technology workshop craft home food play outside costumes

DIY Bluetooth Phone Controlled BB-8 Droid with Arduino UNO

by **Imetomi** on December 25, 2015

Table of Contents

Author:Imetomi Thomas

My name is Thomas (Tamas) and I am 16 years old Hungarian guy. I love the electronics, and I just like to invent something new, that never was before. I want to show another point of view and new ideas with my instructables. Hope you like it!

Intro: DIY Bluetooth Phone Controlled BB-8 Droid with Arduino UNO

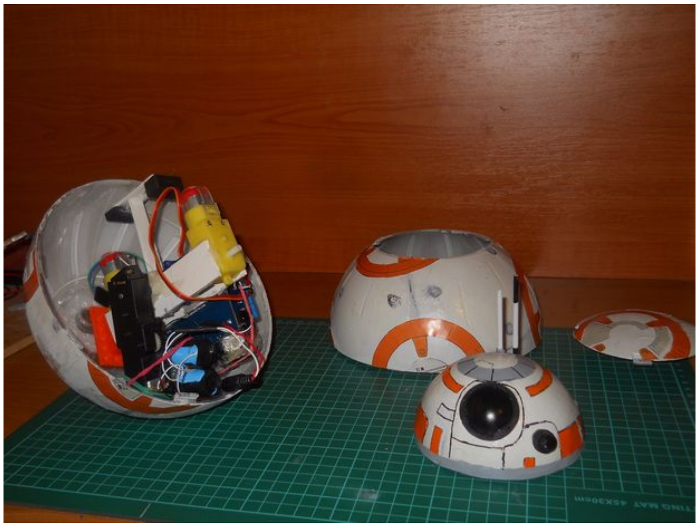
I'm a huge Star Wars fan and when I saw the Star Wars: The Force Awakens I thought that I need a BB-8 droid. It was awesome how this little sphere moved in the movie. So I decided that I'll make this droid that is based on the Arduino UNO (or other microcontroller). In this instructable I'm going to show how to build this droid with the LinkIt ONE that is a very popular board these days on the Instructables, so in a few steps I'll explain the code and if you want to use a LinkIt ONE you have to build a motor driver circuit that is also explained in my instructable. What if you haven't a LinkIt ONE? Don't worry **I used an Arduino UNO** clone to build my project, because the Arduino is much more popular than any other microcontroller, the LinkIt ONE is a short upgrade in four steps for LinkIt ONE users.

Some features:

- **phone controlled**
- **voice controlled**
- **Arduino based**
- **realistic head and body motion**
- **light effects**
- **sound effects**

This BB-8 model can be controlled with a smartphone using an application downloaded from the Play Store, iOS users have to use another app, because this one isn't compatible with iDevices, I'll insert links for different apps for the Apple users. Can be controlled even with voice commandsusing another Android application. BB-8 has light and sound effects just like in the movie. This droid is able to move his head, can turn left and right, moves forward or backward. The body moves realistic too, but there is difference between the original and my mechanics. Eventually is a good replica, it's only a little bit smaller than the original.

So my first intent was to make a realistic R/C replica of the BB-8. My second intent was to build this droid at very low cost. I didn't want to spend a lot of money for this project and I think succeed, **the full cost is less $40**. This is relatively cheap, if you check the prices of the BB-8 toys. If you want to build this droid please read every steps attentively. Every single detail can be useful and important.



Step 1: How does BB-8 works?

This website explains very good how the original BB-8 would work. The mechanism is complex and requires a lot of motors, gyroscope and a better circuit. The head can be kept and moved using magnets that are in the sphere.

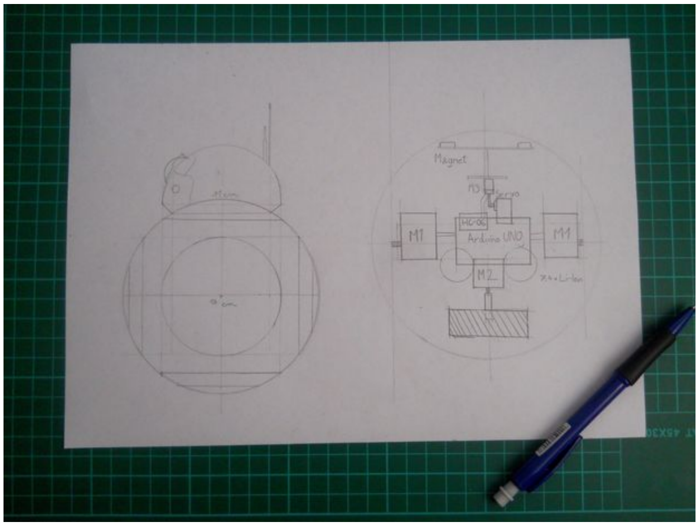
Step 2: Other Maker's Design

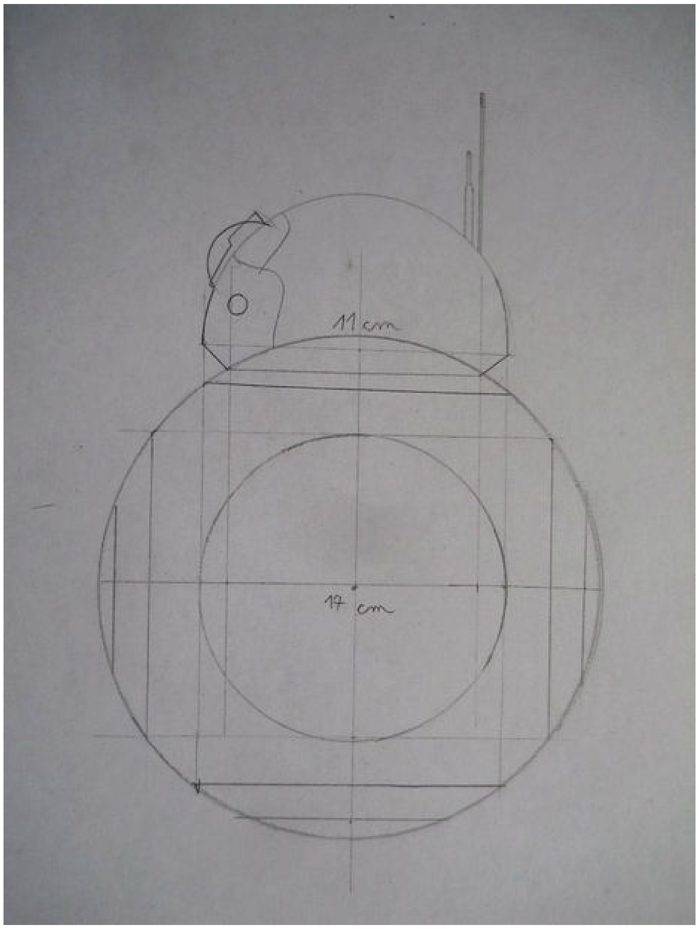
I think James Bruton's design is one of the best that I saw on the internet. It's simply awesome... But Mike Senna's droid is amazing too. I know that mine isn't so good like those, but I tried to make something new, invidual without copying the design of these two amazing makers.

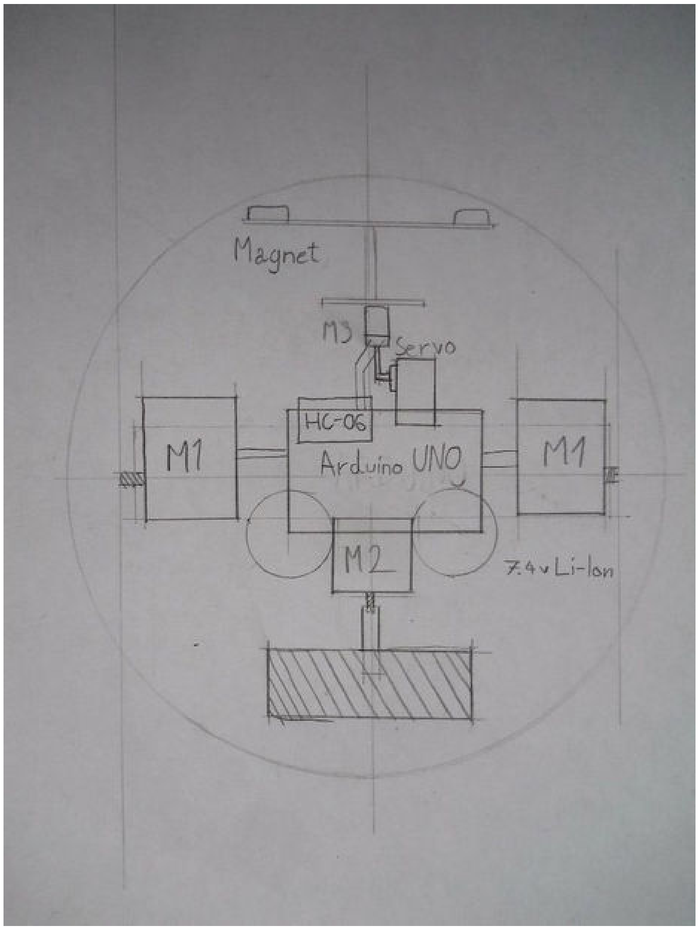
Step 3: My Desing and Concept

I made some drawings about my plan. I used a cardboard globe for head and a plastic hamster ball to make the body. In this hamster ball is placed the **Arduino** and the motors. The **M1** (motor) moves the droid forward and backward, the **M2'**s role is to turn the body left or right. To the M2 should be connected something metal or anything that has big weight, otherwise the droid won't be able to turn left or right. As a weight I simply used the two batteries, so don't need to add extra weight. Later I'll explain this featurette. The **Servo** moves the head on the body forward and backward and the **M3** turns the head left or right. The head can be attached to the droid using magnets.

The size of the bigger ball's diameter is 17 centimeters. The smaller half ball is a cardboard ball that's diameter is 11 centimeters.







Step 4: Choose your microcontroller!

As said I'll show you the instructions with both microcontrollers, but I'll build the droid with the Arduino UNO. So if you want to add special effects like sound use a LinkIt ONE, but I recommend to use a well-known Arduino UNO with an Adafruit motor shield. Both circuits has advantages, like:

LinkIt ONE:

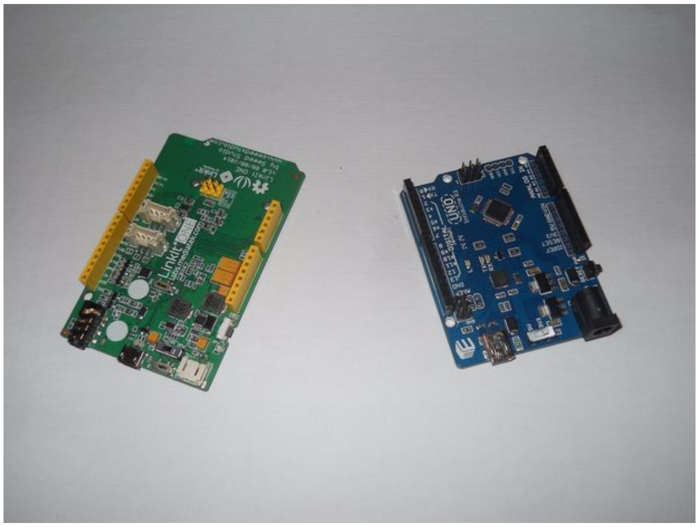
- **build-in Bluetooth**
- **sound effects**
- **light effects**

But is incompatible with Adafruit motor shield, you'll need to make a motor shield... And can't move the head.

