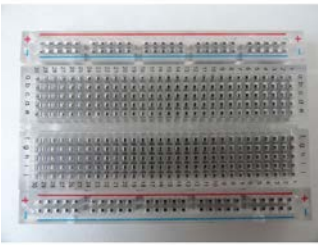


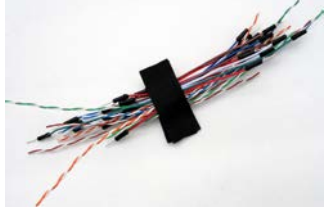

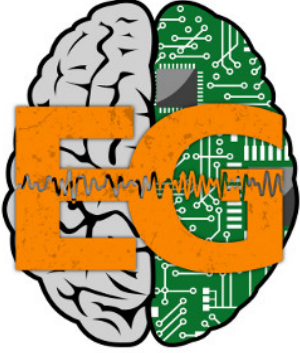
Introduction

The LDR/IR Robot is designed to seek out light and avoid obstacles. The robot uses the Light Dependent Resistor to seek out light and the Infrared Sensor to detect and avoid objects in its path. The following are instructions with pictures to show exactly how we built our robot. Your robot can be designed the way you desire and in your own order. At the end of this manual you will find some pictures of various robots built with this kit. Use your imagination and make your robot unique. Have fun!

Kit Component List

1 x Arduino	 Arduino	1 x Battery Holder	
1 x Acrylic Chassis		1 x On/Off Switch	
2 x Motor Gearbox and Wheel		1 x Power Jack	

3 x LDR		1 x IR Receiver	
12 x LED		2 x IR LED	
14 x Resistor		1 x Servo	
4 x Transistor		1 x Ping Pong Ball	
4 x Diode		2 x Screw	

1 x Bread Board		2 x Velcro	
1 x Straw		1 x Wire Pack	
Popsicle Sticks		2 x Sticker	

The components listed above may differ slightly from the components in your kit.

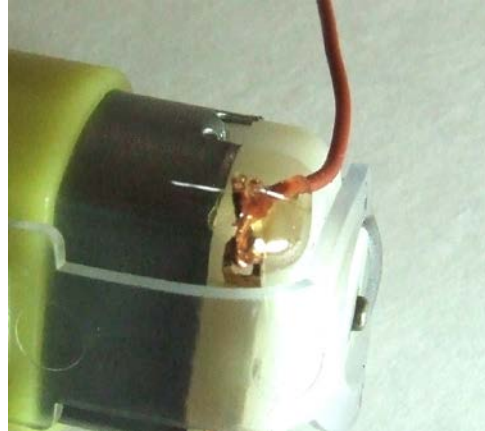
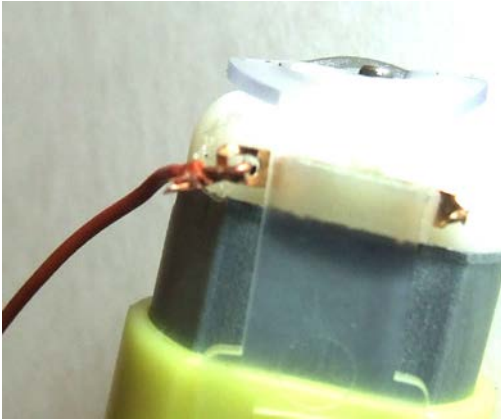
Suggested Tools

Hot Glue Gun		Scissors	
Small Screwdriver		Needle Nose Pliers	
Wire Stripper		Sharpie	
Electrical Tape		Soldering Iron	

Section 1 - Building a Chassis

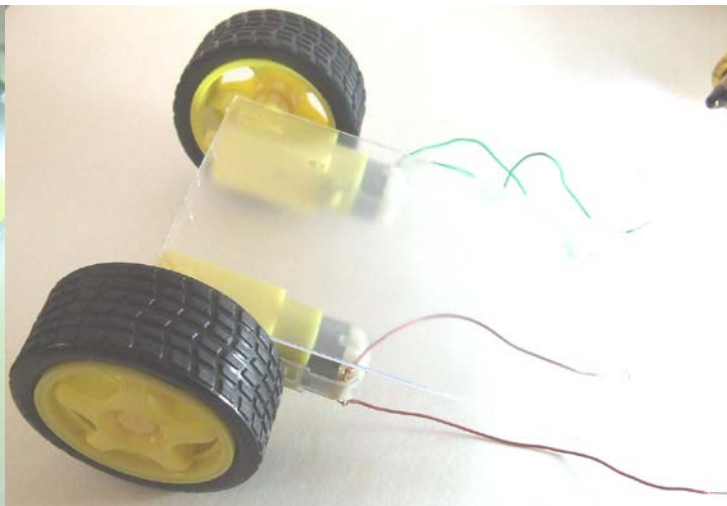
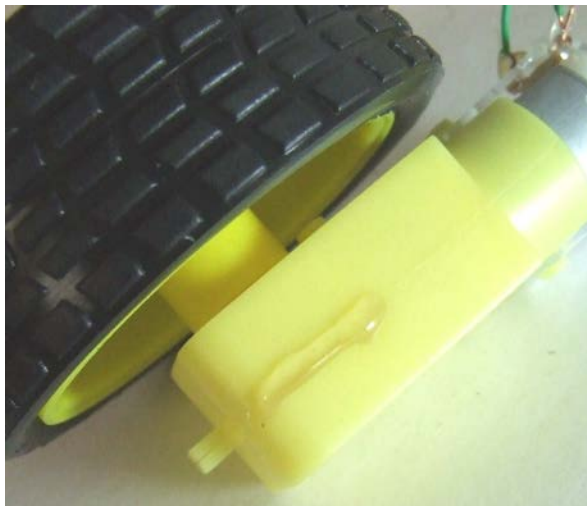
Step One

The chassis of the robot is not predetermined, make it how you want, the only rule is to make it awesome! Find the motor, which is already placed within the yellow gearbox, but can be taken out if you so choose for other projects. Attach a pair of the twisted wires to the motor's metal tabs by first stripping the ends of the wires and then inserting them through the holes in the tabs. Normally this would be done using a soldering gun however it works well to insert the wire through the hole and firmly clamp the wire around it using needle nose pliers. Be careful not to twist the tab on the motor because it might break. Once it is connected firmly, place a drop of hot glue over it to keep it in place.



Step Two

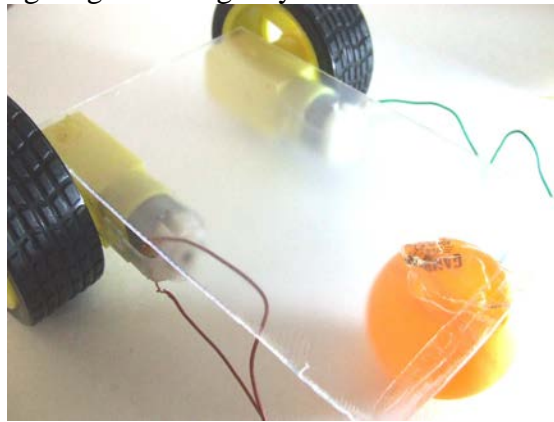
Attach the motor-gearbox to the chassis. This can be done using a dab of hot glue on the chassis and then pressing the motor-gearbox into it. Be sure to use just a little bit of hot glue otherwise it might be hard to get off if it needs to be repositioned. Align the two wheels so that it will naturally drive in a straight line. It is alright if it veers to one side a little because it is sensor driven and will auto self-correct.



