

The Arduino Mothbot

by [natantus](#) on June 14, 2009

Table of Contents

intro: The Arduino Mothbot	2
step 1: Gather your Parts and Tools	2
step 2: The Planning Stage	2
step 3: Connecting the Servos to the Arduino	3
step 4: Testing out the Motors	5
step 5: Integrating the On/Off Button	6
step 6: Testing the On/Off Button	8
step 7: Integrating the Light Sensors	10
step 8: The Final Code	11
step 9: Build the Mothbot Body	13
step 10: Making the Wheels	14
step 11: Completing the Arduino Mothbot	15
Related Instructables	16
Advertisements	16
Comments	16

intro: The Arduino Mothbot

The purpose of this project is to design and build a simple light-following robot using an [Arduino Duemilanove](#) microcontroller board. I really wanted to share a robot project that was cheap, simple to build, and had a complete set of instructions for all of the different steps. I hope I've succeeded and I'd love to get comments about making this instructable even better.

The design of this robot focused around using the book "[Getting Started with Arduino](#)" by Massimo Banzi and published with [makezine.com Make]. I also employed code for running the servos from a project titled: [How to Make an Arduino Controlled Servo Robot \(SERB\)](#).

The Arduino Mothbot is in total a pretty quick robot to build. Assuming you start with all of the parts and don't have to improvise, the project in total should take maybe an hour to build. That is if you follow the instructions and copy the code. However, if you build only one feature at a time and test along the way then this project could take significantly longer. The advantage of the longer track is that you'll probably learn a lot more and have some fun along the way.

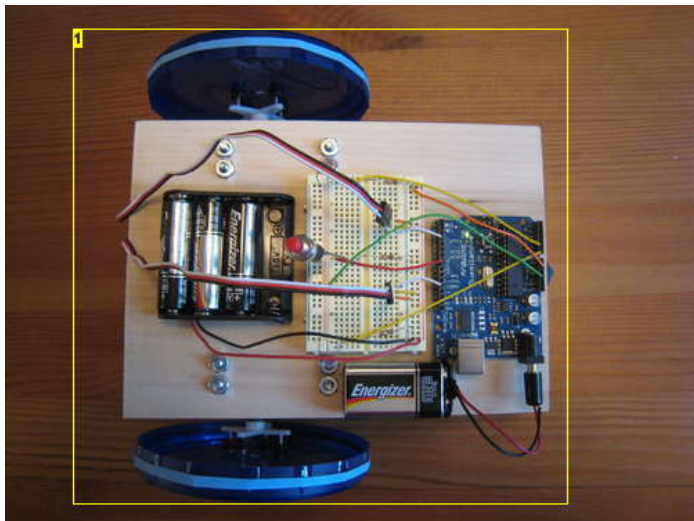


Image Notes

1. The Arduino Mothbot

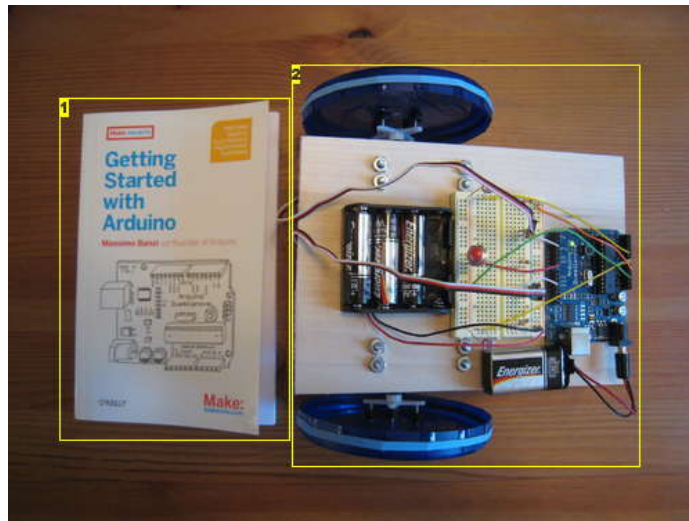


Image Notes

1. Getting Started with Arduino - The book that inspired much of this project
2. Mothbot just hangin' out

step 1: Gather your Parts and Tools

Building this robot is going to cost you roughly \$80 in parts if you've never done anything like this before. The cost for me was significantly less since I've got a lot of electronics lying around to work from. However, I know how frustrating it can be to try and follow an instructable without knowing which parts to get, where to order from, and how much everything will cost up front so I've done all that work for you. Once you've got the parts all squared away it should be a snap to do this project. Follow the following link to my project wiki to get a complete parts list.

[Arduino Mothbot Parts List](#)

Now you may want to get some tools. Since this project employs a solderless breadboard you can do without a lot of fancy electronics equipment. Hopefully you can find the rest of the things you need in a garage:

1. Needle nose pliers
2. Wire Cutters
3. Flat head screw driver
4. Small Phillips (4-sided) screw driver
5. Adjustable wrench or 11/32" hex wrench
6. Drill
7. 1/16", 5/32" and 7/32" drill bits
8. Saw (optional)
9. Safety Goggles

Please use safe practices when using any power tools.

step 2: The Planning Stage

Before I started this project I looked around Instructables at a lot of other projects. I also spent some time reading the book "Getting Started with Arduino" by Massimo Banzi. Nearly everything in this project is done from an example on this website or in the book. I designed the project this way in an attempt to make it accessible to the novice roboticist.

In my planning phase I did not only look at hardware and coding but did my electronics homework as well. I wanted to draw up a simple electronics schematic for this project so I could follow what was going on as I built it. You can see in the picture the different components, power lines, and the Arduino pins. Hopefully it's a clear diagram and also illustrates how simple the electronics for this project is.

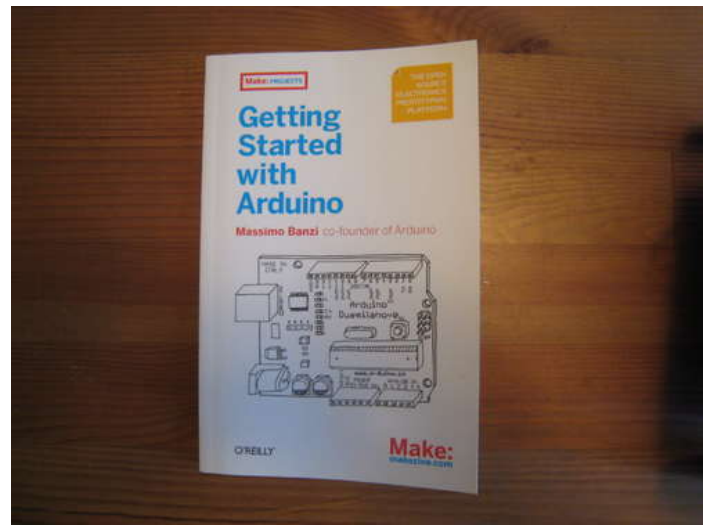
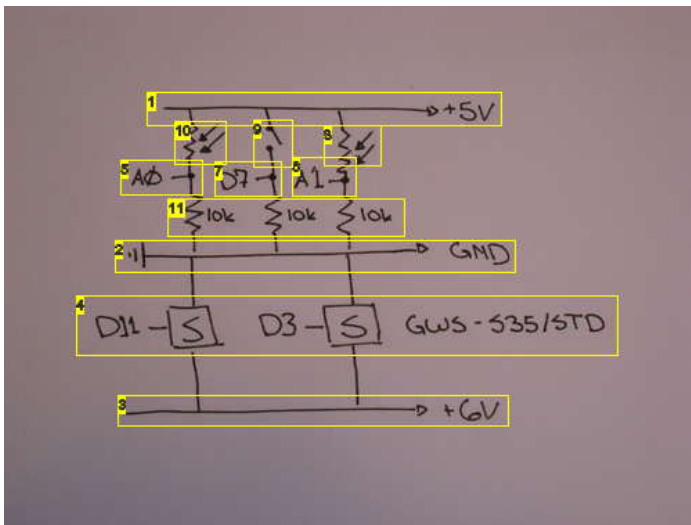


Image Notes

1. +5V power line for buttons and sensors
2. The common ground (GND) for the electronics
3. +6V power line for the servo motors
4. GWS S35/STD servos connected to digital pins 11 and 3 on the Arduino
5. Connection to analog pin 0 on the Arduino
6. Connection to analog pin 1 on the Arduino
7. Connection to digital pin 7 on the Arduino
8. Photoresistor
9. Pushbutton
10. Photoresistor
11. 10k Ohm resistors

step 3: Connecting the Servos to the Arduino

If you're going to build a robot the first thing you probably want to work out is how to get it moving around. Most likely you want to be able to send it forward, backward, right, left and make it stop. If you can't figure out how to command it to move properly you're not likely to be able to make it do anything when you connect all the sensors. Below are the steps to connect the motor to the Arduino.