Arduino UNO:

- **motor shield compatible**
- **light effects**
- **much more realistic motion**

If you use Arduino you'll need a HC-06 Serial Bluetooth module and you can't build in sound effects. In the next four steps I'll show how to use the LinkIt ONE with a DIY motor shield, so if you are an Arduino user skip these steps.



Step 5: LinkIt ONE: Parts

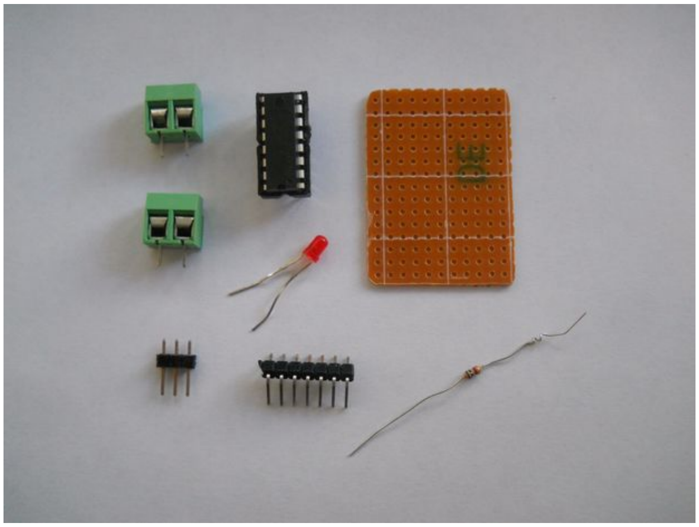
You'll need the following tools:

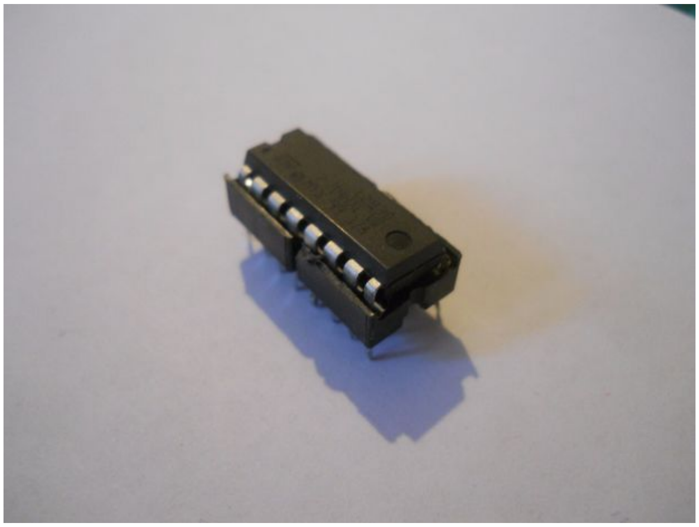
- soldering iron
- glue gun
- wire stripper/cutter
- PC with Arduino IDE and LinkIt ONE library

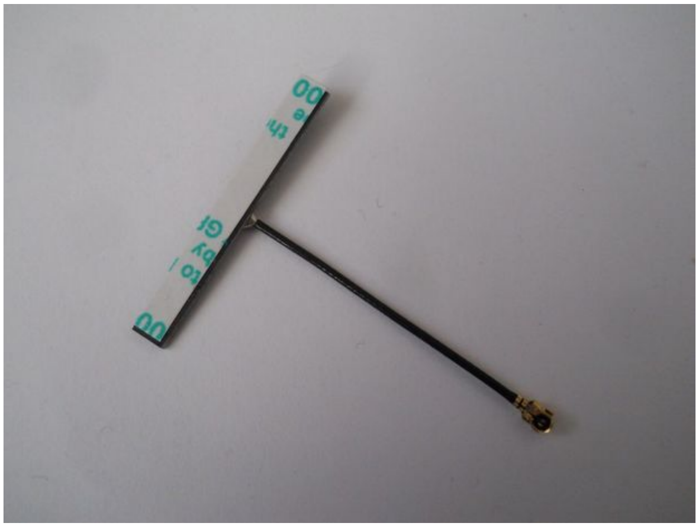
Parts:

- Mediatek LinkIt ONE Developer Board
- 3 DC Motors
- Servo Motor
- L293D Motor Driver
- 1k resistor
- Red/Green/Yellow LED
- Battery (Included with the LinkIt ONE kit)
- Bluetooth Antenna
- External Battery (2x Lithium-Ion Batteries)
- Screw Terminals

The other parts are the same as the Arduino UNO users: motors, sphere, etc...

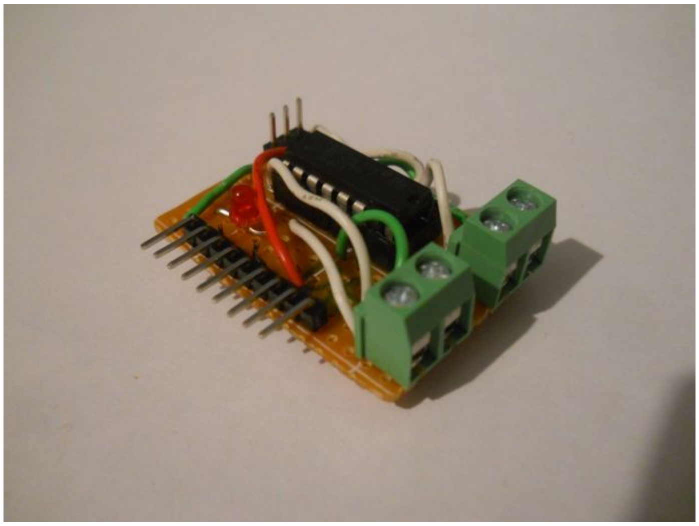


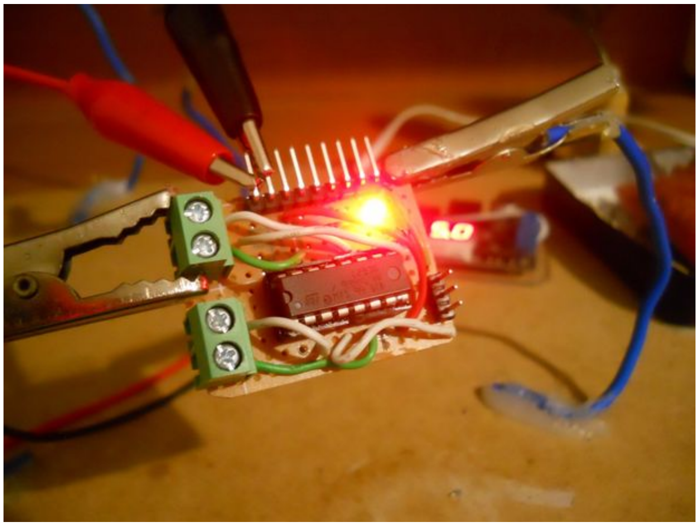


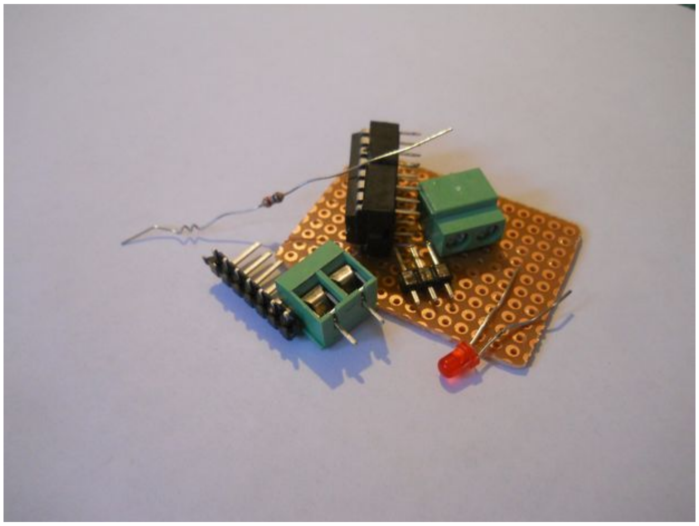


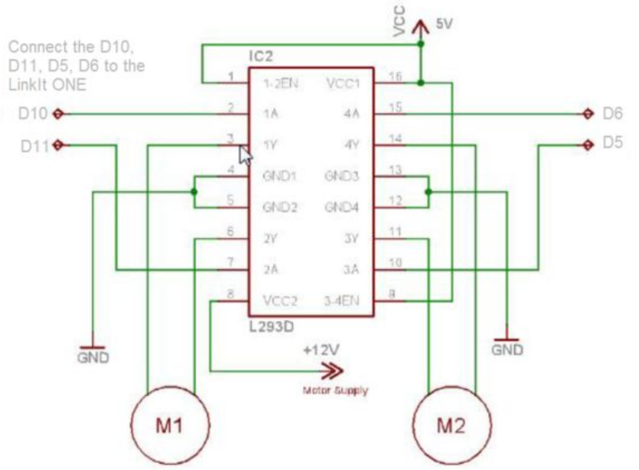
Step 6: LinkIt ONE: The $1 Motor Driver

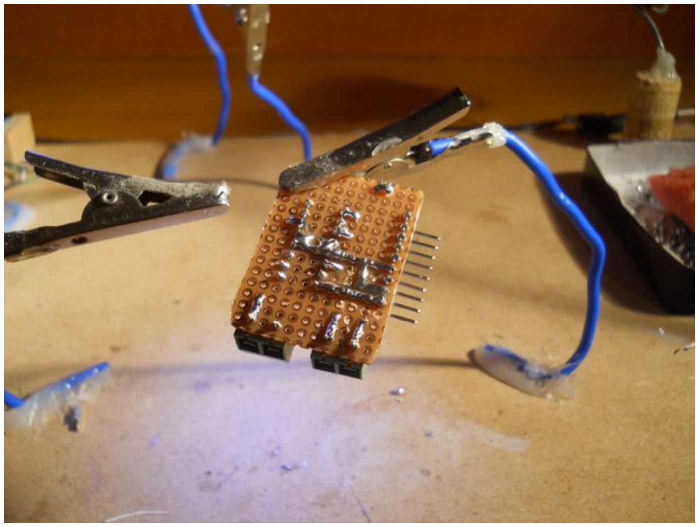
Now you have to make this very low cost motor shield/driver. With this circuit you can control your motors in two directions: forward and backward. I marked the connection point between the LinkIt and the circuit: D5, D6, D10 and D11, these digital output pins can control the code with my code, So if the LinkIt One example pulls up the D6 to HIGH the motor starts.