Step Three

Attach the Ping Pong ball to the chassis in such a way that it keeps it balanced and upright. Depending on the design you chose for the chassis, ping pong ball placement will be more or less crucial. Some designs have needed to extend the ball out farther using a popsicle stick for more of a drag racer look in order to keep from

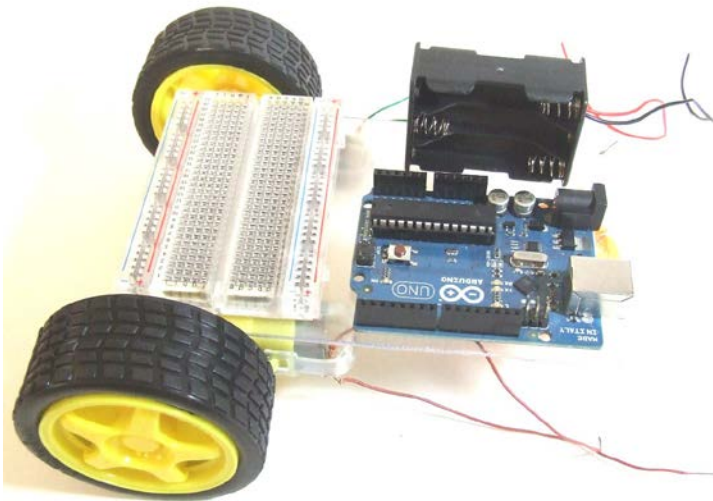
tipping over or flipping. Smash one side of the ball in for greater surface area contact and hot glue the smashed side to the chassis. Remember just a little bit of hot glue goes a long ways!



Step Four

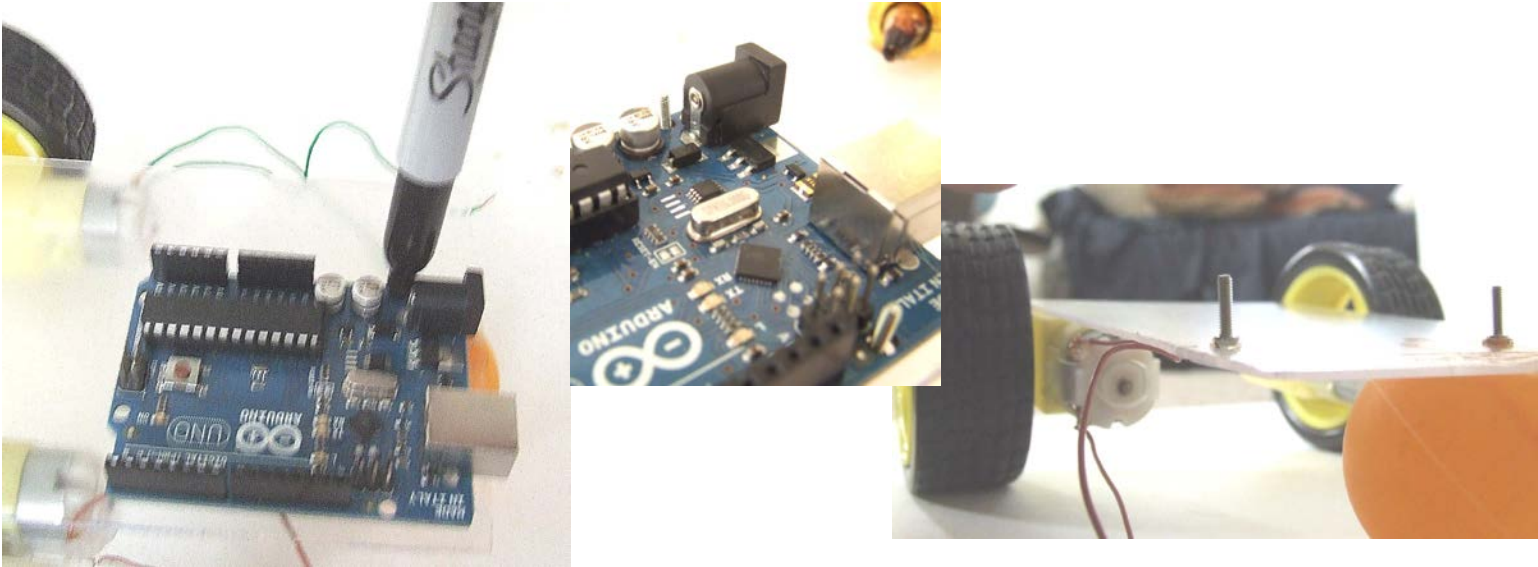
Plan out the location of the breadboard, Arduino and battery pack. When choosing the location of the three components there are three things to keep in mind.

1. The distance the wires will have to travel in order to connect the Arduino to the breadboard which will be connected to the motors and sensors.
2. The batteries will need to be taken out at some point so make sure your design allows for this.
3. Access to the USB Port on the Arduino is important as the USB cable will need room to connect as well as the power jack into the Arduino.



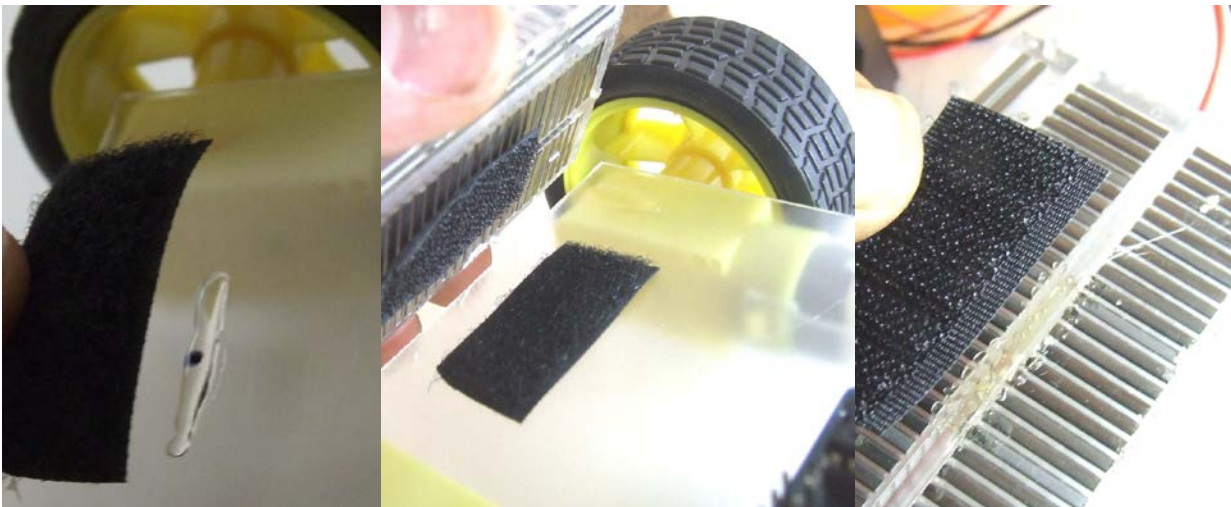
Step Five

Put the Arduino on the chassis in its planned location. Using the sharpie mark the locations of the Arduino's tie down holes. Take the Arduino temporarily off and hot glue the provided screws heads down onto the sharpie marks. Once the glue has dried, place the Arduino onto the screws to test how well it fits. If it doesn't fit well enough, pry it off and try again until you have a snug fit.