1. The first thing to do when setting up the solderless breadboard is to set up the ground (GND) and power (+6V) for the servos. I chose to use the two long strips on the board that would be closest to the Arduino.
2. Once the ground and power lines are identified connect the ground of the Arduino board to the ground strip on the solderless breadboard. Do not connect the power to the solderless breadboard yet.
3. Each servo has three wires that come out of them. Mine have a black, red, and white wire for each. The black is for ground, the red is for power, and the white is the control wire. Cut three jumper wires for each servo of the same size (so 6 in total).
4. Attach the jumper wires to the end of the servo wires and then each servo to the solderless breadboard.
5. Now use jumpers to connect the ground and power from each servo to the ground and power of the solderless breadboard.
6. Now connect the control wires from each servo to the Arduino. Connect the left servo to digital output (PWM) 3 and the right servo to digital output (PWM) 11.
7. Finally, connect the ground and power from the 4AA batteries to the solderless breadboard ground and power. Don't be alarmed if the servos start moving when your Arduino has no power or is not yet programed.
8. Using the code you should now be able to run the motors in the forward, backward, left or right directions using the included functions.

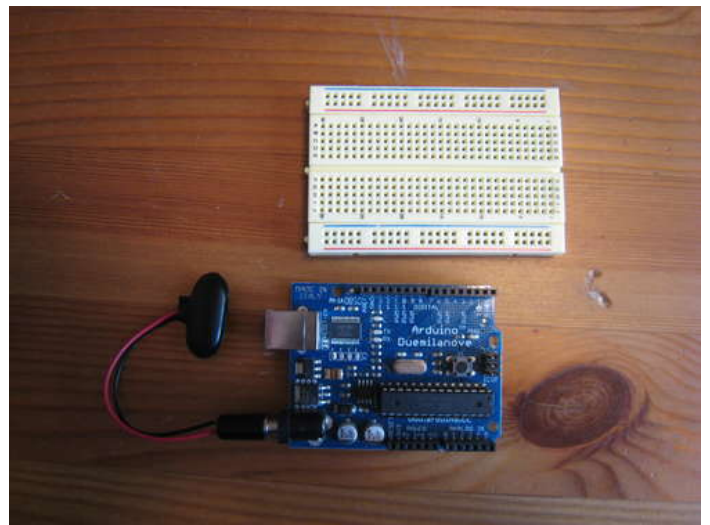
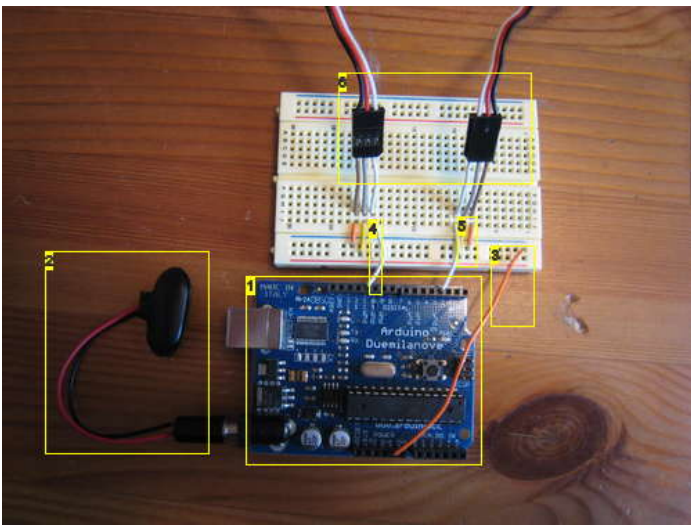
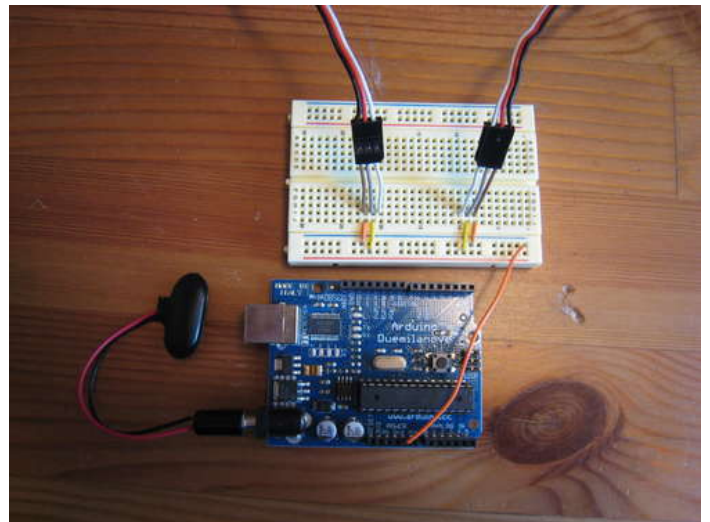
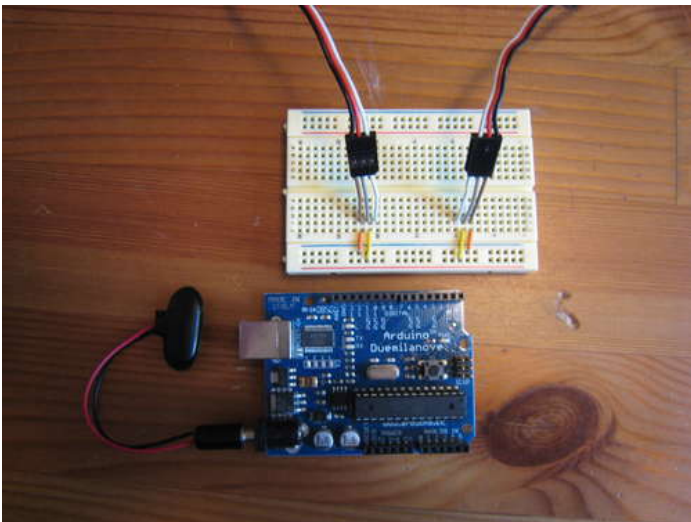
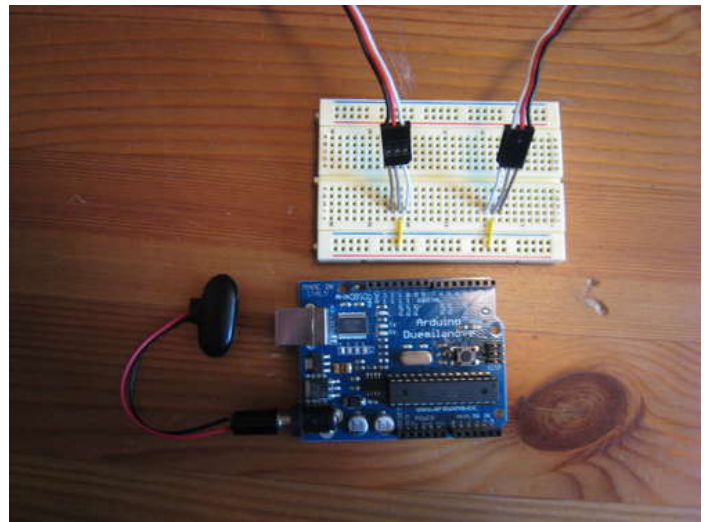
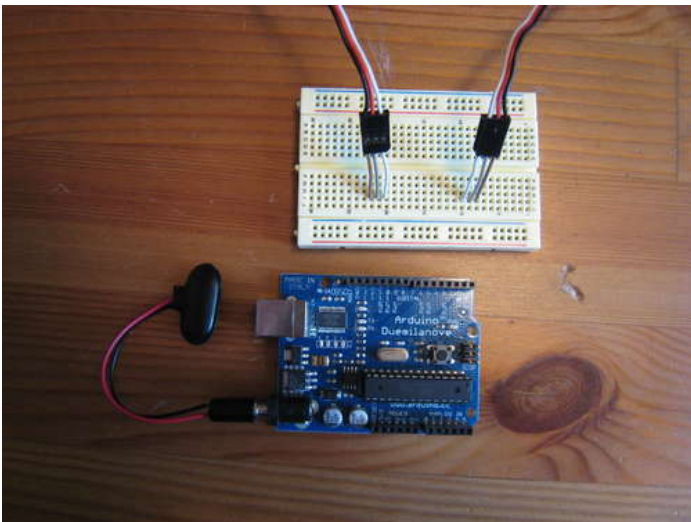


Image Notes

1. Arduino Duemilanove
2. 9V battery connector
3. Ground wire connected to Arduino
4. Signal line to servo
5. Power and Ground to servo, notice they connect to separate bars of the solderless breadboard
6. The servos connected by jumper wires. Use different colors if possible to distinguish power, ground, and signal lines



step 4: Testing out the Motors

I think it's important to include some of the test code I used when putting together the Arduino Mothbot. If you're interested and willing to put in the time to tinker around I think you'll find these code snippets educational and useful in other projects.