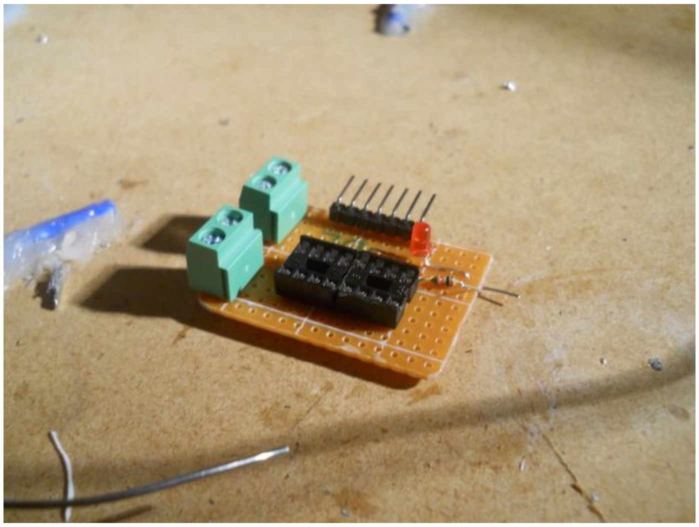


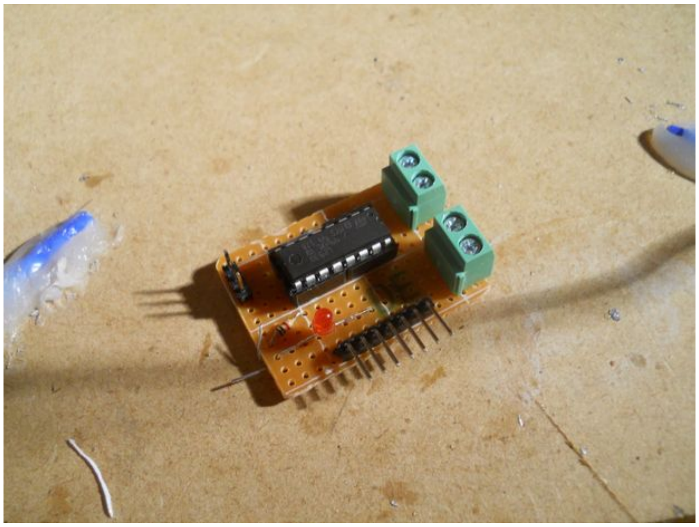


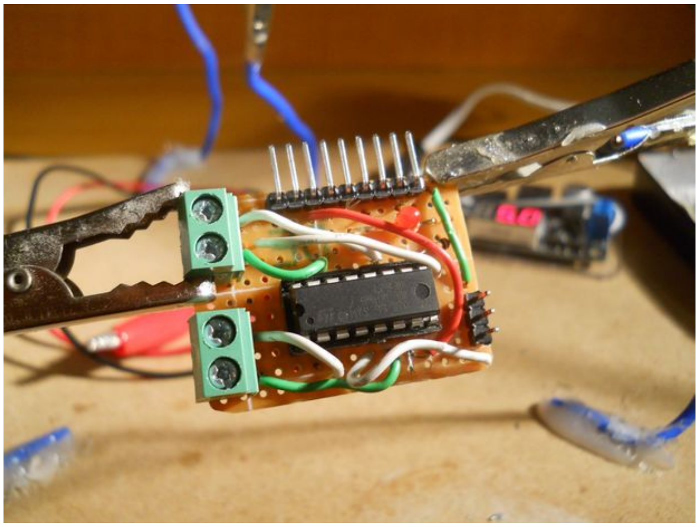


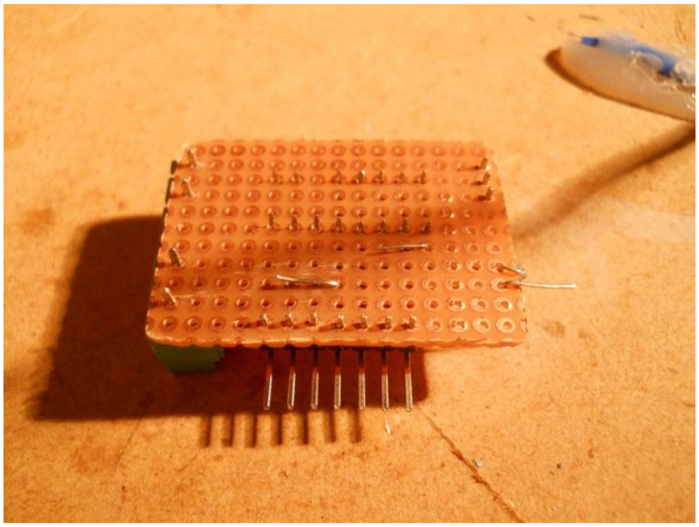


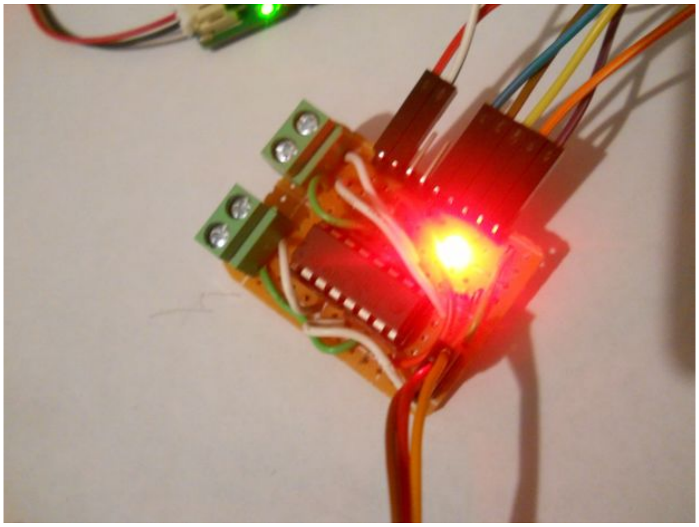






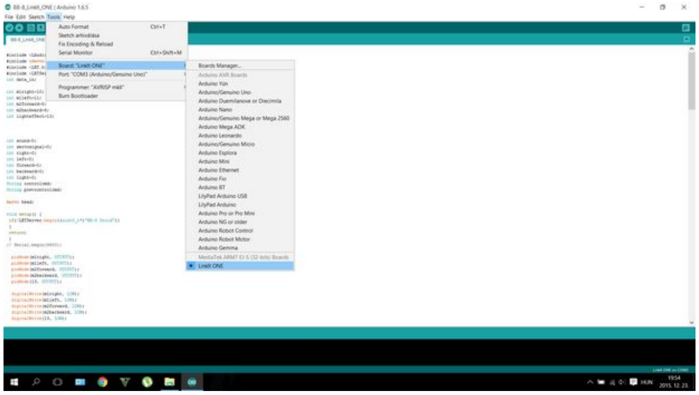




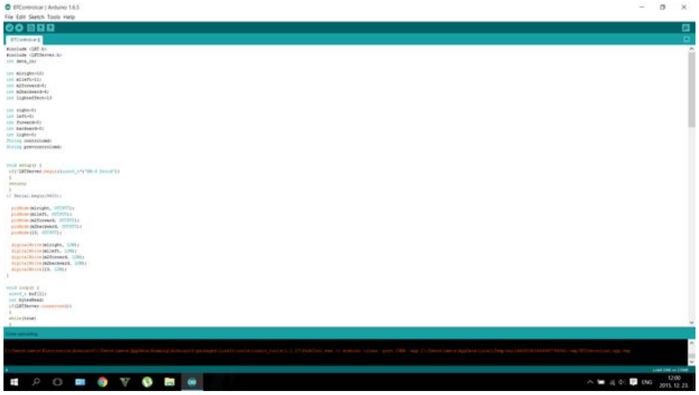


Step 7: LinkIt ONE: The Code

After building the circuit download this or write an own code to recieve the bytes from the application. This code creates a bluetooth server, reads incoming bytes then reacts to them by moving the servo and motors, turning on light or sound effects.



```
#include <LAudio.h>
#include <Servo.h>
#include <LBT.h>
#include <LBTServer.h>
int data_in;

int m1right=13;
int m1left=11;
int m2forward=6;
int m2backward=4;
int LightEffect=13;


int sound=0;
int servosignal=0;
int right=0;
int left=0;
int forward=0;
int backward=0;
int light=0;
String controlcmd;
String prevcontrolcmd;

Servo head;

void setup() {
 if(!LBTServer.begin((uint8_t*)"BB-8 Droid"))
 {
 return;
 }
// Serial.begin(9600);

  pinMode(m1right, OUTPUT);
  pinMode(m1left, OUTPUT);
  pinMode(m2forward, OUTPUT);
  pinMode(m2backward, OUTPUT);
  pinMode(13, OUTPUT);

  digitalWrite(m1right, LOW);
  digitalWrite(m1left, LOW);
  digitalWrite(m2forward, LOW);
  digitalWrite(m2backward, LOW);
  digitalWrite(13, LOW);
```



```
#include <LBT.h>
#include <LBTServer.h>
int data_in;

int m1right=13;
int m1left=11;
int m2forward=6;
int m2backward=4;
int lighteffect=13

int right=0;
int left=0;
int forward=0;
int backward=0;
int light=0;
String controlcmd;
String prevcontrolcmd;


void setup() {
 if(!LBTServer.begin((uint8_t*)"BB-8 Droid"))
 {
 return;
 }
// Serial.begin(9600);

  pinMode(m1right, OUTPUT);
  pinMode(m1left, OUTPUT);
  pinMode(m2forward, OUTPUT);
  pinMode(m2backward, OUTPUT);
  pinMode(13, OUTPUT);

  digitalWrite(m1right, LOW);
  digitalWrite(m1left, LOW);
  digitalWrite(m2forward, LOW);
  digitalWrite(m2backward, LOW);
  digitalWrite(13, LOW);
}

void loop() {
 uint8_t buf[1];
 int bytesRead;
 if(LBTServer.connected())
 {
 while(true)
 {
```

```
Done uploading.
```

File Downloads

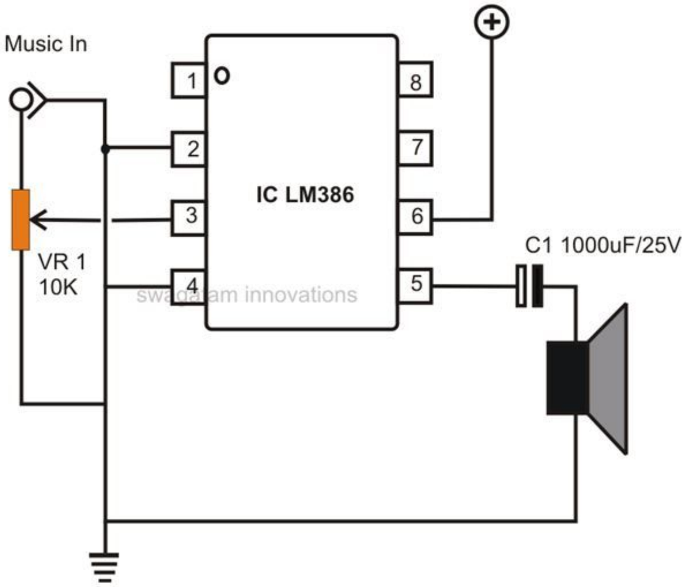
BB-8_LinkIt_ONE.ino (3 KB)