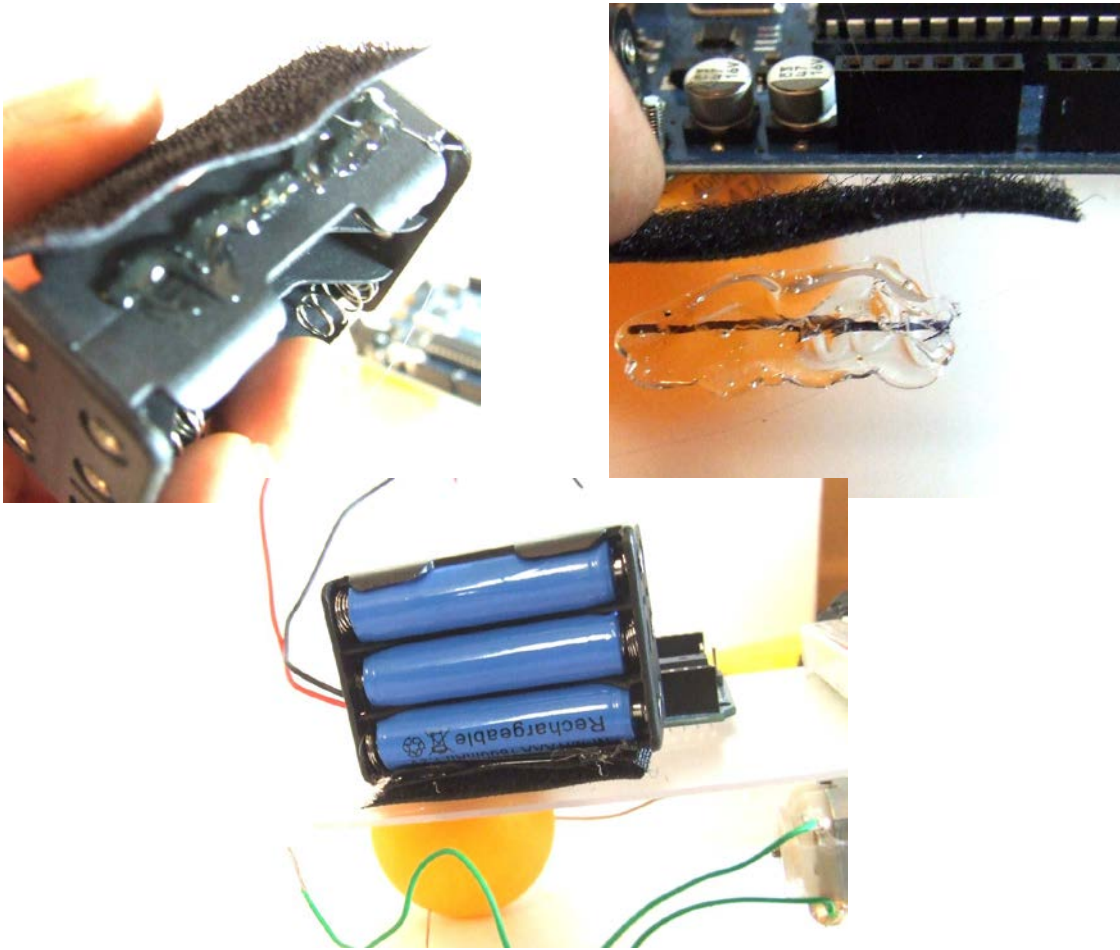
Step Six

Put the breadboard down in the location you have chosen and mark it. You can do this next step one of two ways. The first way is to simply pull off the paper from the back of the breadboard and reveal the sticky side and place it down where you want it. The second way is to hot glue the two small Velcro pieces to the chassis and the other two to the back of the breadboard back removing a small portion of the paper from the back side. This way it is removable.



Step Seven

Attach the battery pack by hot gluing one side of the Velcro to the edge of the battery pack and the other side onto the chassis where you want it. Use plenty of hot glue to cover the entire surface to ensure that it sticks firmly. (Remember it needs to be easily accessible.)



Section 2 – Wiring

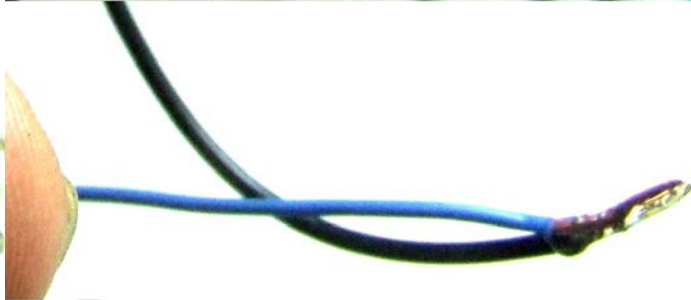
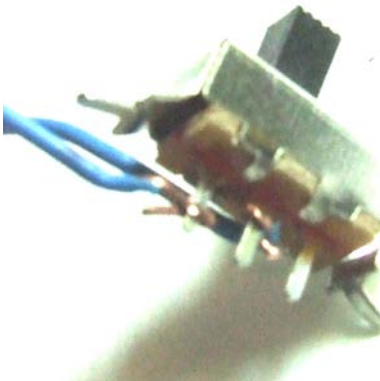
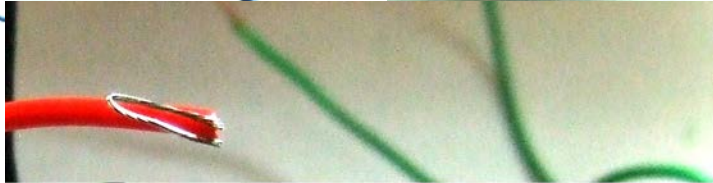
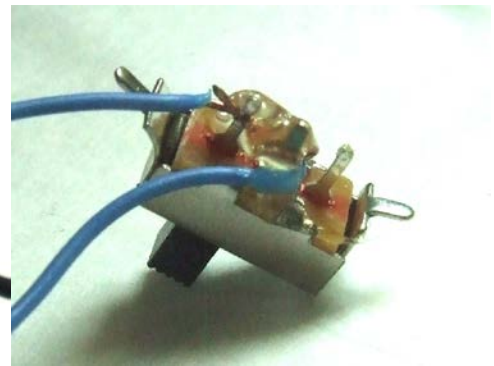
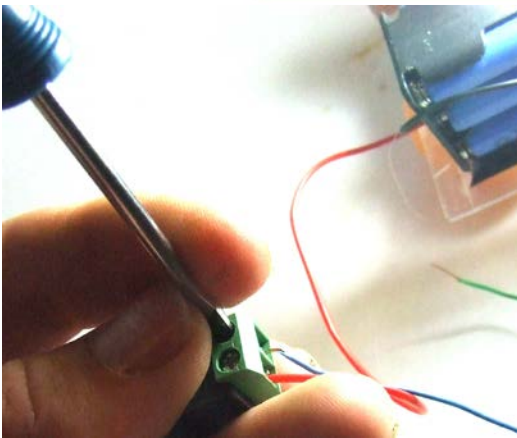
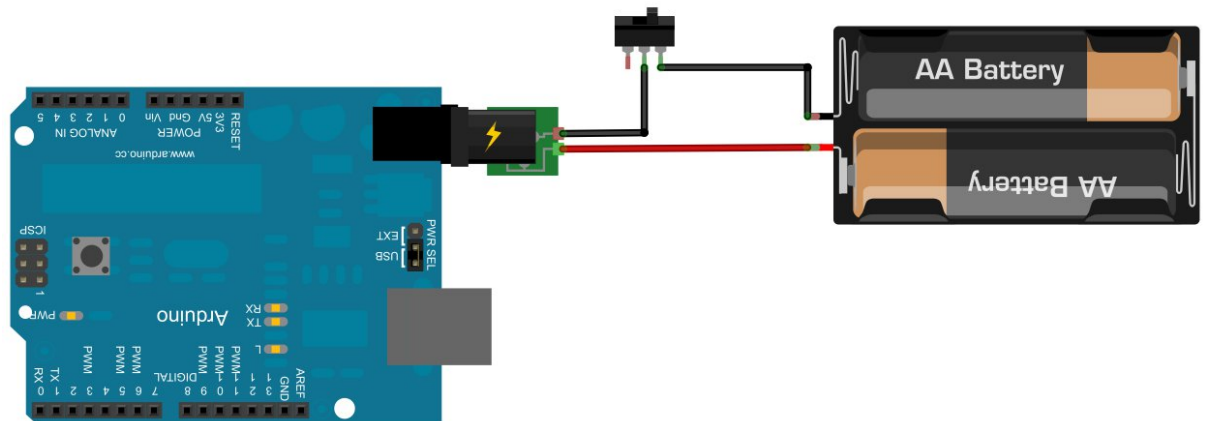
Step One