Before I post any code below I want to make it known that the following is based on another great project called [How to Make an Arduino Controlled Servo Robot \(SERB\)](#). I learned a lot from following the work on that instructable and want to give credit where it is due.

```
/*
 * Arduino Mothbot
 *
 * Digital Pin Wiring:
 *   pin 11 - Right Servo Signal
 *   pin 3 - Left Servo Signal
 *
 * License: This work is licenced under the Creative Commons
 *   Attribution-Share Alike 3.0 Unported License. To
 *   view a copy of this licence, visit
 *   http://creativecommons.org/licenses/by-sa/3.0/
 *   or send a letter to Creative Commons, 171 Second
 *   Street, Suite 300, San Francisco, California 94105,
 *   USA.
 */

//-----
// START OF ARDUINO MOTHBOT SETUP
//--- Library
#include <Servo.h>

//--- Pin Definitions
#define LEFTSERVOPIN 3 //The digital pin that the left servo is connected to
#define RIGHTSERVOPIN 11 //The digital pin that the right servo is connected to

//--- Servo Setup
Servo leftServo;
Servo rightServo;

//--- Speed Setup
int robotSpeed = 75; //set the speed of the robot
int rightSpeed = 50;
int leftSpeed = 50;

//--- Delay Threshold
int delayParam = 10; //Supported Times - 0 - 255 (0 to 25.5 Seconds) value * 100 milliseconds

// END OF ARDUINO MOTHBOT SETUP
//-----

//-----
//START OF ARDUINO MOTHBOT PROGRAM
//--- The program setup
void setup()
{
  Serial.begin(9600); //Starts the serial port
  robotSetup(); //sets the state of all neccessary
                //pins and adds servos to your sketch
}

//--- The main program code
void loop()
{
  Serial.println("Forward");
  goForward();
  delay(delayParam * 100);
  goStop();
}

//END OF ARDUINO MOTHBOT PROGRAM
//-----

//-----
//START OF ARDUINO MOTHBOT FUNCTIONS

//--- The setup for the robot
void robotSetup(){
  //--- Set the speed of the robot
  setSpeed(robotSpeed);

  //--- Set up the servos
  pinMode(LEFTSERVOPIN, OUTPUT); //sets the left servo signal pin
                                 //to output
  pinMode(RIGHTSERVOPIN, OUTPUT); //sets the right servo signal pin
                                   //to output
  leftServo.attach(LEFTSERVOPIN); //attaches left servo
  rightServo.attach(RIGHTSERVOPIN); //attaches right servo

  //--- Tell the robot to stop the servos
  goStop();
}

//--- Set the speed of the robot between 0-(stopped) and 100-(full speed)
void setSpeed(int newSpeed){
  setSpeedLeft(newSpeed); //sets left speed
  setSpeedRight(newSpeed); //sets right speed
}

//--- Set the speed of the left wheel
void setSpeedLeft(int newSpeed){
  if(newSpeed >= 100) {newSpeed = 100;} //if speed is greater than 100
http://www.instructables.com/id/The-Arduino-Mothbot/

```

```

//make it 100
if(newSpeed <= 0) {newSpeed = 0;} //if speed is less than 0 make
//it 0
leftSpeed = newSpeed * 0.9; //between 0 and 90
}

//--- Set the speed of the right wheel
void setSpeedRight(int newSpeed){
  if(newSpeed >= 100) {newSpeed = 100;} //if speed is greater than 100
//make it 100
  if(newSpeed <= 0) {newSpeed = 0;} //if speed is less than 0 make
//it 0
  rightSpeed = newSpeed * 0.9; //scales the speed to be
}

//--- Move the robot forward
void goForward(){
  leftServo.write(90 + leftSpeed);
  rightServo.write(90 - rightSpeed);
}

//--- Move the robot backward
void goBackward(){
  leftServo.write(90 - leftSpeed);
  rightServo.write(90 + rightSpeed);
}

//--- Move the robot right
void goRight(){
  leftServo.write(90 + leftSpeed);
  rightServo.write(90 + rightSpeed);
}

//--- Move the robot left
void goLeft(){
  leftServo.write(90 - leftSpeed);
  rightServo.write(90 - rightSpeed);
}

//--- Stop the robot
void goStop(){
  leftServo.write(90);
  rightServo.write(90);
}

//END OF ARDUINO MOTHBOT FUNCTIONS
//-----

```

step 5: Integrating the On/Off Button

Now you may be wanting to turn your robot on and off with the push of a button. The Arduino itself will run code in an endless loop until you unplug it, which can be quite frustrating when you plug in your robot on the table and it starts running away from you! Integrating the button is a great step in this process because you'll also learn how to use buttons for other things, like creating a bumper to detect when the robot hits a wall.

As a note, you'll notice I removed the servos from the solderless breadboard for most of my pictures. This just helps make the image more clear when I'm showing different steps.

1. To begin, disconnect power from the servo motors before doing any more work. Remember to do this every time you add something to this project.
2. Now you may want to be able to turn your robot on and off as opposed to having the robot immediately start moving when you connect the power.
3. Identify a strip on the opposite side of the solderless breadboard to be power for on/off button (and later the sensors).
4. Using a long jumper wire connect the power (+5V) from the Arduino to the strip you just identified.
5. Connect two jumper wires to the momentary switch and plug one end into the (+5V) power
6. Plug the other end of the momentary switch into a smaller strip in the middle of the solderless breadboard.
7. From that same strip connect a 10K ohm resistor to the strip and the other end into ground
8. Finally, connect a wire from the strip with the switch and the resistor on one end and place the other end in digital input 7 on the Arduino.
9. Now, with the code you should be able to use the button to turn on and off the robot. If you use the code with the LED (digital output 13) you will see the on-board LED turn on and off with the robot. This is a great way to test the Arduino code if you have the power to the motors disconnected.