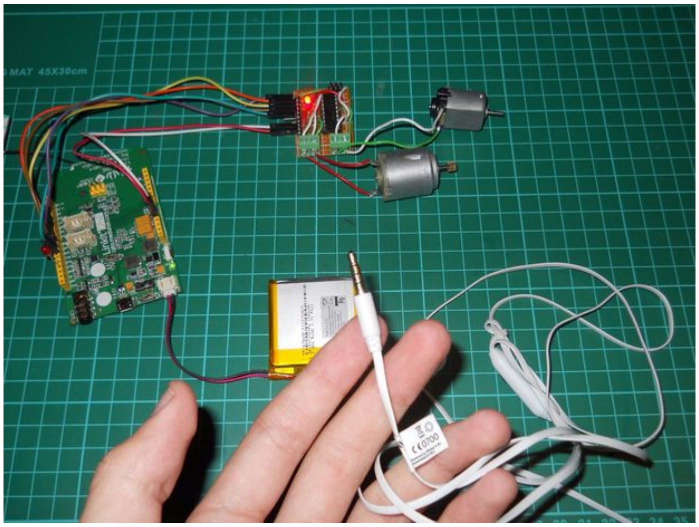
[NOTE: When saving, if you see .tmp as the file ext, rename it to 'BB-8_LinkIt_ONE.ino']

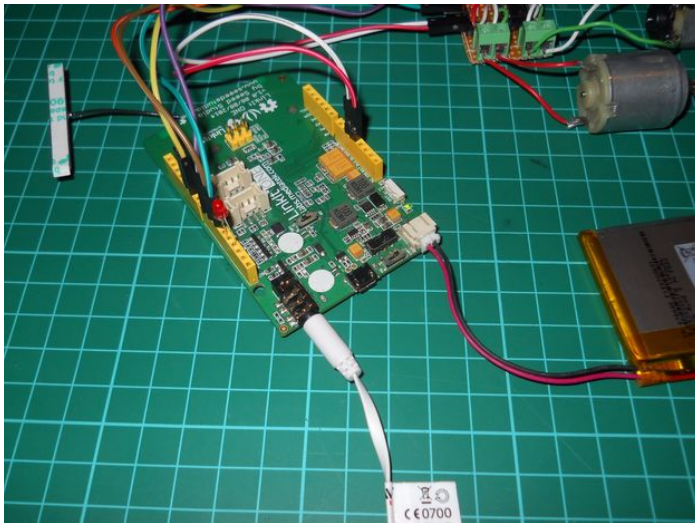
Step 8: LinkIt ONE: The Audio Amp for Sound Effect

Download BB8.mp3 file put on an SD card and plug in the LinkIt ONE.

If you want to add sound effect build this small circiut that amplifies the sound signal from the microcontroller. I couldn't build this circuit because I needed time for the Arduino Uno part, so I just made a quick test with my earphone, but if you have questions feel free to ask.





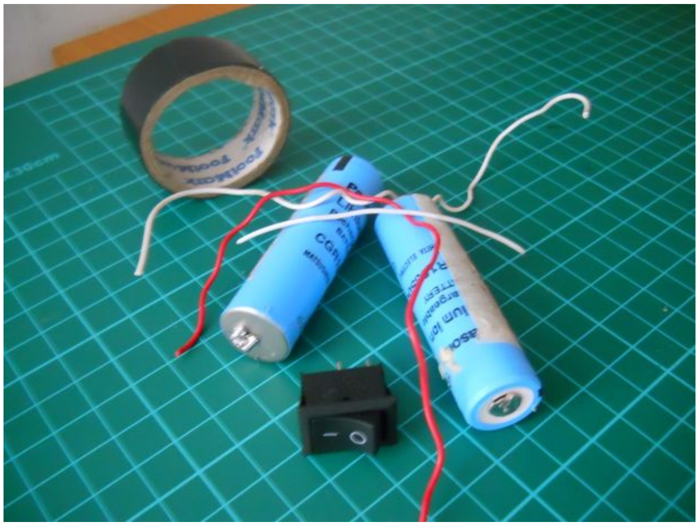


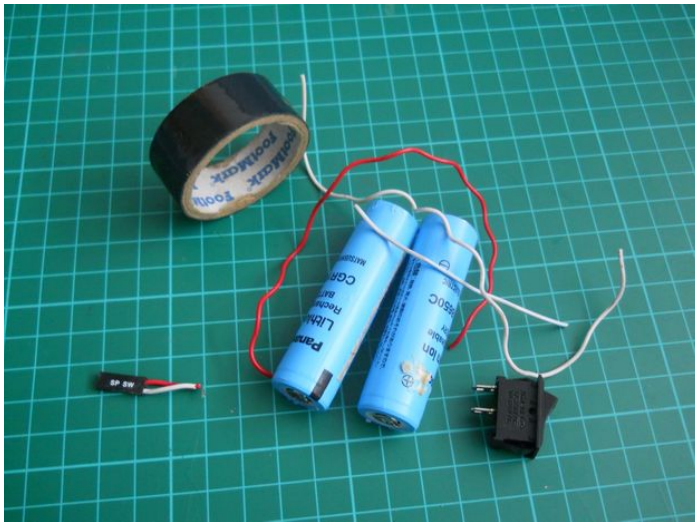
File Downloads

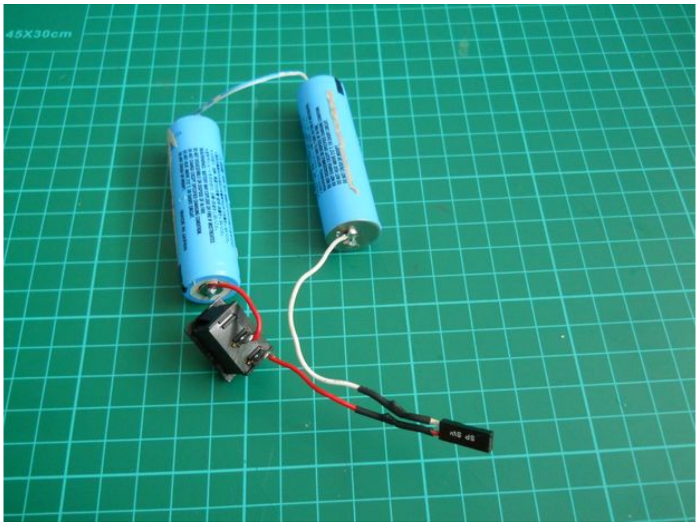
BB8.mp3 (216 KB)
[NOTE: When saving, if you see .tmp as the file ext, rename it to 'BB8.mp3']

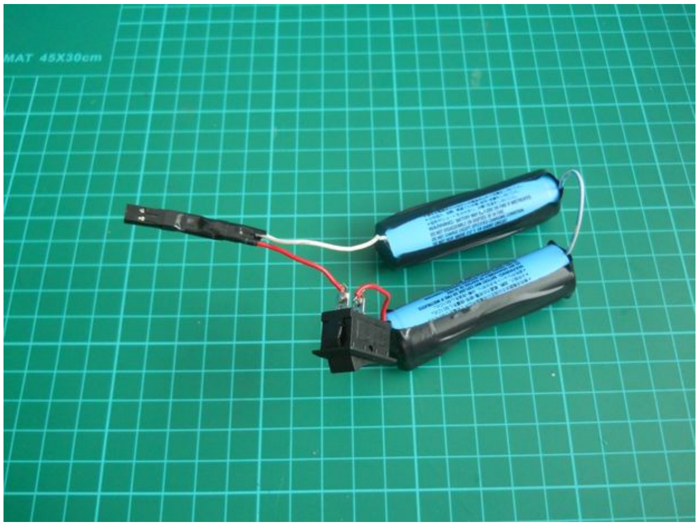
Step 9: LinkIt ONE External Battery

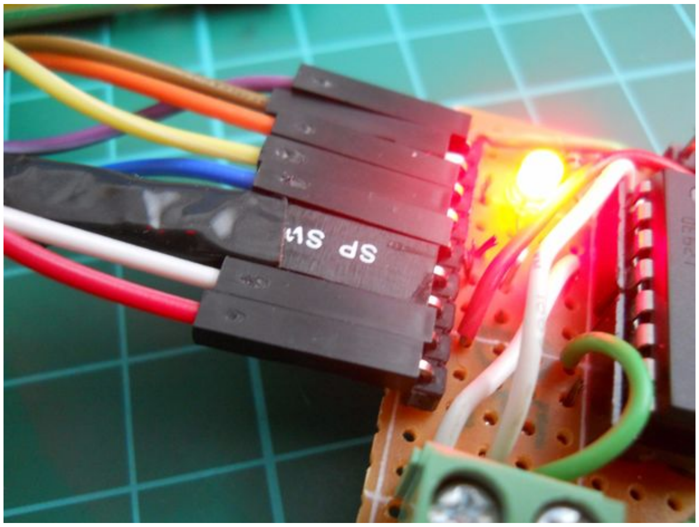
You have to make a 7.4v battery for the motors. My solution was to connect two 18650 batteries in series.