A) Wire the battery negative or positive lead to the On/Off switch. Attach a loose wire onto the other pin on the switch. When this is done and both wires are firmly attached to the switch, hot glue the bottom of the switch to keep the wires secured. A better way would be to solder them to the pins if that is an option for you.

B) Now take the other lead either positive or negative from the battery depending on which one you choose and wire it directly to the power jack. Unscrew the power jack's screws, fold the bare wire back onto its own wire covering, then insert the wires making sure black is connected to the negative label and red is connected to the positive label. This should create a circuit with the on/off switch enabling or disabling power.

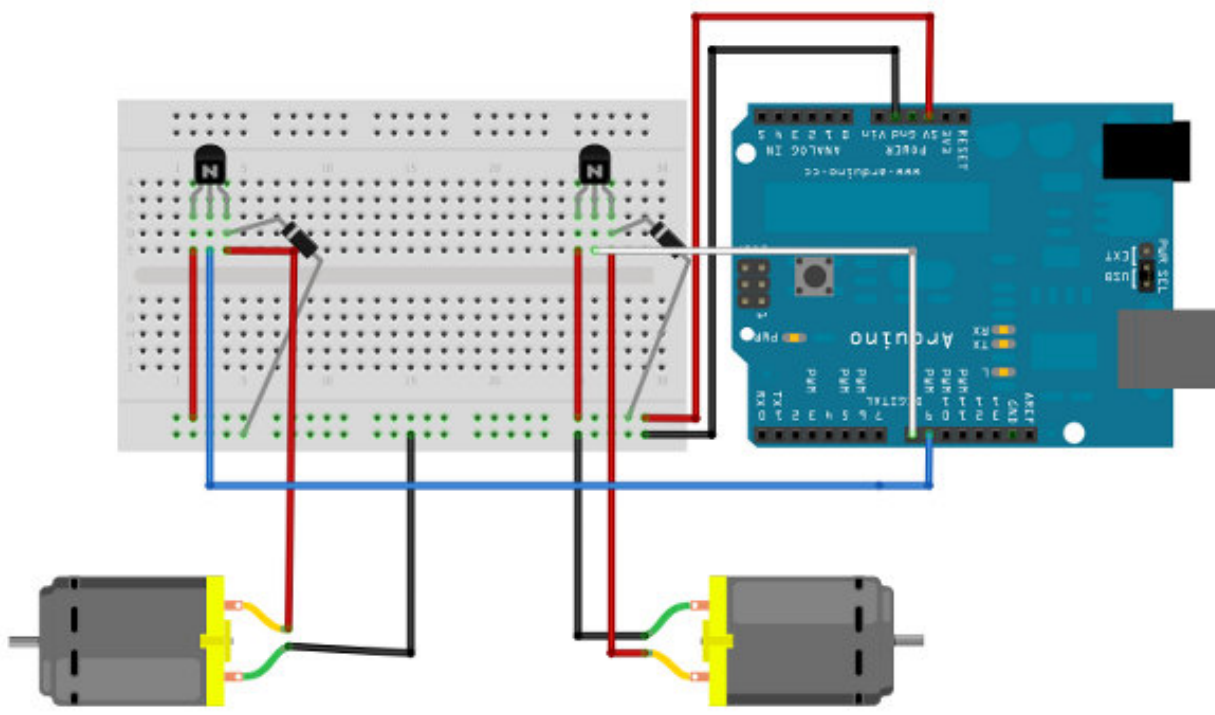
C) Next mount the switch to the chassis somewhere accessible and easy for you to switch on or off. Use hot glue to place the switch in the desired position.

D) Test to make sure good connections were made by flipping the switch on and confirming the Arduino's power light stays constant.



Step Two

We will now create our first circuit using the Arduino board. Find the two transistors and two diodes in the clear bag without a color and place them on the breadboard. Use the following diagram to build the circuit using the jumper wires to connect from the Arduino to the breadboard.

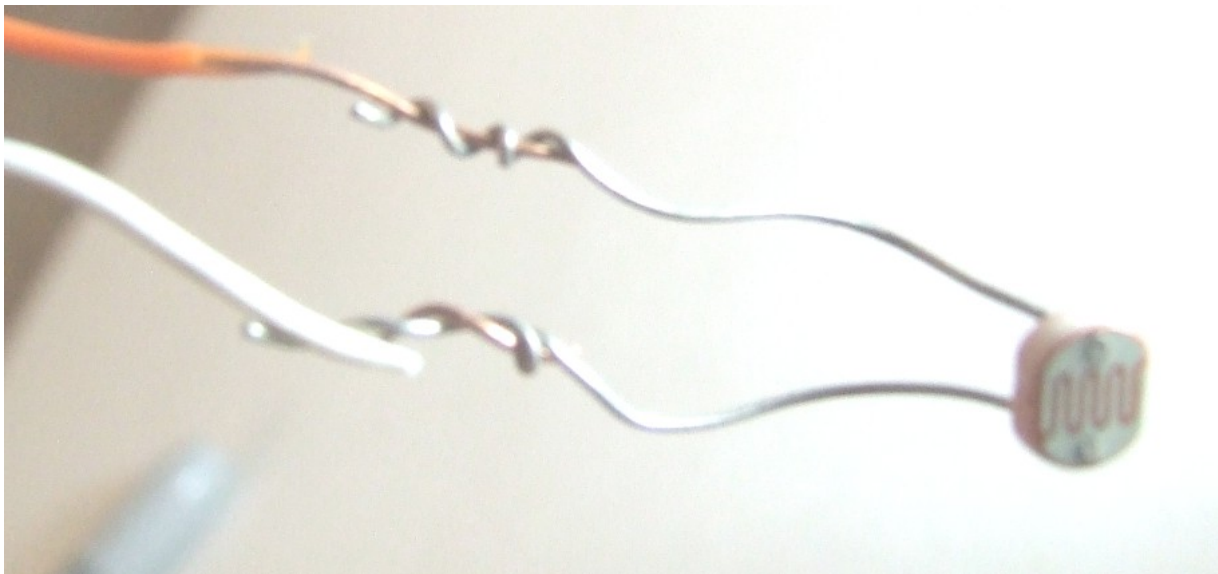


TIP: If the motors seem to be going the wrong direction switch the polarity of the motor by making the positive side the negative and the negative the positive.