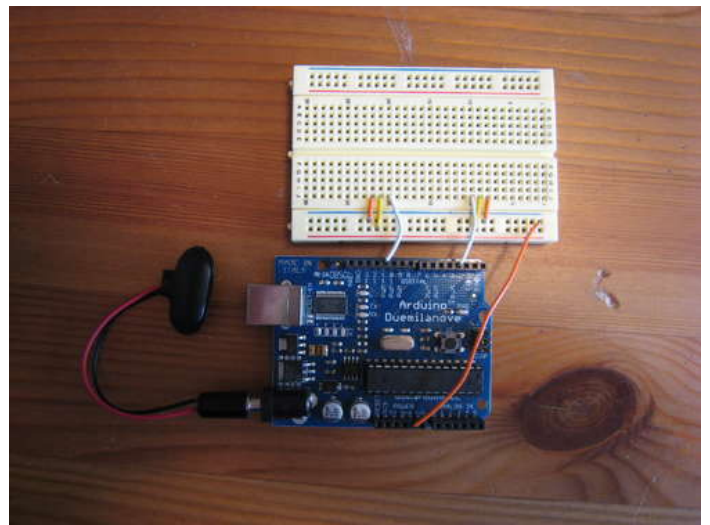
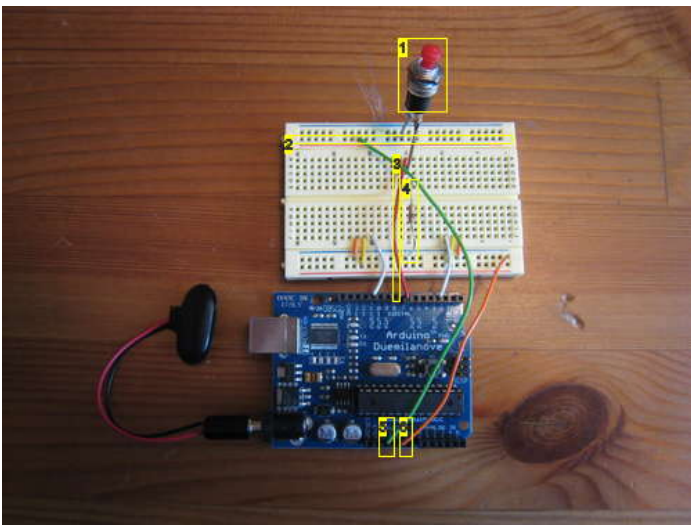
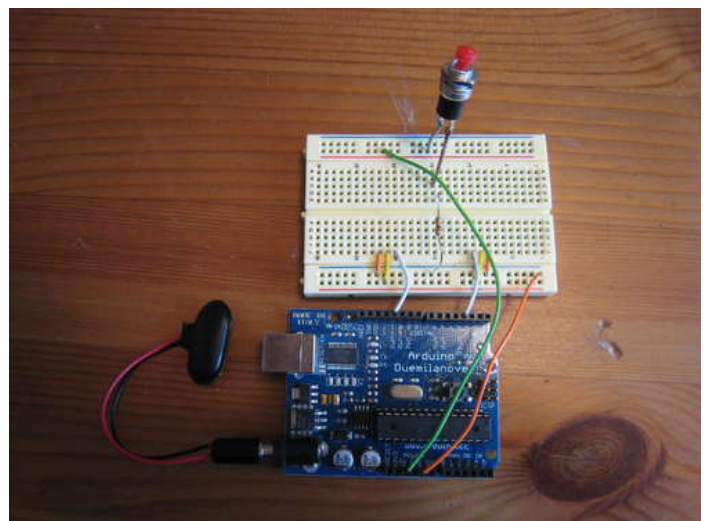
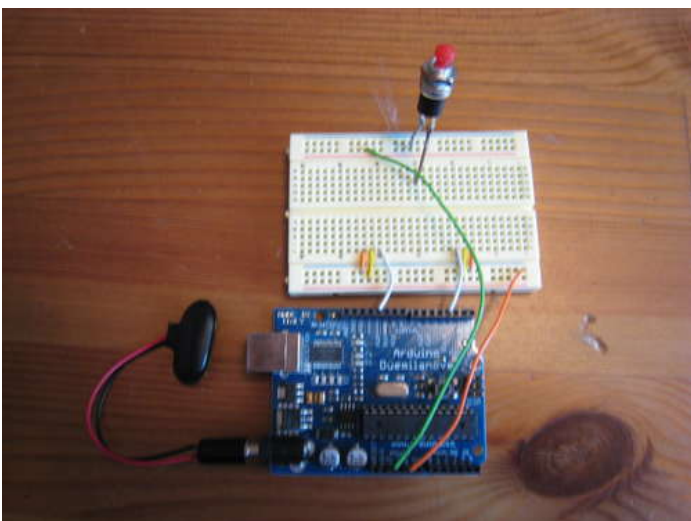
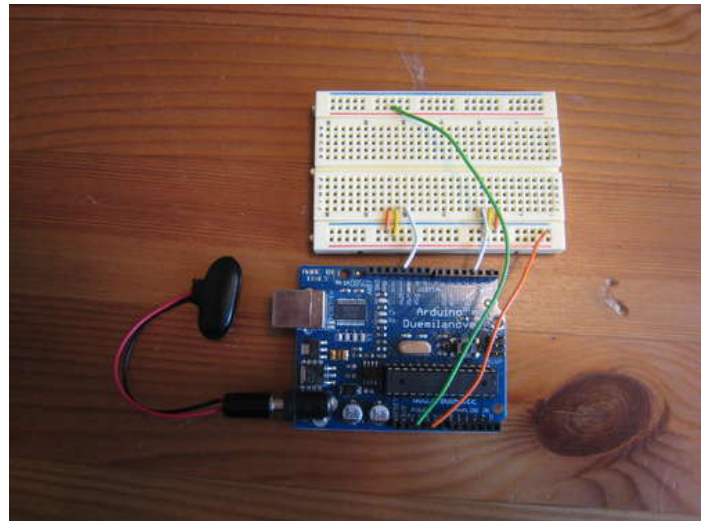
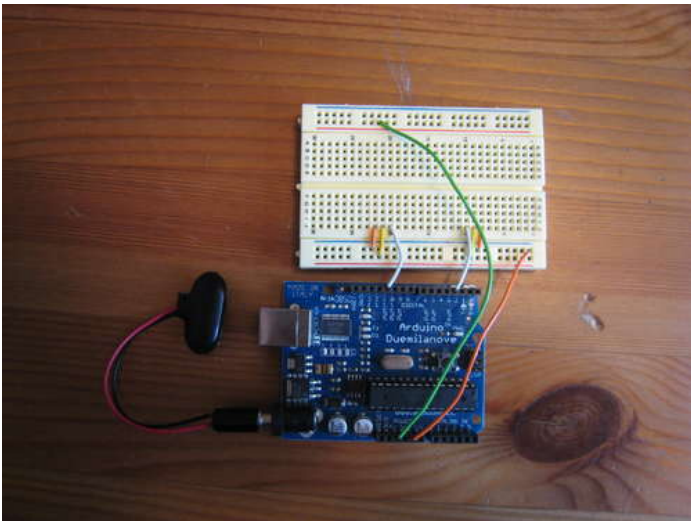
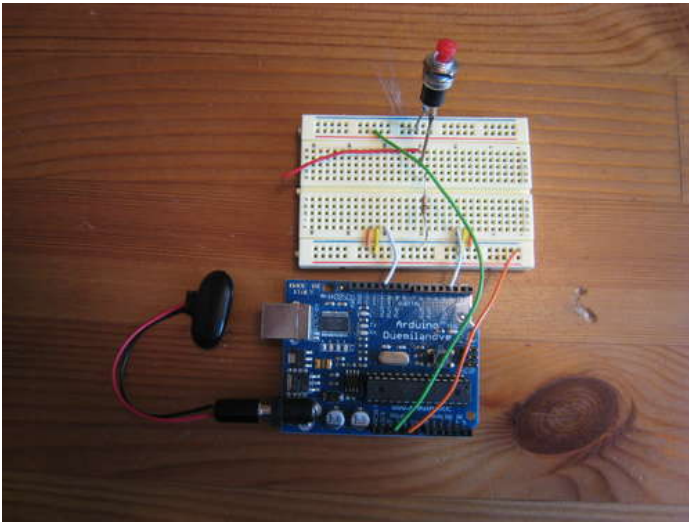


Image Notes

1. Pushbutton connected to +5V and center strip
2. +5V power line from Arduino
3. Signal wire to Arduino digital pin 7
4. 10k Ohm resistor connected to center strip and ground
5. +5V power line
6. Common ground





step 6: Testing the On/Off Button

This new code includes the information for using the On/Off button and making the onboard LED blink.

```

/*
 * Arduino Mothbot
 *
 * Digital Pin Wiring:
 *   pin 7 - Pushbutton Signal
 *   pin 11 - Right Servo Signal
 *   pin 3 - Left Servo Signal
 *
 * License: This work is licenced under the Creative Commons
 * Attribution-Share Alike 3.0 Unported License. To
 * view a copy of this licence, visit
 * http://creativecommons.org/licenses/by-sa/3.0/
 * or send a letter to Creative Commons, 171 Second
 * Street, Suite 300, San Francisco, California 94105,
 * USA.
 */

//-----
// START OF ARDUINO MOTHBOT SETUP
//--- Library
#include <Servo.h>

//--- Pin Definitions
#define LEFTSERVOPIN 3 //The digital pin that the left servo is connected to
#define RIGHTSERVOPIN 11 //The digital pin that the right servo is connected to

#define PUSHBUTTON 7 //The digital pin that the pushbutton sensor is connected to
#define LED 13 //The LED on the arduino

//--- Servo Setup
Servo leftServo;
Servo rightServo;

//--- Speed Setup
int robotSpeed = 75; //set the speed of the robot
int rightSpeed = 50;
int leftSpeed = 50;

//--- On/Off values
int pushButtonVal = 0;
int old_pushButtonVal = 0;
int pushButtonState = 0;

//--- Delay Threshold
int delayParam = 10; //Supported Times - 0 - 255 (0 to 25.5 Seconds) value * 100 milliseconds

// END OF ARDUINO MOTHBOT SETUP
//-----

//-----
//START OF ARDUINO MOTHBOT PROGRAM
//--- The program setup
void setup()
{
  Serial.begin(9600); //Starts the serial port
  robotSetup(); //sets the state of all necessary
                //pins and adds servos to your sketch
}

//--- The main program code
void loop()
{
  //--- Get information from the pushbutton sensor
  pushButtonVal = digitalRead(PUSHBUTTON);