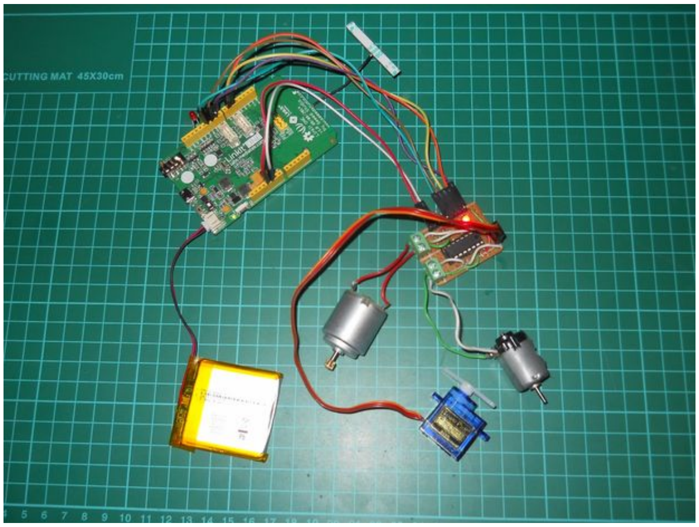


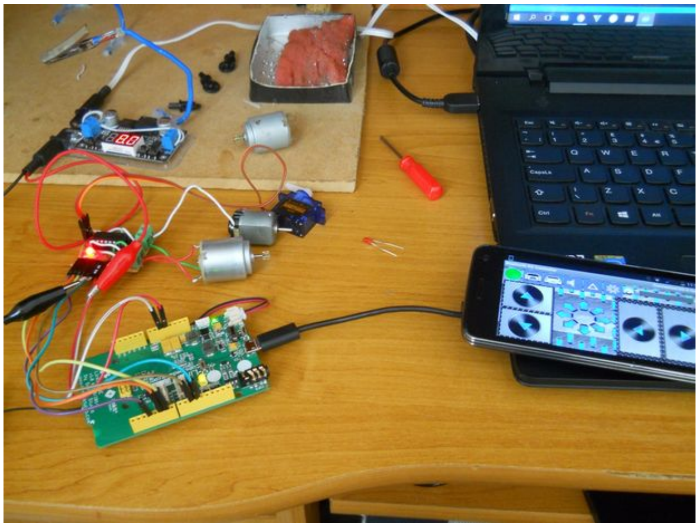


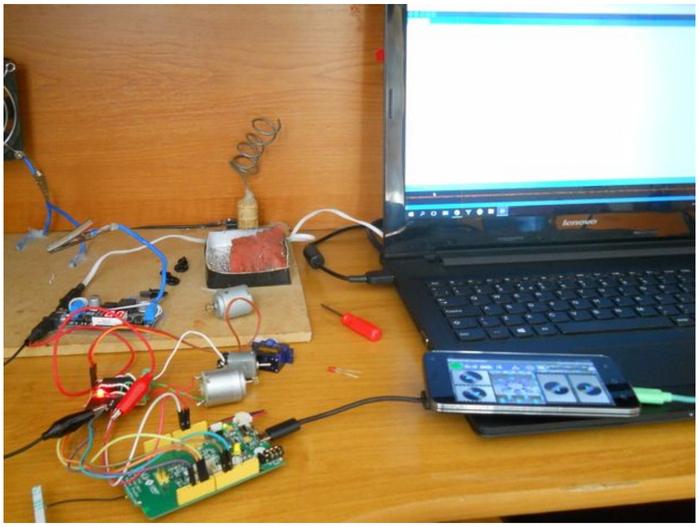


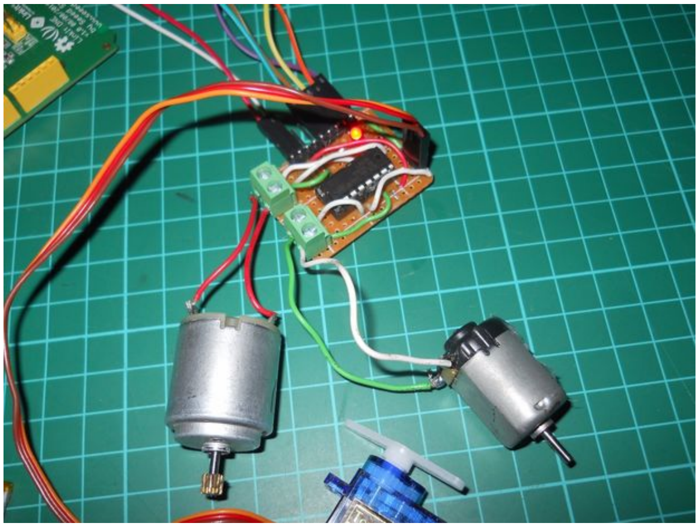
Step 10: LinkIt ONE: The test

I used some motors to test the circuit and the coed and they worked fine. At first launch the servo had some issue, but after this worked without problems. The sound through the earphones was very good. Look at the video if you want to see how it works. With this step I finished the explanation and instructions for the LinkIt ONE. Go to the mechanism part to see how to build the droid.









Step 11: Arduino UNO: Parts

You'll need the following tools:

- soldering iron
- glue gun
- wire stripper/cutter
- PC with Arduino IDE and Adafruit Motor Shield Library

Parts:

Electronics:

- Arduino Uno (buy from Amazon or eBay)
- HC-06 Serial Bluetooth Module
- Motor Shield
- 18650 Lithium-Ion Battery (I salvaged mine from an old laptop, if you buy online beware fake Ultrafire batteries)
- DC Motors with Gears (3 Pieces)
- Servo Motor
- Blue LED
- Jumper Wires
- 5.5mm Power Jack

Mechanism:

- Some Lego Parts
- A Hamster Ball
- A Plastic Globe (for head)
- Srews and Zip Ties
- Plastic or Plexy Glass sheets
- Magnets

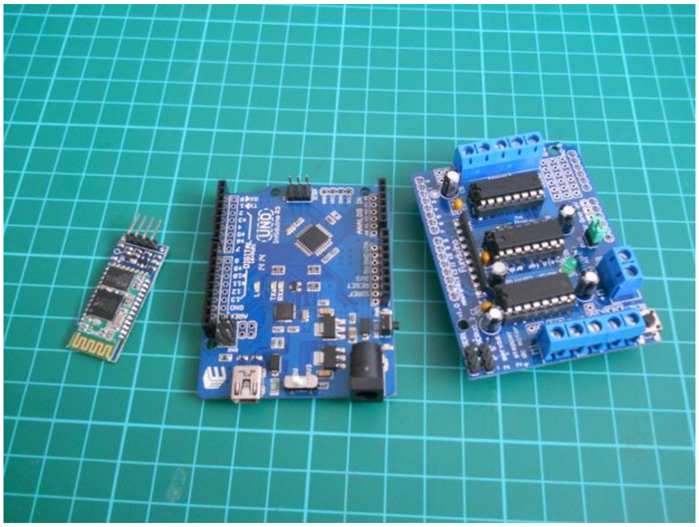
Appearence and Paint:

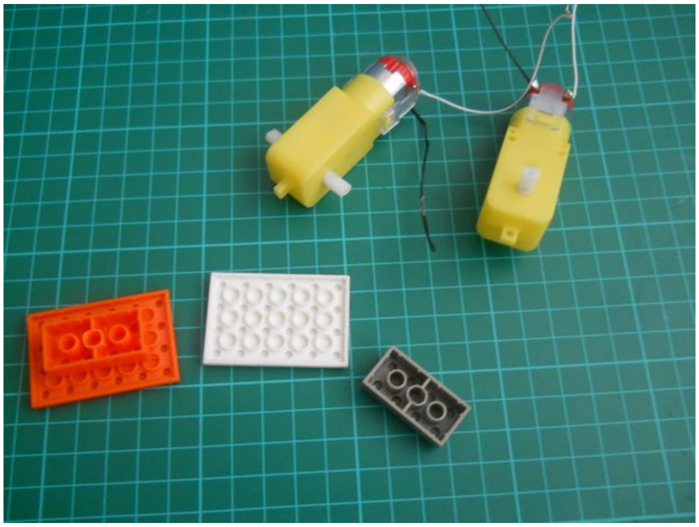
- White Paint Spray
- Black Paint
- Blue and Orange Paint
- Ping-Pong Ball (for eye)
- Black Marker

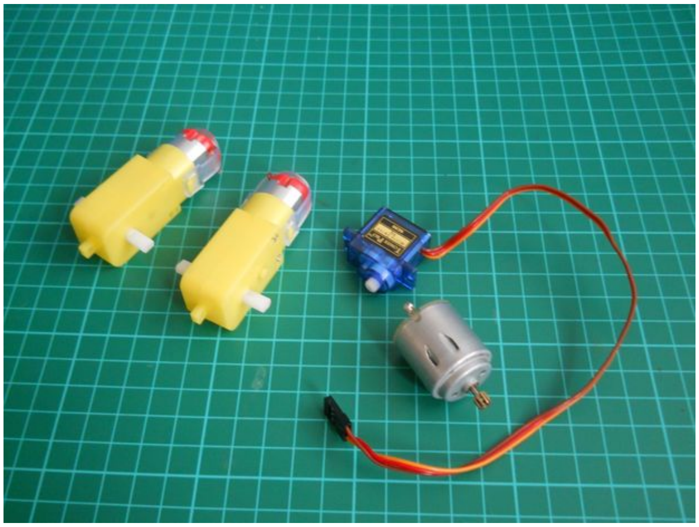
Software:

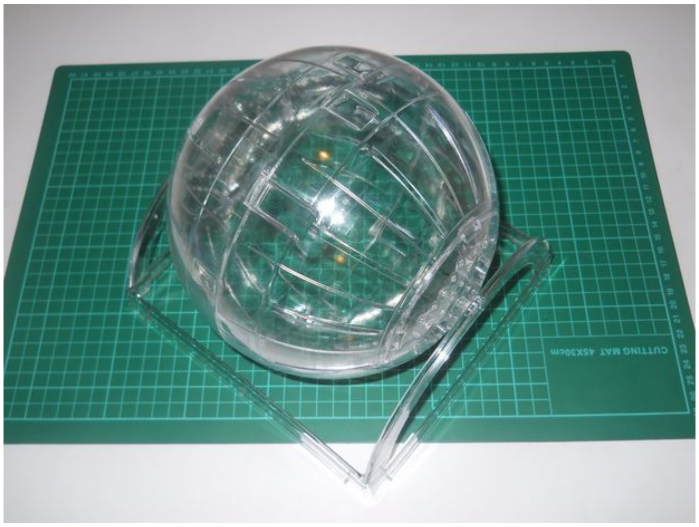
- Ardiono IDE
- Adafruit Motor Shield Library
- BT RC Car Android App

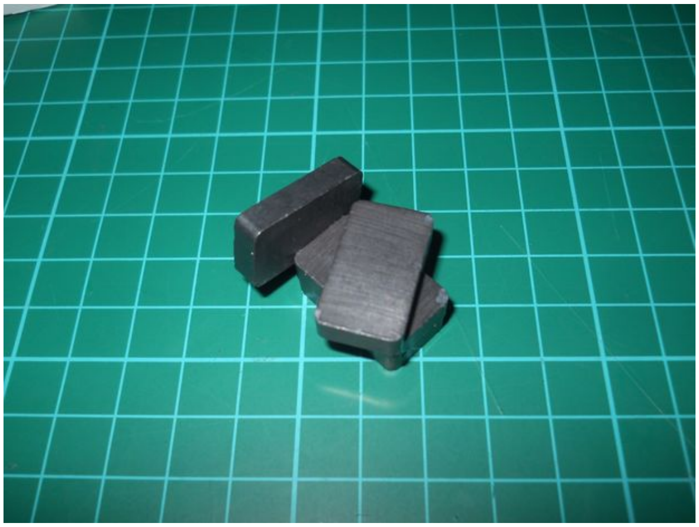
For paint you can use anything or any technique, example: print the circles of BB-8 then glue on the spehe.