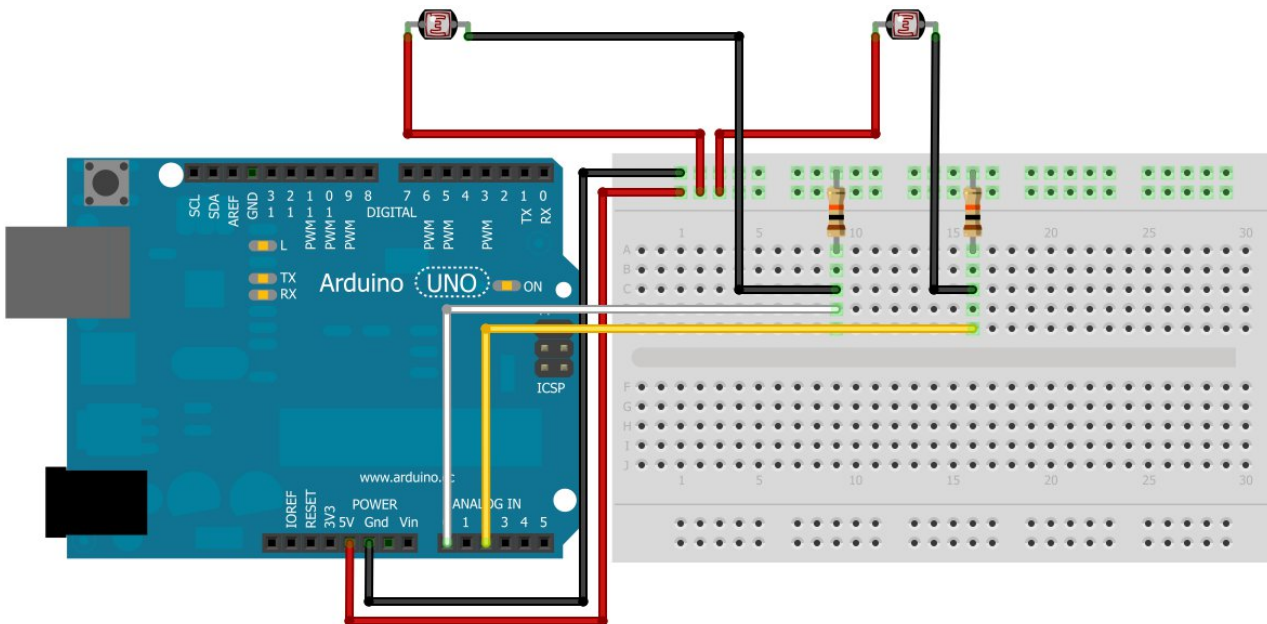
The motor on the left side of the robot should be connected to digital pin 8 and the motor on the right side connected to digital pin 9.

Step Three

Attach the long twisted wires to the Light Dependent Resistor which can be found in the green bag. Twist the end of the LDR and the end of the wire around each other. Use the needle nose pliers to get a good solid twist. If you want you may put a dab of glue to hold them together.



Wire the Light Detecting Circuit as follows (The 10K Ohm Resistors are used in this step):




The LDR connected to analog pin 2 should be the right sensor and the LDR connected to analog pin 0 should be the left sensor. (If you switch these your robot will be a light fearing robot.)

Be sure the LDR's are pointed diagonally. One to the left and one to the right facing toward the level the lights are coming from.

Test the circuit using this code:


(<http://www.egrobotics.com/instructions/code/testldr.ino>)

```
void setup() {
  Serial.begin(9600);
}
void loop() {
  int sensor1Value = analogRead(A0);
  Serial.print("Analog Pin 0 Reading = ");
  Serial.println(sensor1Value);
  delay(1000);
  int sensor2Value = analogRead(A2);
  Serial.print("Analog Pin 2 Reading = ");
  Serial.println(sensor2Value);
  delay(1000);
}
```

Open the Serial Monitor by clicking the  in the top right corner of the Arduino software interface or by hitting Ctrl+Shift+M. This should bring up the serial monitor and start displaying values. Cover one sensor and you should see one of the values drop. Cover the other sensor and its value should also drop. If it doesn't, be sure you used the correct resistor, the connections are good, and you are not in a dark room.

Section 3 – LDR Completion and Testing

Step One

Copy the following code into the arduino sketch on your computer and upload the code by clicking  .
<http://www.egrobotics.com/instructions/code/robotldr.ino>

```
const int RightSensor = 2;
const int LeftSensor = 0;

int SensorLeft;
int SensorRight;
int SensorDifference;

void setup() {

  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(LeftSensor, INPUT);
  pinMode(RightSensor, INPUT);
  Serial.begin(9600);
  Serial.println("\nBeginning Light Seeking Behavior");
}

void loop() {
  SensorLeft = 1023 - analogRead(LeftSensor);
  delay(1);
  SensorRight = 1023 - analogRead(RightSensor);
  delay(1);
  SensorDifference = abs(SensorLeft - SensorRight);

  Serial.print("Left Sensor = ");
  Serial.print(SensorLeft);
  Serial.print("\t");
  Serial.print("Right Sensor = ");
  Serial.print(SensorRight);
  Serial.print("\t");

  if (SensorLeft > SensorRight && SensorDifference > 75) {
    Serial.println("Left");
    digitalWrite(8, HIGH);
    delay(250);
    digitalWrite(8, LOW);
    delay(100);
  }

  if (SensorLeft < SensorRight && SensorDifference > 75) {
    Serial.println("Right");
```



```
digitalWrite(9, HIGH);  
delay(250);  
digitalWrite(9, LOW);  
delay(100);  
}  
  
else if (SensorDifference < 75) {  
  Serial.println("Forward");  
  digitalWrite(8, HIGH);  
  digitalWrite(9, HIGH);  
  delay(500);  
  digitalWrite(8, LOW);  
  digitalWrite(9, LOW);  
  delay(250);  
  
}  
Serial.print("\n");  
}
```

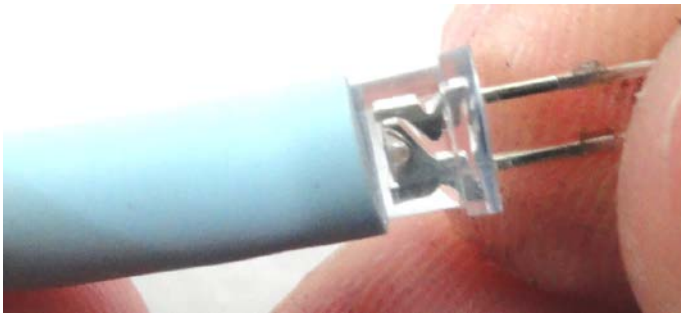
Step Two

You have now completed the LDR part of the robot. It should now be able to seek out the brightest light source in a room. Turn it on and test it out. Also try turning off the lights in a room and leaving a door open with light coming through and see if it will travel through it. Another fun thing to do is take a flash light and try to direct its course with it!

Section 4 – IR Addition

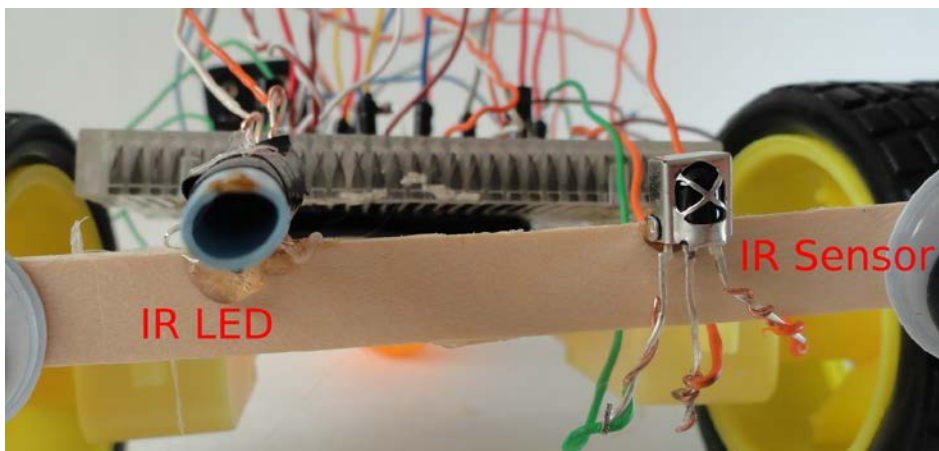
Step One:

Wire the IR Sensor and IR LED (found in the clear bag) with the twisted wires using the same method used for the LDRs. Strip both ends of all 5 wires (2 for the IR LED and 3 for the IR Receiver), twist the wires around the connections and use a dab of hot glue to make the connections hold together. Next place the plastic straw or cover over the IR LED to make it unidirectional. You may need to wrap the IR LED in electrical or black duct tape to keep the IR light from seeping out in all directions. Be sure to wrap the back of the LED as well.