```

```

//--- Set the state of the pushbutton
if((pushButtonVal == HIGH) && (old_pushButtonVal == LOW)){
  Serial.println("Button Pushed");
  pushButtonState = 1 - pushButtonState;
  delay(10); // Simple de-bouncing
}

//--- Reset the state of the push button
old_pushButtonVal = pushButtonVal;

//--- If pushbutton state HIGH then turn on the robot and
// listen to the light sensor information
if(pushButtonState == 1){
  digitalWrite(LED, HIGH); // turn LED ON
  Serial.println("Forward");
  goForward();
  delay(delayParam * 100);
  goStop();
}
}
//END OF ARDUINO MOTHBOT PROGRAM
//-----
//-----
//START OF ARDUINO MOTHBOT FUNCTIONS

//--- The setup for the robot
void robotSetup(){
  //--- Set the speed of the robot
  setSpeed(robotSpeed);

  //--- Set up the servos
  pinMode(LEFTSERVOPIN, OUTPUT); //sets the left servo signal pin
  //to output
  pinMode(RIGHTSERVOPIN, OUTPUT); //sets the right servo signal pin
  //to output
  leftServo.attach(LEFTSERVOPIN); //attaches left servo
  rightServo.attach(RIGHTSERVOPIN); //attaches right servo

  //--- Set up the pushbutton
  pinMode(PUSHBUTTON, INPUT); //sets the pushbutton sensor pin to input

  //--- Set up the LED to see robot state
  pinMode(LED, OUTPUT); //sets the led pin as output

  //--- Tell the robot to stop the servos
  goStop();
}

//--- Set the speed of the robot between 0-(stopped) and 100-(full speed)
void setSpeed(int newSpeed){
  setSpeedLeft(newSpeed); //sets left speed
  setSpeedRight(newSpeed); //sets right speed
}

//--- Set the speed of the left wheel
void setSpeedLeft(int newSpeed){
  if(newSpeed >= 100) {newSpeed = 100;} //if speed is greater than 100
  //make it 100
  if(newSpeed <= 0) {newSpeed = 0;} //if speed is less than 0 make
  //it 0
  leftSpeed = newSpeed * 0.9; //between 0 and 90
}

//--- Set the speed of the right wheel
void setSpeedRight(int newSpeed){
  if(newSpeed >= 100) {newSpeed = 100;} //if speed is greater than 100
  //make it 100
  if(newSpeed <= 0) {newSpeed = 0;} //if speed is less than 0 make
  //it 0
  rightSpeed = newSpeed * 0.9; //scales the speed to be
}

//--- Move the robot forward
void goForward(){
  leftServo.write(90 + leftSpeed);
  rightServo.write(90 - rightSpeed);
}

//--- Move the robot backward
void goBackward(){
  leftServo.write(90 - leftSpeed);
  rightServo.write(90 + rightSpeed);
}

//--- Move the robot right
void goRight(){
  leftServo.write(90 + leftSpeed);
  rightServo.write(90 + rightSpeed);
}

//--- Move the robot left
void goLeft(){
  leftServo.write(90 - leftSpeed);
  rightServo.write(90 - rightSpeed);
}

//--- Stop the robot
void goStop(){

```

```
leftServo.write(90);
rightServo.write(90);
}
```

```
//END OF ARDUINO MOTHBOT FUNCTIONS
```

```
//-----
```

step 7: Integrating the Light Sensors

What would an Arduino Mothbot be if it didn't have light sensors? The point of this simple project is to make a robot that is attracted to the brightest light. For this we'll need to integrate some light sensors, also known as photo-resistors.

1. Again, disconnect the power from the servo motors before doing this step
2. The setup for the light sensors will be done twice. It is almost the exact same setup as the momentary switch. Actually, it is the same setup, but this time you will use the light sensor (photo-resistor) instead of a momentary switch.
3. Because this robot will use the two light sensors to pick a direction to drive it is recommended that you set up each light sensor on opposite sides of the solderless breadboard or as far apart as possible.
4. Connect one end of a light sensor to the (+5V) power line and the other end into a small strip in the center of the board.
5. Connect a 10k ohm resistor to that same strip and the other end to ground
6. Now connect a jumper wire from the small strip (where the photo resistor and regular resistor are connected) and plug the other end into an analog input.
7. Connect the left sensor to analog input 0 on the Arduino and the right sensor to analog input 1.
8. You should now be able to use the light sensors to move the servos.

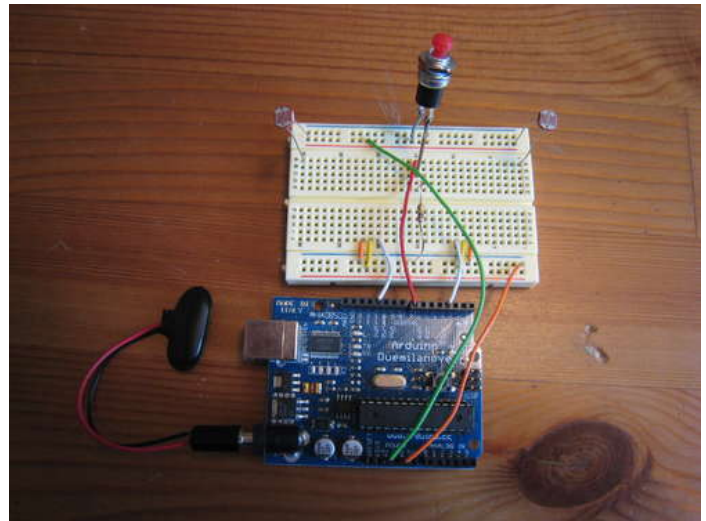
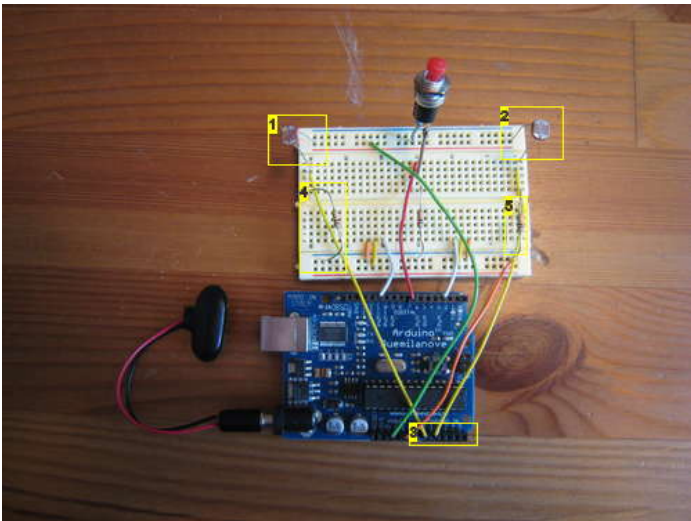
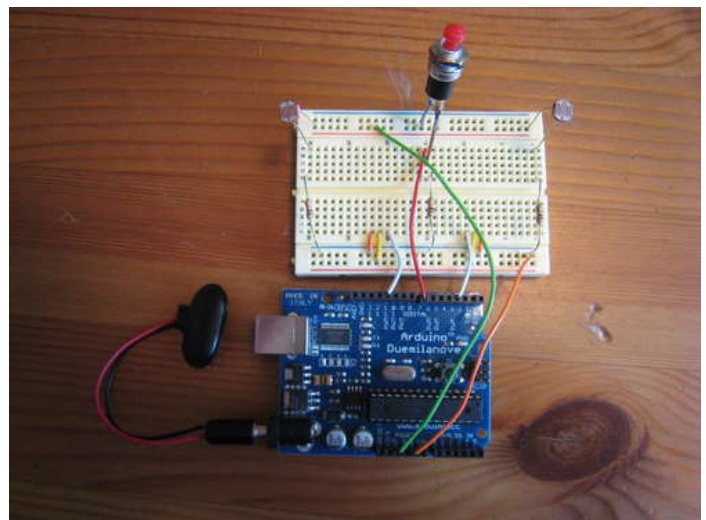
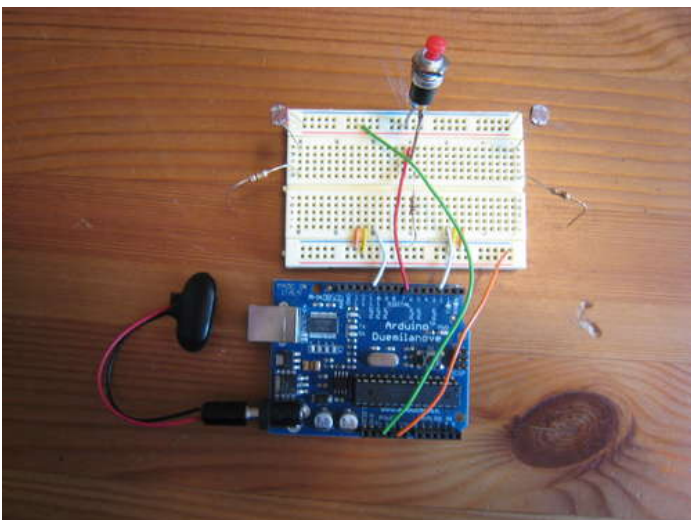
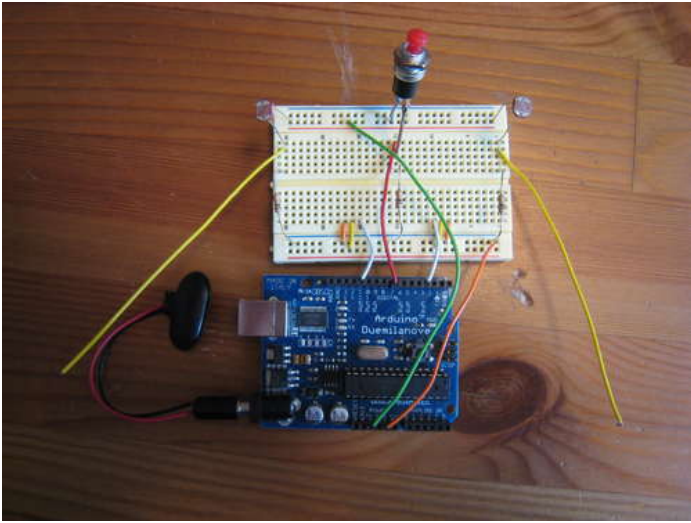