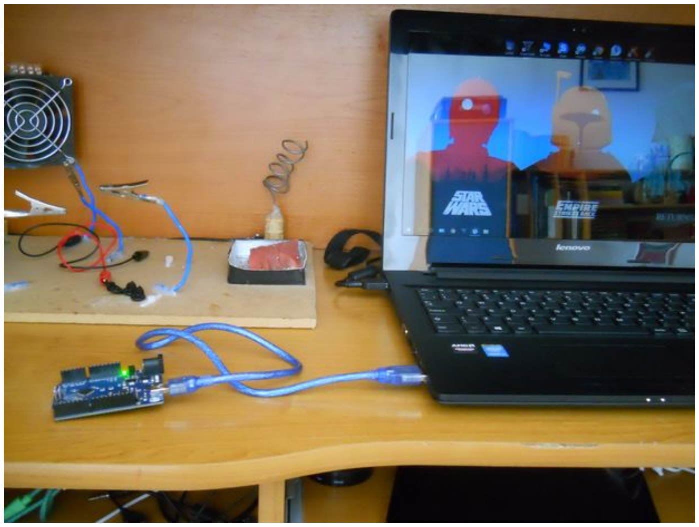
Step 12: The Ardiuno Code

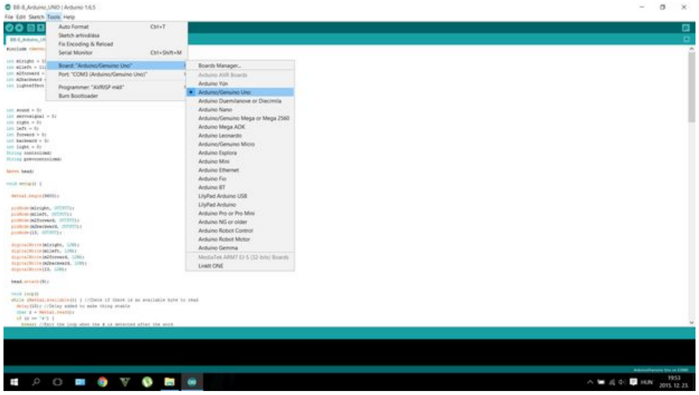
This part is easy for every Arduino users. Just download or copy-paste this code in your IDE then upload to your Arduino. Of course you can use Mega, Leaonardo, Duemilanove and etc... The concept is easy. The Arduino uses HC-06 serial module to read the data coming from the phone. Then reacts to the incoming bytes by moving motors or turning on/off lights. In the code I inserted some explanations, but if you have questions feel free to ask.

You can test the code and the Arduino if you open the Serial monitor and send a letter to the microcontroller. I sent "W" and the light turned on. When I sent "w" the light turned off. So the code worked.

```
<p>/* The code was made by Tamás Imets for an Arduino UNO project<br>   This code controls the BB-8 Droid from the Star Wars: The Force Awakens,
  and works with Bluetooth RC Car application, using a HC-06 serial bluetooth module
  See more details at instructables.com/member/Imetomi
  Friday,  December 25, 2015
*/</p><p>#include <servo.h>
#include <softwareserial.h>
#include <afmotor.h>
int command = 0;  //incoming serial data</afmotor.h></softwareserial.h></servo.h></p><p>AF_DCMotor motor1(1, MOTOR12_64KHZ); //initialize motor1 that
AF_DCMotor motor2(2, MOTOR12_64KHZ); //motor2 is connected to M2 and this turns BB-8 left and right
AF_DCMotor motorhead(3); //this motor turns the head left or right</p><p>Servo head;
void setup() {
 Serial.begin(9600); // set up Serial library at 9600 bps - this is the speed the serial interface will work all
 head.attach(9); //attach servo to pin D9 or to servo2 on motor shield
 motor1.setSpeed(255); //set default speed
 motor2.setSpeed(255); //set default speed
 motorhead.setSpeed(1); //the motor should spin very slowly
 pinMode(13, OUTPUT); //connect here the blue LED
}</p><p>void loop() {
 if (Serial.available() > 0) { //if the Arduino detects incoming data
   // read the incoming byte:
   command = Serial.read();
 }
 switch (command) //set different cases of the "command" variable
 {
   case 'S': {
       motor1.run(RELEASE); // stopped
       motor2.run(RELEASE); // stopped
       command = '*';
     } break;</p><p>    case 'F': { //go forward
       motor2.run(RELEASE);
       motor1.run(FORWARD);
     } break;</p><p>    case 'B': { //go backward
       motor2.run(RELEASE);
       motor1.run(BACKWARD);
     } break;</p><p>    case 'R': { //spin right
       motor1.run(RELEASE);
       motor2.run(FORWARD);
     } break;</p><p>    case 'L': { //spin left
       motor1.run(RELEASE);
       motor2.run(BACKWARD);
     } break;</p><p>    case 'G': { //forward left
       motor1.run(FORWARD);
       motor2.run(BACKWARD);
     } break;</p><p>    case 'I': { //forward right
       motor1.run(FORWARD);
       motor2.run(FORWARD);
     } break;</p><p>    case 'H': { //backward left
       motor1.run(BACKWARD);
       motor2.run(BACKWARD);
     } break;</p><p>    case 'J': { //backward right
       motor1.run(BACKWARD);
       motor2.run(FORWARD);
     } break;</p><p>    case 'W': {
       digitalWrite(13, HIGH); //lights on
     } break;</p><p>    case 'w': {
       digitalWrite(13, LOW); //lights off
     } break;</p><p>    case 'U': {
       motor1.setSpeed(180); //set lower speed
     } break;</p><p>    case 'u': {
       motor1.setSpeed(255); //set max speed
```

```
    } break;</p><p>    case 'V': {
      motorhead.run(FORWARD); //turn head left
    } break;</p><p>    case 'v': {
      motorhead.run(RELEASE); //stop spinning
    } break;</p><p>    case 'X': {
      motorhead.run(BACKWARD); //turn head right
    } break;</p><p>    case 'x': {
      motorhead.run(RELEASE); //stop spinning
    } break;</p><p>    case '0': {
      head.write(95); //basic servo state
    } break;</p><p>    case '1': {
      head.write(50); //move servo
    } break;</p><p>    case '2': {
      head.write(60); //move servo
    } break;</p><p>    case '3': {
      head.write(70); //move servo
    } break;</p><p>    case '4': {
      head.write(80); //move servo
    } break;</p><p>    case '5': {
      head.write(90); //move servo
    } break;</p><p>    case '6': {
      head.write(95); //move servo
    } break;</p><p>    case '7': {
      head.write(110); //move servo
    } break;</p><p>    case '8': {
      head.write(120); //move servo
    } break;</p><p>    case '9': {
      head.write(130); //move servo
    } break;</p><p>    case 'q': {
      head.write(135); //move servo
    } break;
    //you can add more extra feautres by writing a case command (example: different lights, sound, etc...)
  }
}</p>
```





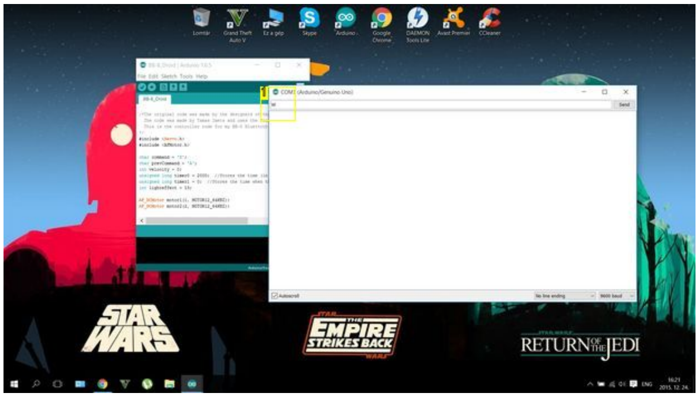


Image Notes
1. Testing

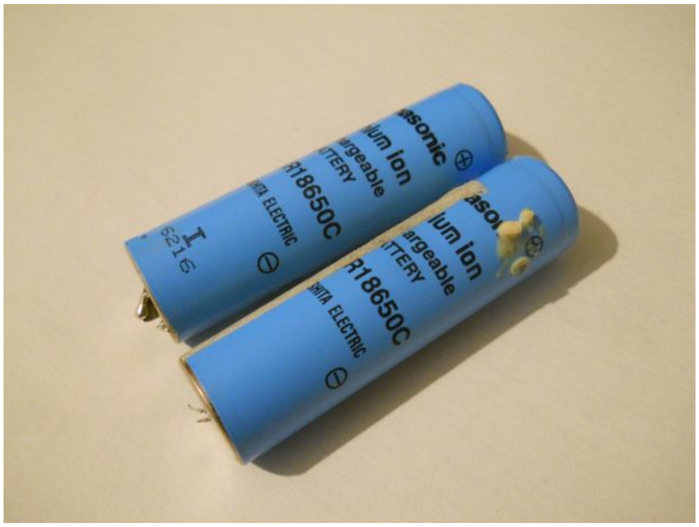
File Downloads

BB-8_Arduino_UNO.ino (3 KB)
[NOTE: When saving, if you see .tmp as the file ext, rename it to 'BB-8_Arduino_UNO.ino']

Step 13: Choosing the Power Source

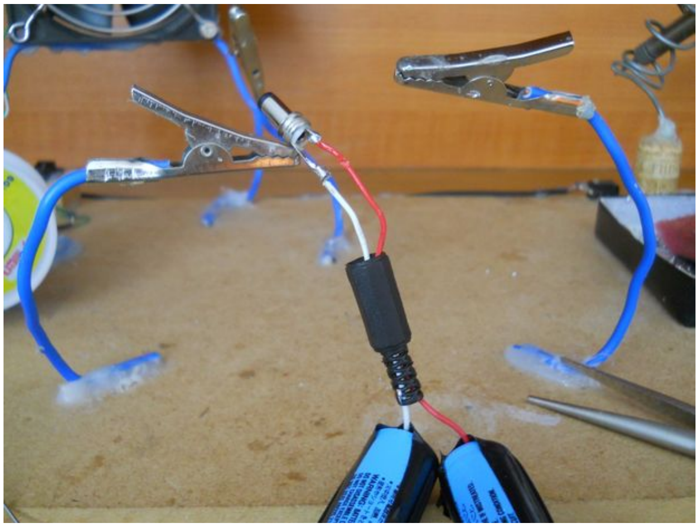
If you have, you can use old cell phone batteries for the power source but I reccomned 18650 cells. I chosed those Panasonic batteries, that were salvaged from an old laptop