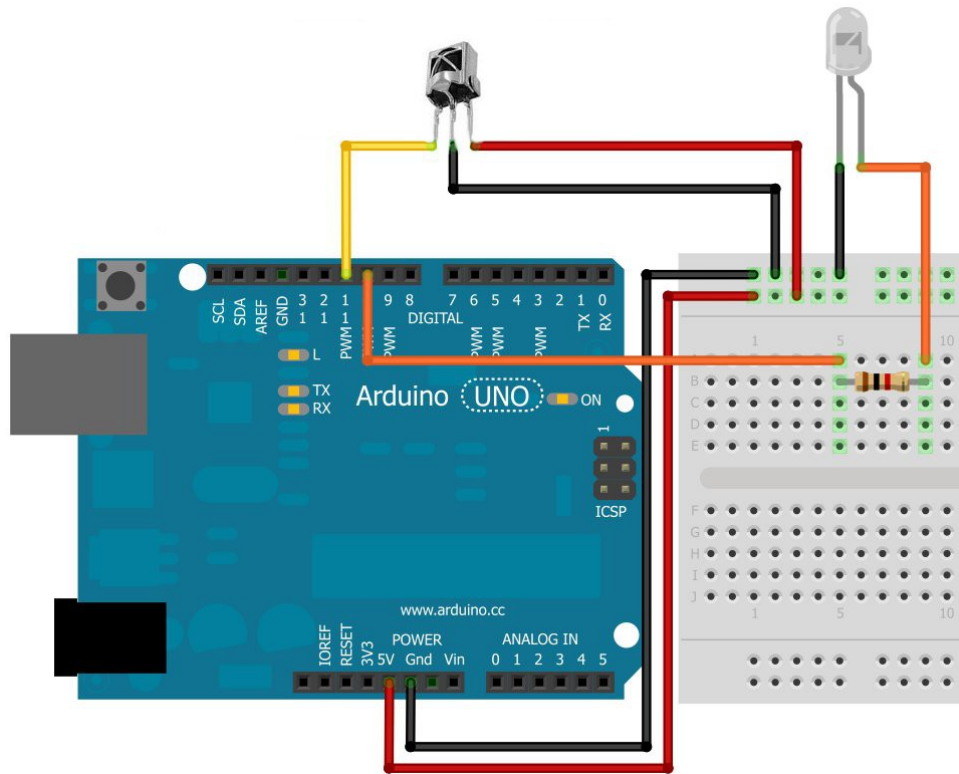
Step Two:

Attach the IR LED and IR Sensor to the front of your robot or wherever you want it to detect an object. It helps to have them both fixed to the chassis. If the sensors are left loose they will bounce around and make it difficult to get an accurate reading. Be sure the IR Sensor is behind the front of the IR Straw.



Step Three:

Wire the IR Sensor and IR LED as shown below. The resistor is a 470 Ohm Resistor.



Step Four:

Use the following code to determine if the IR detection is set up properly. Upload the code, then put your hand in front of the IR LED and the Arduino's LED should turn on. When you move your hand away the LED should turn off. (In the picture above the LED is labeled with an "L")

Code:

(<http://www.egrobotics.com/instructions/code/testir.ino>)

```
#define IRsensorPin 11
#define IRledPin 10
#define D13ledPin 13
```

```
void IR38Write() {
  for(int i = 0; i <= 384; i++) {
    digitalWrite(IRledPin, HIGH);
    delayMicroseconds(13);
    digitalWrite(IRledPin, LOW);
    delayMicroseconds(13);
  }
}

void setup(){
  pinMode(IRledPin, OUTPUT);
  digitalWrite(IRledPin, LOW);
  pinMode(D13ledPin, OUTPUT);
  digitalWrite(D13ledPin, LOW);
}
```

```

void loop(){
  IR38Write();
  if (digitalRead(IRsensorPin)==LOW){
    digitalWrite(D13ledPin, HIGH);
  } else {
    digitalWrite(D13ledPin, LOW);
  }
  delay(100);
}

```

Troubleshooting:

If the LED is always on or comes on and off often then add more tape around the straw or put a divider wall between the IR LED and IR Sensor.

If the LED does not come on be sure everything is connected and plugged into the correct pins. Be sure the angles are correct. The light needs to be able to bounce off the object and return into the black part of the IR Sensor.

Step Five:

If your IR LED and Receiver are working properly the next thing to do is try out the following code that will activate both the LDR and IR to have the robot seek out light, but at the same time attempt to avoid obstacles that are in the way.

Code:

(<http://www.egrobotics.com/instructions/code/robotavoid.ino>)

```
#include <Servo.h>
```

```
Servo myservo;
```

```
#define IRsensorPin 11
```

```
#define IRledPin 10
```

```
const int RightSensor = 2;
```

```
const int LeftSensor = 0;
```

```
int IR;
```

```
int SensorLeft;
```

```
int SensorRight;
```

```
int SensorDifference;
```

```
void IR38Write() {
```

```
  for(int i = 0; i <= 384; i++) {
```

```
    digitalWrite(IRledPin, HIGH);
```

```
    delayMicroseconds(13);
```

```
    digitalWrite(IRledPin, LOW);
```

```
    delayMicroseconds(13);
```

```
  }
```

```
}
```

```
void setup() {
```

```
  myservo.attach(5);
```



```

pinMode(IRledPin, OUTPUT);
digitalWrite(IRledPin, LOW);
pinMode(8, OUTPUT);
pinMode(9, OUTPUT);
pinMode(LeftSensor, INPUT);
pinMode(RightSensor, INPUT);
Serial.begin(9600);
Serial.println("\nBeginning Light Seeking Behavior");
}

```

```

void loop() {
  IR38Write();
  IR = digitalRead(IRsensorPin);
  delay(50);
  SensorLeft = 1023 - analogRead(LeftSensor);
  delay(1);
  SensorRight = 1023 - analogRead(RightSensor);
  delay(1);
  SensorDifference = abs(SensorLeft - SensorRight);

```

```

  Serial.print("Left Sensor = ");
  Serial.print(SensorLeft);
  Serial.print("\t");
  Serial.print("Right Sensor = ");
  Serial.print(SensorRight);
  Serial.print("\t");

```

```

  if (SensorLeft > SensorRight && SensorDifference > 75 && IR == HIGH) {
    Serial.println("Left");
    digitalWrite(8, HIGH);
    delay(250);
    digitalWrite(8, LOW);
    delay(100);
  }

```

```

  if (IR == LOW){
    digitalWrite(8, HIGH);
    delay(1500);
    digitalWrite(8, LOW);
    digitalWrite(8, LOW);
    delay(150);
  }