Image Notes

1. Left light sensor connected to +5V and center strip
2. Right light sensor connected to +5V and center strip
3. Light sensors connected to analog pins 0 and 1 on the Arduino
4. 10k Ohm resistor connected to center strip and common ground
5. 10k Ohm resistor connected to center strip and common ground





step 8: The Final Code

Here is the final code used to run the Arduino Mothbot. In the code I've included print statements to the Arduino serial port. If you have the Arduino connected through the USB port of your computer you should be able to see the print statements that tell you which way the robot is planning to go. You may want to adjust the light sensor threshold value to fine tune the behavior of the robot. The threshold depends mainly on your sensors and the ambient light of the location you are in.

```

/*
 * Arduino Mothbot
 *
 * Digital Pin Wiring:
 *   pin 7 - Pushbutton Signal
 *   pin 11 - Right Servo Signal
 *   pin 3 - Left Servo Signal
 *
 * Analog Pin Wiring:
 *   pin 0 - Left Light Sensor Signal
 *   pin 1 - Right Light Sensor Signal
 *
 * License: This work is licenced under the Creative Commons
 * Attribution-Share Alike 3.0 Unported License. To
 * view a copy of this licence, visit
 * http://creativecommons.org/licenses/by-sa/3.0/
 * or send a letter to Creative Commons, 171 Second
 * Street, Suite 300, San Francisco, California 94105,
 * USA.
 */

//-----
// START OF ARDUINO MOTHBOT SETUP
//--- Library
#include <Servo.h>

//--- Pin Definitions
#define LEFTSERVOPIN 3 //The digital pin that the left servo is connected to
#define RIGHTSERVOPIN 11 //The digital pin that the right servo is connected to

#define LEFTLIGHTSENSOR 0 //The analog pin the left light sensor is connected to
#define RIGHTLIGHTSENSOR 1 //The analog pin the right light sensor is connected to

#define PUSHBUTTON 7 //The digital pin that the pushbutton sensor is connected to

#define LED 13 //The LED on the arduino

//--- Servo Setup
Servo leftServo;
Servo rightServo;

//--- Speed Setup
int robotSpeed = 75; //set the speed of the robot
int rightSpeed = 50;
int leftSpeed = 50;

//--- Sensor Values
int leftLightSensorVal = 0;
int rightLightSensorVal = 0;
int lightSensorThreshold = 200;

//--- On/Off values
int pushButtonVal = 0;
int old_pushButtonVal = 0;
int pushButtonState = 0;

//--- Delay Threshold
int delayParam = 10; //Supported Times - 0 - 255 (0 to 25.5 Seconds) value * 100 milliseconds

// END OF ARDUINO MOTHBOT SETUP
//-----

```

```

//-----
//START OF ARDUINO MOTHBOT PROGRAM
//--- The program setup
void setup()
{
  Serial.begin(9600);           //Starts the serial port
  robotSetup();                //sets the state of all necessary
                               //pins and adds servos to your sketch
}

//--- The main program code
void loop()
{
  //--- Get information from the light sensor
  leftLightSensorVal = analogRead(LEFTLIGHTSENSOR);
  rightLightSensorVal = analogRead(RIGHTLIGHTSENSOR);

  //--- Get information from the pushbutton sensor
  pushButtonVal = digitalRead(PUSHBUTTON);

  //--- Set the state of the pushbutton
  if((pushButtonVal == HIGH) && (old_pushButtonVal == LOW)){
    Serial.println("Button Pushed");
    pushButtonState = 1 - pushButtonState;
    delay(10); // Simple de-bouncing
  }

  //--- Reset the state of the push button
  old_pushButtonVal = pushButtonVal;

  //--- If pushbutton state HIGH then turn on the robot and
  //      listen to the light sensor information
  if(pushButtonState == 1){
    digitalWrite(LED, HIGH); // turn LED ON
    if(leftLightSensorVal - rightLightSensorVal >= lightSensorThreshold){
      Serial.println("Left");
      goLeft();
      delay(delayParam * 100);
      goStop();
    } else if(rightLightSensorVal - leftLightSensorVal >= lightSensorThreshold){
      Serial.println("Right");
      goRight();
      delay(delayParam * 100);
      goStop();
    } else {
      Serial.println("Forward");
      goForward();
      delay(delayParam * 100);
      goStop();
    }
  } else {
    Serial.println("Stop");
    digitalWrite(LED, LOW); // turn LED OFF
    goStop();
  }
}
//END OF ARDUINO MOTHBOT PROGRAM
//-----

//-----
//START OF ARDUINO MOTHBOT FUNCTIONS

//--- The setup for the robot
void robotSetup(){
  //--- Set the speed of the robot
  setSpeed(robotSpeed);

  //--- Set up the servos
  pinMode(LEFTSERVOPIN, OUTPUT); //sets the left servo signal pin
                                  //to output
  pinMode(RIGHTSERVOPIN, OUTPUT); //sets the right servo signal pin
                                   //to output
  leftServo.attach(LEFTSERVOPIN); //attaches left servo
  rightServo.attach(RIGHTSERVOPIN); //attaches right servo

  //--- Set up the pushbutton
  pinMode(PUSHBUTTON, INPUT); //sets the pushbutton sensor pin to input

  //--- Set up the LED to see robot state
  pinMode(LED, OUTPUT); //sets the led pin as output

  //--- Tell the robot to stop the servos
  goStop();
}

//--- Set the speed of the robot between 0-(stopped) and 100-(full speed)
void setSpeed(int newSpeed){
  setSpeedLeft(newSpeed); //sets left speed
  setSpeedRight(newSpeed); //sets right speed
}

//--- Set the speed of the left wheel
void setSpeedLeft(int newSpeed){
  if(newSpeed >= 100) {newSpeed = 100;} //if speed is greater than 100
                                       //make it 100
  if(newSpeed <= 0) {newSpeed = 0;} //if speed is less than 0 make
                                     //it 0
  leftSpeed = newSpeed * 0.9; //between 0 and 90
}

```

```

//--- Set the speed of the right wheel
void setSpeedRight(int newSpeed){
  if(newSpeed >= 100) {newSpeed = 100;} //if speed is greater than 100
                                        //make it 100
  if(newSpeed <= 0) {newSpeed = 0;} //if speed is less than 0 make
                                        //it 0
  rightSpeed = newSpeed * 0.9; //scales the speed to be
}

//--- Move the robot forward
void goForward(){
  leftServo.write(90 + leftSpeed);
  rightServo.write(90 - rightSpeed);
}

//--- Move the robot backward
void goBackward(){
  leftServo.write(90 - leftSpeed);
  rightServo.write(90 + rightSpeed);
}

//--- Move the robot right
void goRight(){
  leftServo.write(90 + leftSpeed);
  rightServo.write(90 + rightSpeed);
}

//--- Move the robot left
void goLeft(){
  leftServo.write(90 - leftSpeed);
  rightServo.write(90 - rightSpeed);
}

//--- Stop the robot
void goStop(){
  leftServo.write(90);
  rightServo.write(90);
}

//END OF ARDUINO MOTHBOT FUNCTIONS
//-----