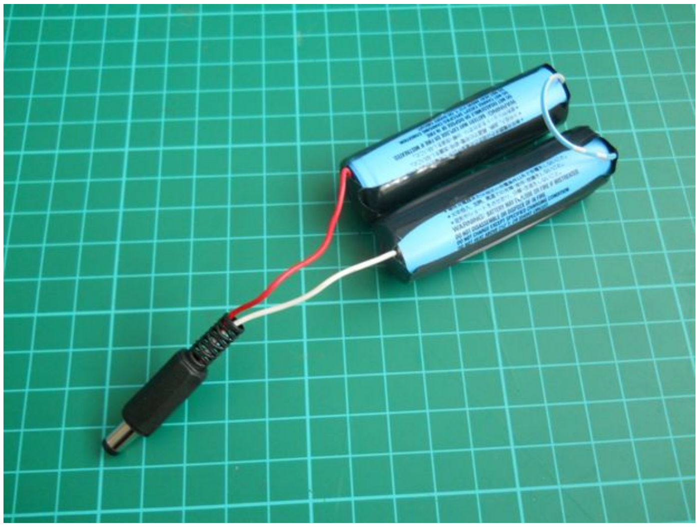
nasonic
thium Ion
hargeable
ATTERY
R18650C
SHITA ELECTRIC
6216
NOKIA
BL-5C
3.7 V
RECHARGEABLE Li-ion BATTERY.
USE SPECIFIED NOKIA CHARGER ONLY.
MUST BE DISPOSED OF PROPERLY.
MAY EXPLODE IF DAMAGED OR DISPOSED OF IN FIRE.
DO NOT SHORT-CIRCUIT.
Li-Ion
MADE IN CHINA



Step 14: Build the Power Source

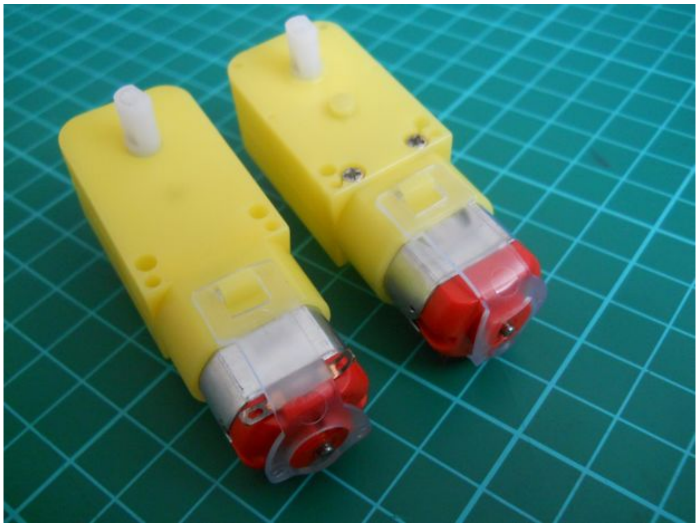
The recipe is simple: connect the batteries in series the solder the wires to the 5.5mm power jack. Now you have a 7.4v rechargeable power supply. You can charge up with 8.4 voltpower supply.

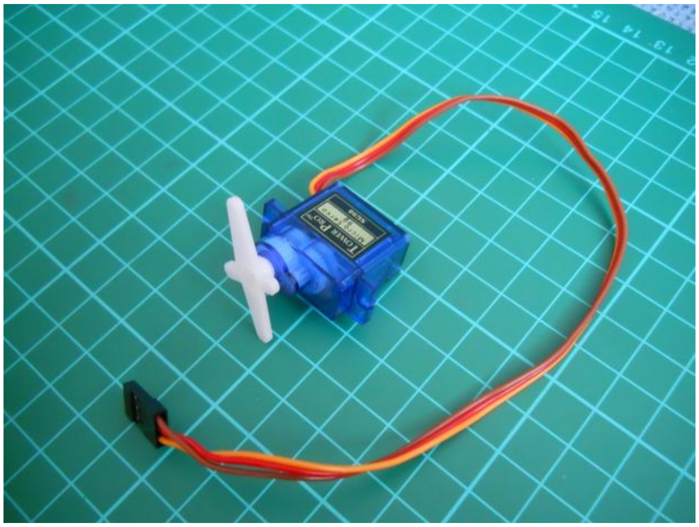




Step 15: Choosing Motors

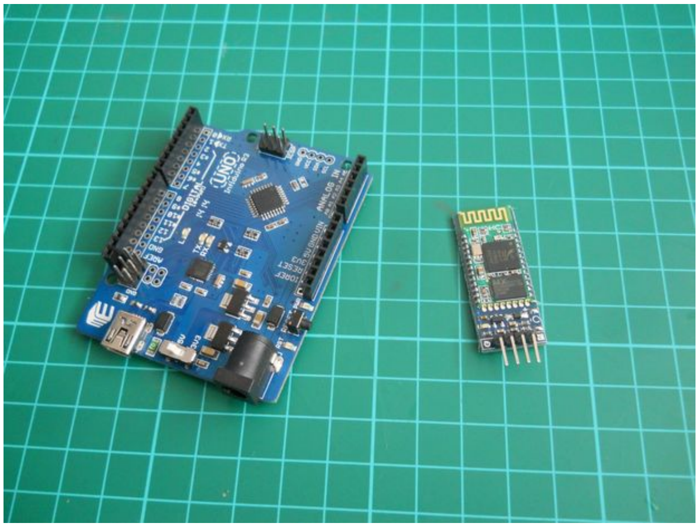
It was very hard to choose the right motors for this project. I needed motors with gearing that are enough powerful to move that very heavy sphere. Finally I bought 2 geared motors from the Amazon and they were very efficient. You should buy three of them to solve the steering problem.

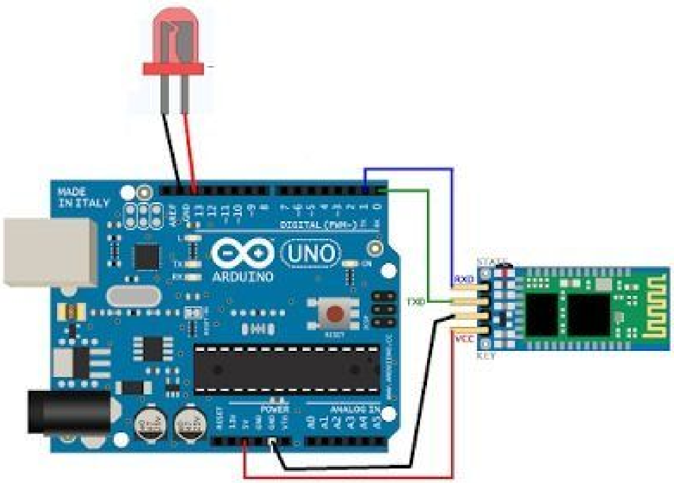


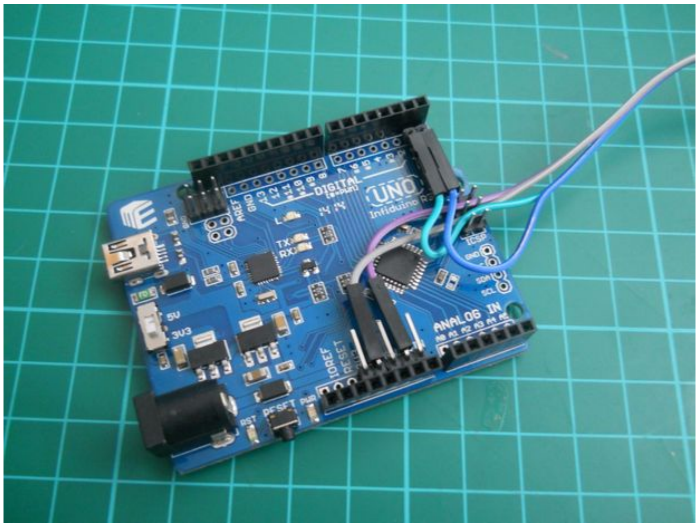


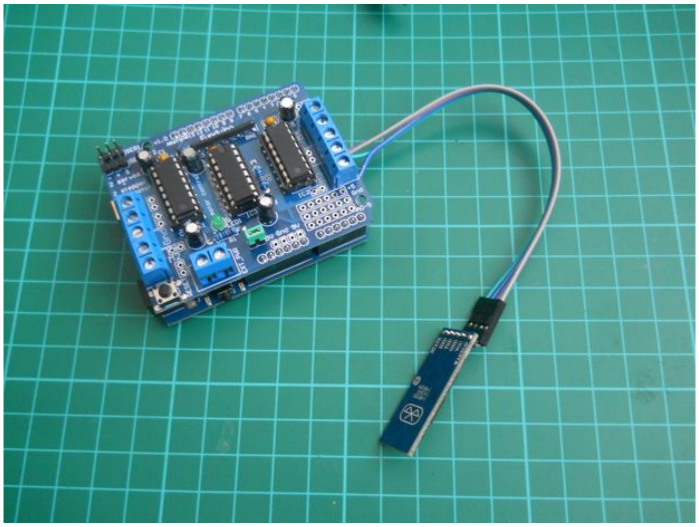
Step 16: The Circuit (Hardwere Part)

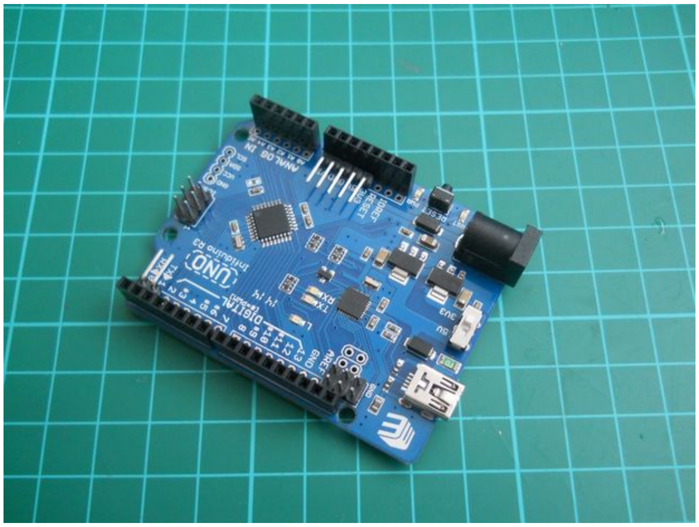
I soldered some extra legs for the Arduino I/O pins: D13, 5v, 3.3v, GND, RX, TX. Using these extra connection legs I was able to connect easily the bluetooth module and the LED. If you have a different Arduino you have to look for another solution. Connect everything to the Arduino just like on the schematic.