```

```

  if (SensorLeft < SensorRight && SensorDifference > 75 && IR == HIGH) {
    digitalWrite(9, HIGH);
    delay(250);
    digitalWrite(9, LOW);
  }

```

```

    delay(100);
}

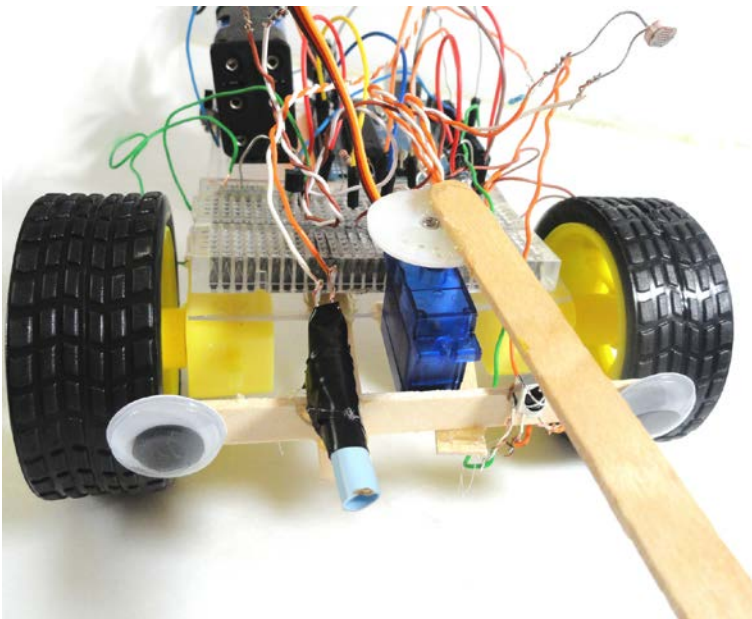
else if (SensorDifference < 75 && IR == HIGH) {
    Serial.println("Forward");
    digitalWrite(8, HIGH);
    digitalWrite(9, HIGH);
    delay(500);
    digitalWrite(8, LOW);
    digitalWrite(9, LOW);
    delay(250);
}
Serial.print("\n");
}

```

Section 5 – Adding a Servo

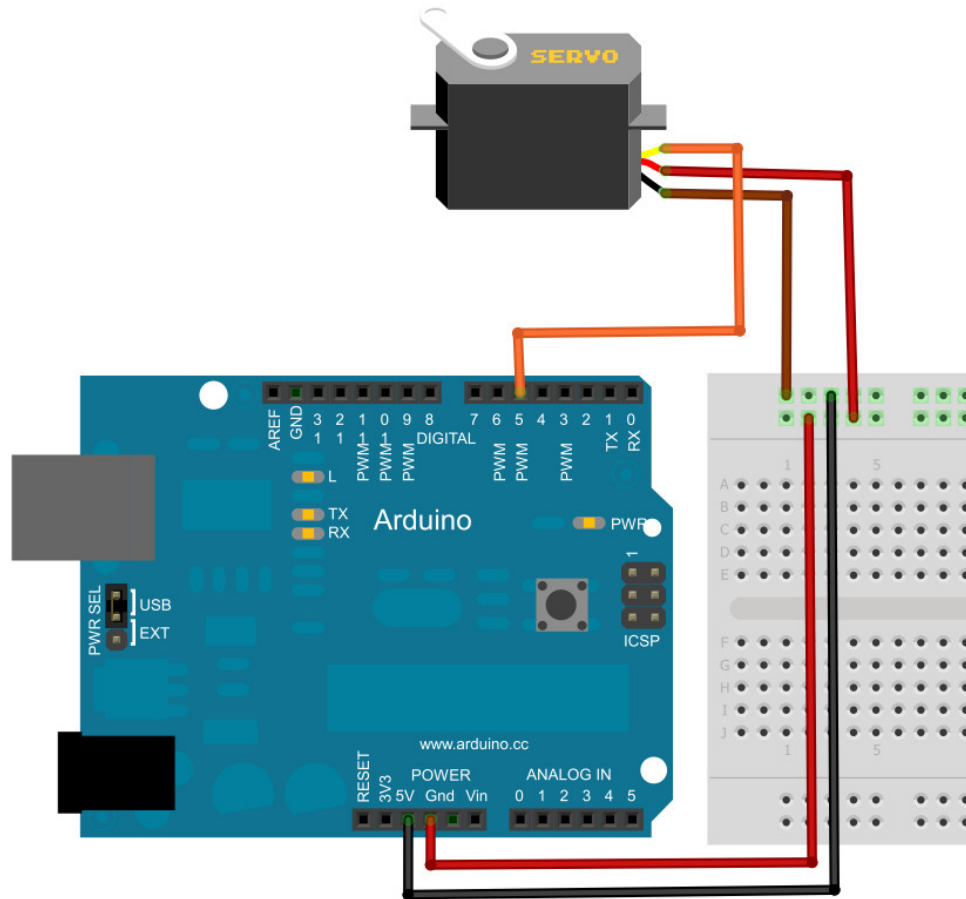
Step One:

This section will depend on what you want to do with the servo. You can use it as an attack arm or use it to lift something or to help move your vehicle out of the way when it is stuck. It is up to you on how it is placed and how it is used. Find a long stiff object to attach to a servo horn or use the Popsicle sticks provided. Use a hot glue gun to attach it to the servo horn. Attach the servo horn to the servo using a small screw. You can attach the servo using hot glue to the chassis of the robot. (Be sure the servo is attached and supported well as the servo may push itself loose.)



Step Two:

Wire the servo into the Arduino and Breadboard as shown below. It is easy to use the jumper wires and stick them into the end of the servo connector and then into the breadboard or arduino.



Step Three:

To test if your servo is properly wired and installed check it by uploading the example code for Servo Sweep. This is easily done by clicking **File > Examples > Servo > Sweep**. When that opens change *myservo.attach(9);* this code line to be *myservo.attach(5);* to match your pin you have attached it to.

Once you have uploaded this code it should have the servo sweep back and forth if you have it properly configured.

Step Four:

Upload this code to have your robot seek light and attack objects in its path with the servo.

Code:

(<http://www.egrobotics.com/instructions/code/robotattack.ino>)

```
#include <Servo.h>
```

```
Servo myservo;
```

```
#define IRsensorPin 11
```

```
#define IRledPin 10
```

```

const int RightSensor = 2;
const int LeftSensor = 0;

int IR;
int SensorLeft;
int SensorRight;
int SensorDifference;

void IR38Write() {
  for(int i = 0; i <= 384; i++) {
    digitalWrite(IRledPin, HIGH);
    delayMicroseconds(13);
    digitalWrite(IRledPin, LOW);
    delayMicroseconds(13);
  }
}

void setup() {
  myservo.attach(5);
  pinMode(IRledPin, OUTPUT);
  digitalWrite(IRledPin, LOW);
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(LeftSensor, INPUT);
  pinMode(RightSensor, INPUT);
  Serial.begin(9600);
  Serial.println("\nBeginning Light Seeking Behavior");
}

void loop() {
  IR38Write();
  IR = digitalRead(IRsensorPin);
  delay(50);
  SensorLeft = 1023 - analogRead(LeftSensor);
  delay(1);
  SensorRight = 1023 - analogRead(RightSensor);
  delay(1);
  SensorDifference = abs(SensorLeft - SensorRight);

  Serial.print("Left Sensor = ");
  Serial.print(SensorLeft);
  Serial.print("\t");
  Serial.print("Right Sensor = ");
  Serial.print(SensorRight);
  Serial.print("\t");

  if (SensorLeft > SensorRight && SensorDifference > 75 && IR == HIGH) {
    Serial.println("Left");
    digitalWrite(8, HIGH);
  }