```

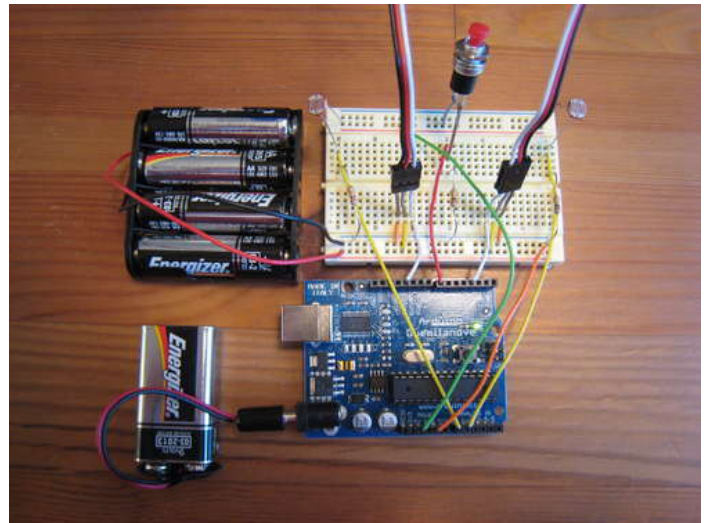
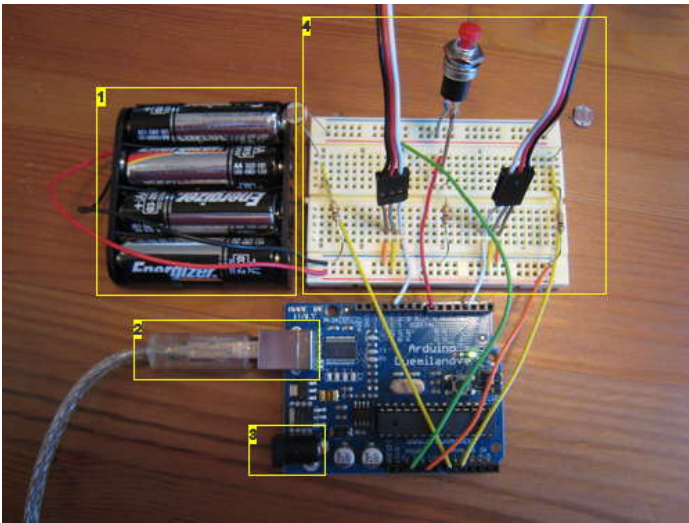


Image Notes

1. +6V power supply for servo motors
2. USB uploading the code to the arduino. This is disconnected when the robot is in operation
3. 9V power supply will power Arduino when robot is in operation
4. The completed circuit!

step 9: Build the Mothbot Body

The robot you're building is really no good unless it can hold itself together. For this reason it needs a body. I tried my best to make this as simple a construction project as possible. You are, however, going to have to do a little work on your own to figure out the right measurements. I suggest the age old "measure twice, cut once" method.

1. The body of the robot is made out of a small sheet of poplar wood I bought at the hardware store precut to 6" x 24". I cut mine down to 6" x 8" using the saw provided in the hardware store.
2. Next I drilled holes towards the front of the board to attach the servo brackets for each servo. For this I used an 5/32" size drill bit.
3. I also drilled in a hole at the rear of the board for the caster wheel that balances the robot. For this I used an 7/32" size drill bit. I chose to use a slightly smaller drill bit so I could get a tight friction fit with my caster wheel since I wasn't using a nut and bolt combination to attach it.
4. Then I attached the brackets to the board with the nuts and bolts. This was done using the flat head screw driver and the adjustable wrench.
5. After attaching the brackets I attached each servo to the brackets with the nuts and bolts.

<http://www.instructables.com/id/The-Arduino-Mothbot/>

6. Finally, I pushed the caster wheel into the whole.

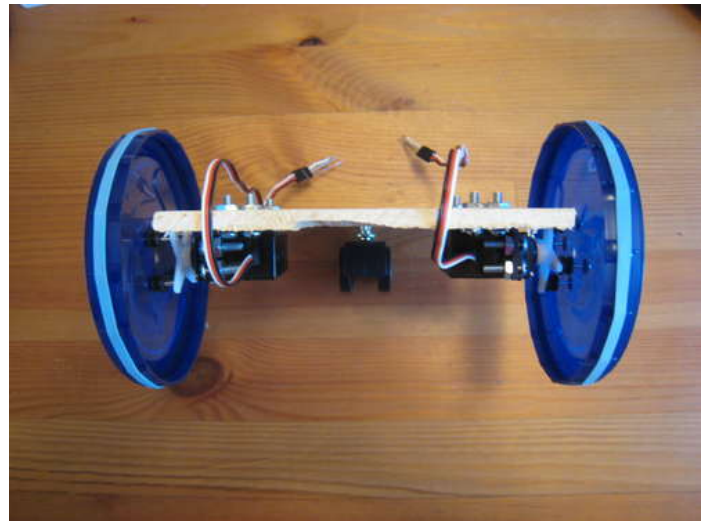
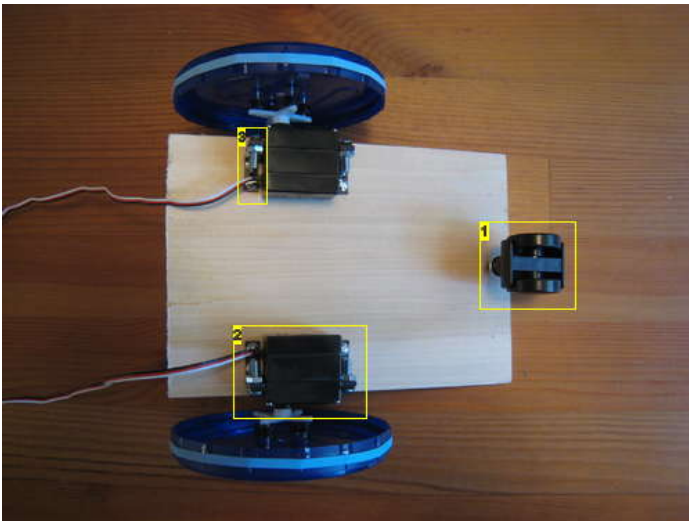


Image Notes

1. Caster wheel from plastic file drawer
2. Servo
3. Bracket connecting servo to wood board

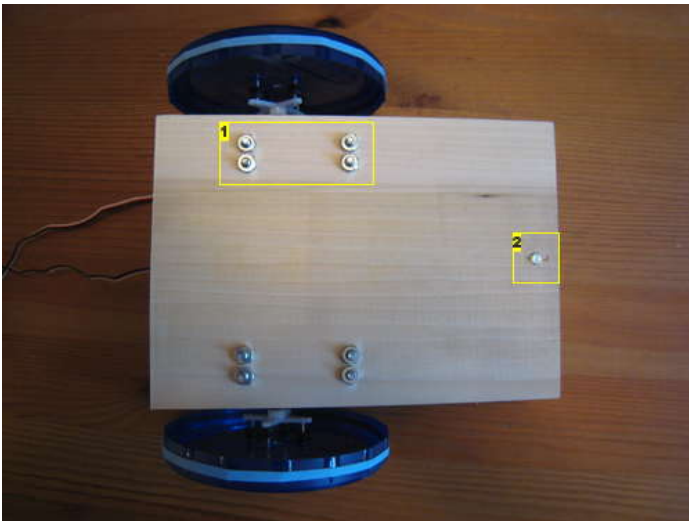


Image Notes

1. Brackets connect to board with nuts and bolts
2. Caster wheel was friction fitted here

step 10: Making the Wheels

The wheels were a tricky problem for me. I had actually bot some certified robot wheels but realized they were a) too heavy and b) I had no way to attach them to my chosen servos. That's when I remembered using jar lids in high school for a similar project. So it was off to the store in search of a suitable robot wheel alternative.

Each wheel is made from the lid from a Ziploc Twist 'n Loc container. Other good lids are those on peanut butter jars or other food goods. I don't advocate wasting food but save up your lids and you might find one is the right size for your robot project. I used the leftover containers to hold parts I've collected.