```



```

delay(250);
  digitalWrite(8, LOW);
  delay(100);

}

if (IR == LOW){
  delay(500);
  digitalWrite(8, HIGH);
  digitalWrite(9, HIGH);
  myservo.write(30);
  delay(200);
  myservo.write(130);
  delay(200);
  digitalWrite(8, LOW);
  digitalWrite(9, LOW);
  delay(250);
}

if (SensorLeft < SensorRight && SensorDifference > 75 && IR == HIGH) {
  digitalWrite(9, HIGH);
  delay(250);
  digitalWrite(9, LOW);
  delay(100);
}

else if (SensorDifference < 75 && IR == HIGH) {
  Serial.println("Forward");
  digitalWrite(8, HIGH);
  digitalWrite(9, HIGH);
  delay(500);
  digitalWrite(8, LOW);
  digitalWrite(9, LOW);
  delay(250);
}
Serial.print("\n");
}

```

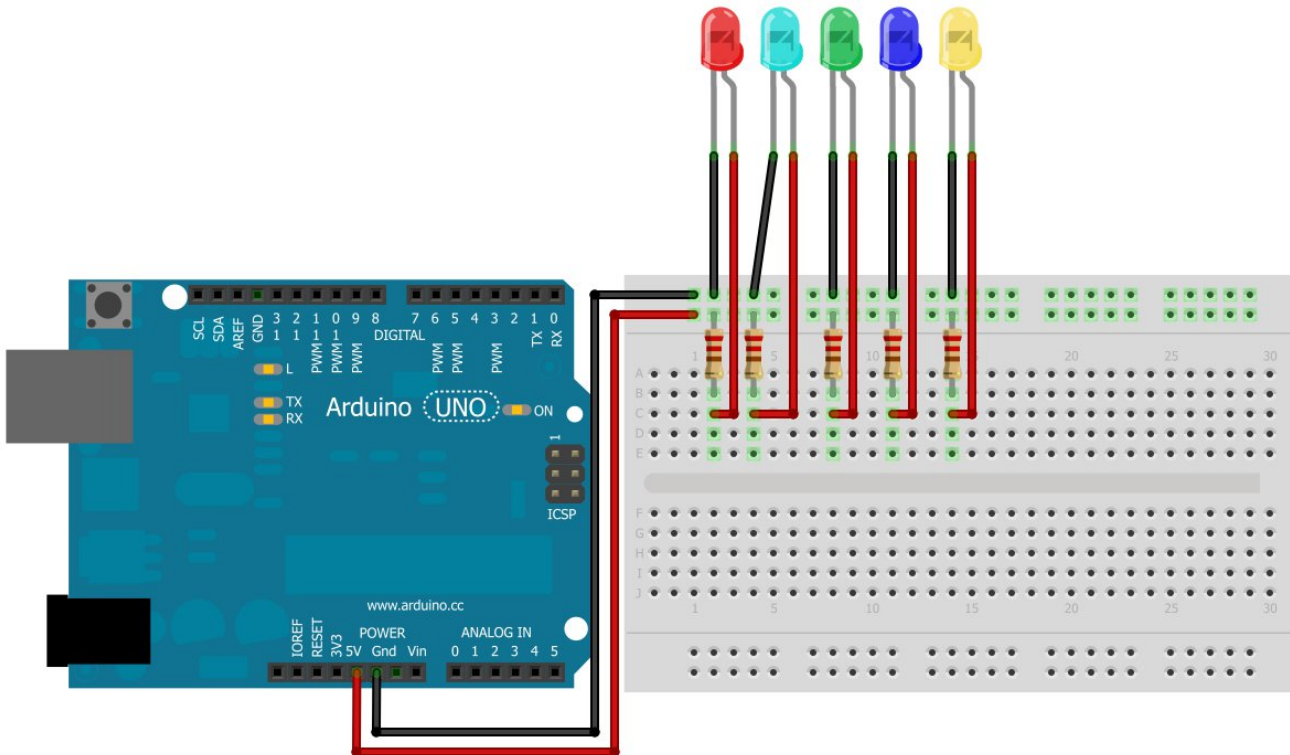
Step Five:

Now that you have completed the wiring and coding for your robot it is time to make the robot look unique! Use the provided materials and materials of your own to make the robot look or act the way you want it. Take a look at the Optional section that will help you with wiring extra LEDs and other additional modifications. Thanks for building this robot! We would love to see a picture and hear a story! Please email us with a picture or story at Robots@egrobotics.com

Optional Additions and Modifications

Add LEDS

Add LEDs to trick it out! The Resistors used here are 220 Ohm Resistors.



Robot Seek Light, Attack, and Avoid

This code will allow the robot to seek out light and when it detects an object in the way it will first swing the servo and then it will turn to avoid the object.

Code:

(<http://www.egrobotics.com/instructions/code/robotattackavoid.ino>)

```
#include <Servo.h>
Servo myservo;
#define IRsensorPin 11
#define IRledPin 10
const int RightSensor = 2;
const int LeftSensor = 0;
int IR;
int SensorLeft;
int SensorRight;
int SensorDifference;
void IR38Write() {
  for(int i = 0; i <= 384; i++) {
    digitalWrite(IRledPin, HIGH);
    delayMicroseconds(13);
    digitalWrite(IRledPin, LOW);
    delayMicroseconds(13);
  }
}
void setup() {
  myservo.attach(5);
  pinMode(IRledPin, OUTPUT);
```

```

digitalWrite(IRledPin, LOW);
pinMode(8, OUTPUT);
pinMode(9, OUTPUT);
pinMode(LeftSensor, INPUT);
pinMode(RightSensor, INPUT);
Serial.begin(9600);
Serial.println(" \nBeginning Light Seeking Behavior");
}
void loop() {
  IR38Write();
  IR = digitalRead(IRsensorPin);
  delay(50);
  SensorLeft = 1023 - analogRead(LeftSensor);
  delay(1);
  SensorRight = 1023 - analogRead(RightSensor);
  delay(1);
  SensorDifference = abs(SensorLeft - SensorRight);
  Serial.print("Left Sensor = ");
  Serial.print(SensorLeft);
  Serial.print("\t");
  Serial.print("Right Sensor = ");
  Serial.print(SensorRight);
  Serial.print("\t");
  if (SensorLeft > SensorRight && SensorDifference > 75 && IR == HIGH) {
    Serial.println("Left");
    digitalWrite(8, HIGH);
    delay(250);
    digitalWrite(8, LOW);
    delay(100);
  }
  if (IR == LOW){
    myservo.write(130);
    delay(200);
    myservo.write(180);
    delay(200);
    digitalWrite(8, LOW);
    digitalWrite(9, HIGH);
    delay(800);
    digitalWrite(8, LOW);
    digitalWrite(9, LOW);
    delay(250);
  }
  if (SensorLeft < SensorRight && SensorDifference > 75 && IR == HIGH) {
    digitalWrite(9, HIGH);
    delay(250);
    digitalWrite(9, LOW);
    delay(100);
  }
  else if (SensorDifference < 75 && IR == HIGH) {
    Serial.println("Forward");
    digitalWrite(8, HIGH);
    digitalWrite(9, HIGH);
    delay(500);
    digitalWrite(8, LOW);
    digitalWrite(9, LOW);
    delay(250);
  }
  Serial.print("\n");
}

```

Robot Designs

Here are some pictures of various robots designed with the LDR/IR Beginner Robotics Kit.