1. The first thing I did was pick the servo horn I wanted for the wheels. I picked the ones that had four horns and that were included with my servos when I bought them.
2. Before you do anything, drill a hole in the center of the wheel. I recommend doing this with your 5/32" drill bit. You'll need this so you can get to the screw that connects the horn to the servo.
4. Now screw the lid to the horn. I used four included screws with each servo to connect the lids to the horns. It might be easier if you pre-drill tiny holes through the lid like I did. I used a 1/16" drill bit for this. But be careful, drilling through this plastic with a heavy drill and a tiny bit can be difficult.
5. Now connect the horns to the servos using the small Phillips (4-sided) screw driver.
6. Finally, wrap rubber bands around each wheel to give you more traction. I got my rubber bands from produce I bought at the grocery store. Hopefully you have a few lying around.
7. At this point the entire body and wheels should be assembled.

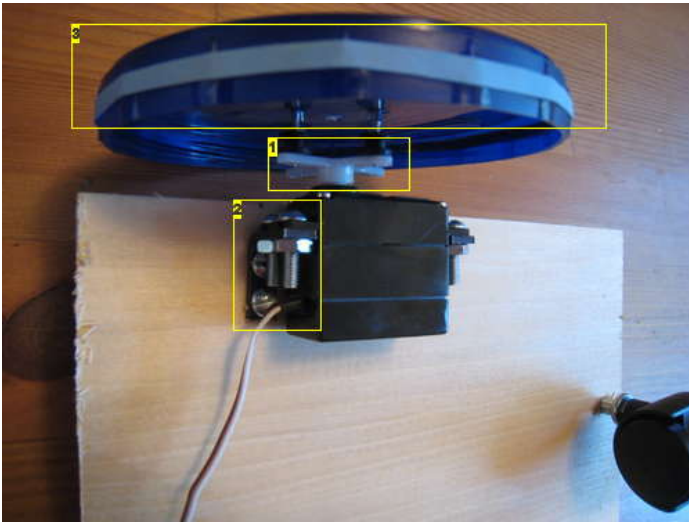


Image Notes

1. Servo horn that came with servo
2. Better view showing how nuts and bolts connect servo, bracket, and board together
3. Ziploc container lid with rubber band for friction

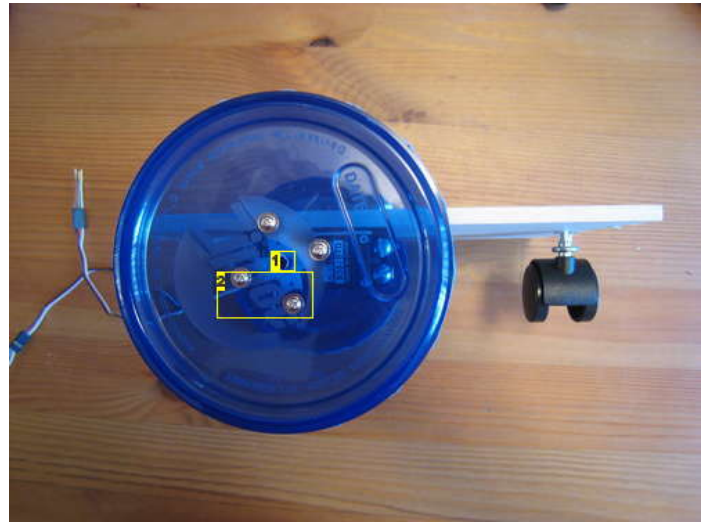


Image Notes

1. Hole drilled to give access to screw that mounts wheel to servo
2. Screws connecting wheel to servo horn

step 11: Completing the Arduino Mothbot

With the body and wheels assembled it's easy to place the Arduino and solderless breadboard just atop the robot body. Make sure you can still reach the USB input on the Arduino in case you need to change the programming. I used some black electrical tape underneath each to stick them to the body. Electrical tape is easy to remove and holds quite well.

1. Tape the Arduino and solderless breadboard to the top of the robot body that you've built.
2. Using tape again it's a good idea to connect the 4AA battery holder and the 9V battery to the body. Make sure the wires reach.
3. Connect the servo wires to the solderless breadboard if you had removed them previously.
4. Connect the Arduino power
5. Connect the servo motor power
6. Now place your robot on the ground and press the on/off switch! It should now come to life and chase the light around the room:)

As a future add-on project I would include a simple bumper or wall sensor. This would be a switch, much like the On/Off button used in this project. However, when the button was pushed it would tell the robot to reverse direction, turn left or right, and continue with the program. Once that is completed this robot would be a great little testing platform for other sensors and devices.

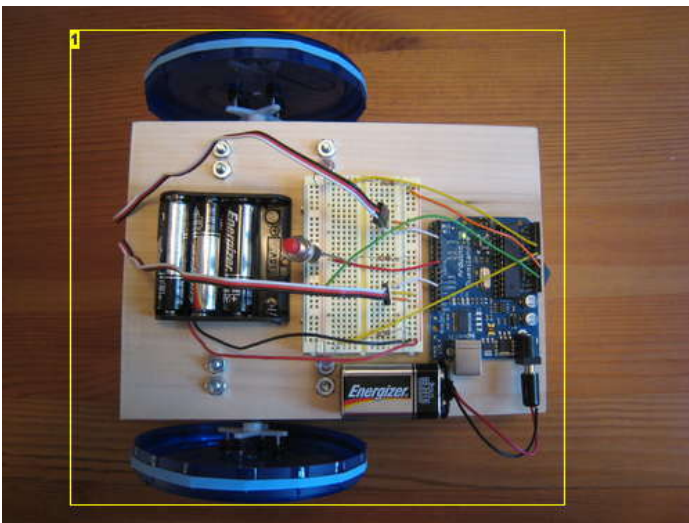


Image Notes

1. The Arduino Mothbot

Related Instructables



How To Modify A Hitec HS-65HB Servo W/Kryptonite Gears For Continuous Rotation by Zeitron



Easy DIY Home Automation (using servo switches) by oomlout



How to Make an Obstacles Avoiding Robot- Arduino Style by robotkid249



My Arduino Line Following Robot !!! - With Pololu QTR-6A IR Reflectance Sensor Array (video) by DuFFxP93



Arduino Controlled Servo Robot (SERB) by oomlout



Robotic Talking Turret by RazorConcepts



Ard-e: The robot with an Arduino as a brain by imadami



simpleWalker: 4-legged 2-servo walking robot by edwindertien

Comments

7 comments

[Add Comment](#)



kyle.marsh says:

Jun 15, 2009. 3:40 PM [REPLY](#)

Great intro robot! Mayhaps I'll actually build the aquatic mothbot I dreamed up for my pool now....

The jerky motion of a robot that only moves straight or pivots tends to bug me (although it *is* a "moth" bot), so one possible extension of the mothbot is to make it turn, rather than pivot. To smooth out your mothbot, instead of making both wheels turn the same speed, try varying the speed based on the relative readings from the light sensors. For example:

--If the reading from the two sensors is the same (within a reasonable threshold) put both motors full ahead.

--If the reading from the right sensor is twice that of the left, set the left motor twice as fast as the right motor.

--Fiddle with the parameters until you get something you like. You may want to turn down your delay so you get faster updates and smoother performance.

Another possible extension that can help make your robot more robust to different environments is to add a potentiometer (variable resistor) to change the threshold you use on your light sensors on-the-go without having to reprogram your bot.

Thanks for the good project!



Paulys says:

Jun 15, 2009. 3:39 PM [REPLY](#)

Nice job!

Do you have any video of the robot in action?



Bongmaster says:

Jun 14, 2009. 5:03 PM [REPLY](#)

nicey documented :) the explanation of the light sensor code may help me with my bot (pretty much same thing but with bumper switches and h-bridges as well as light sensing :)



natantus says:

Jun 15, 2009. 8:30 AM [REPLY](#)

Glad you liked it! If my project helps you with your light sensor then my project was a success;) When you get yours done please let me know so I can take a look too.



Bongmaster says:

Jun 15, 2009. 9:26 AM [REPLY](#)

will do :)



oomlout says:

Jun 15, 2009. 6:39 AM [REPLY](#)

Love the DIY Arduino robot base, and great Instructable.

(Otis (our office SERB) is wondering if a play date can be scheduled)

..the oomlout team..



natantus says:

Jun 15, 2009. 8:32 AM [REPLY](#)

Thanks for the encouragement! The SERB instructable was a huge help in getting this project off the ground. I'll also see if I can't clear my calendar and send my new robot over for a play date sometime soon;)