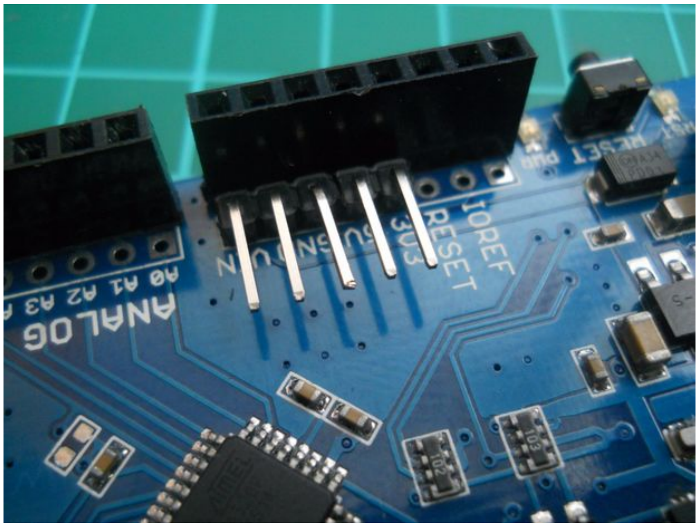


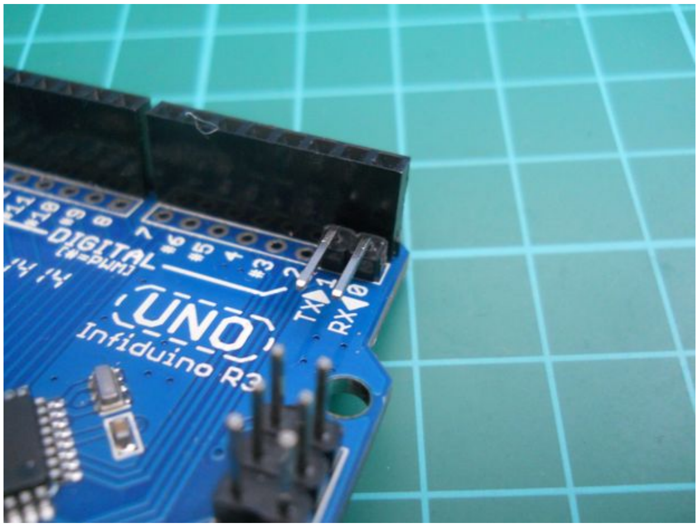








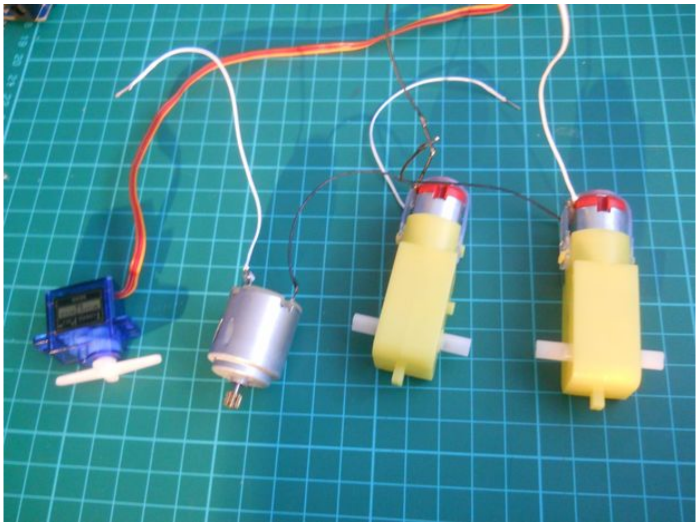


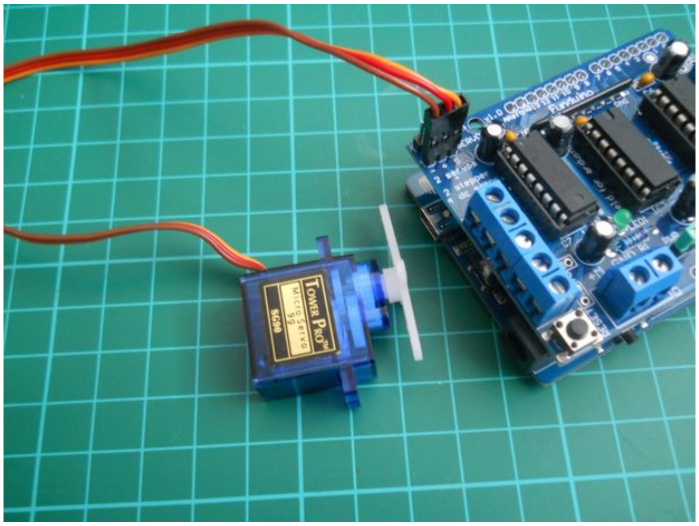


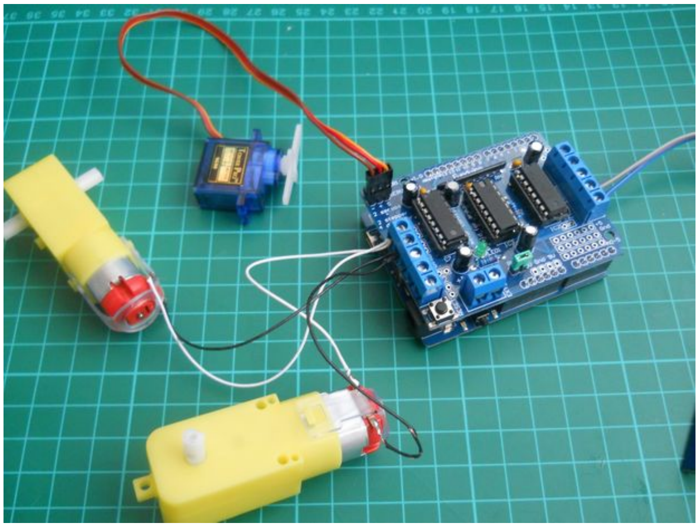
Step 17: The Motor Shield

Important note: Later I changed the grey motor. Use a third geared motor. "Difficult to see, always in motion is the future"-Master Yoda

Back to the project: Connect the servo motor to SERVO_2, two motors to the M1 and one motor to the M2. The to yellow motors are connected to the M1. These motors move the droid forward and backward.







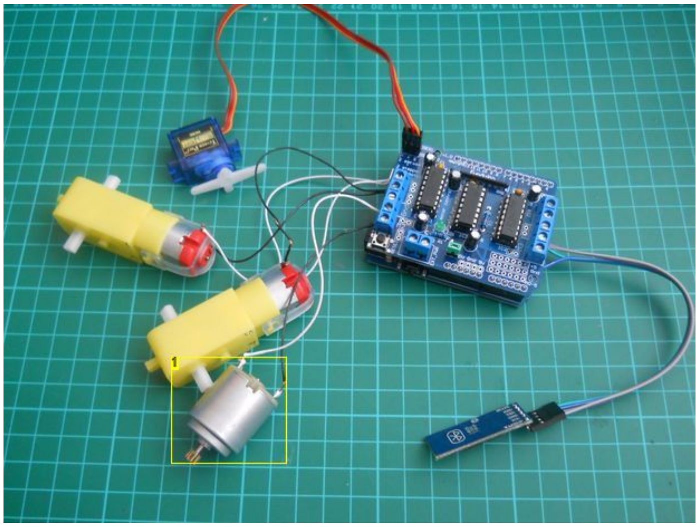
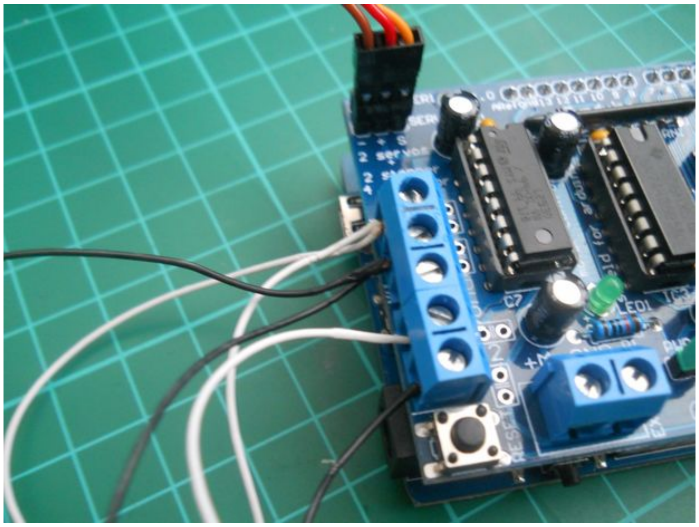
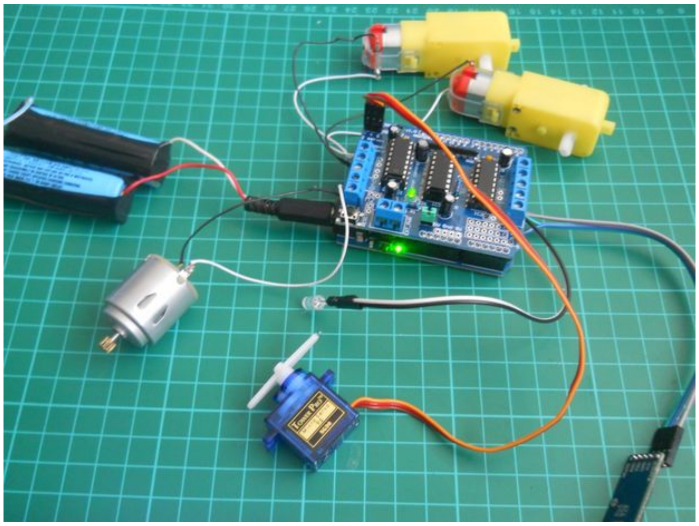


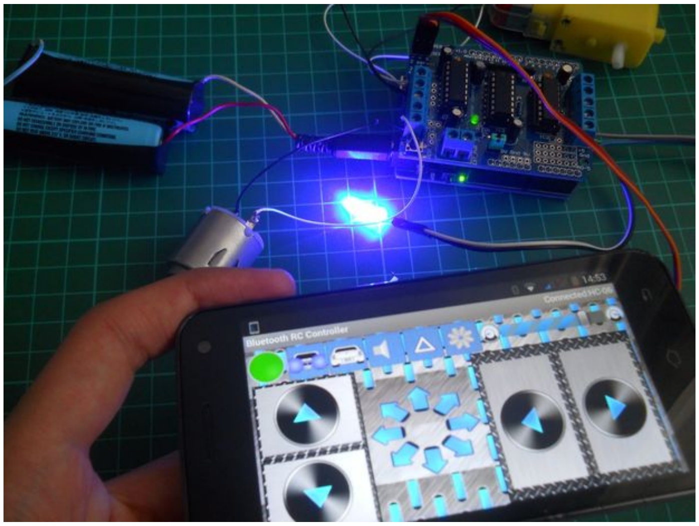
Image Notes
1. This motor was changed to another.

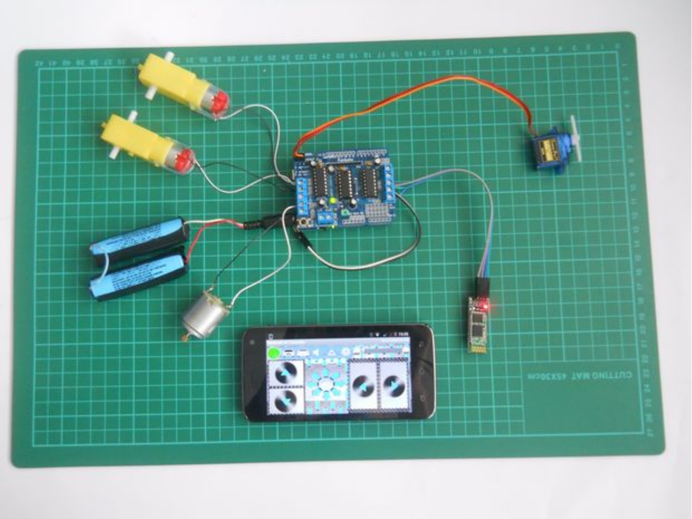


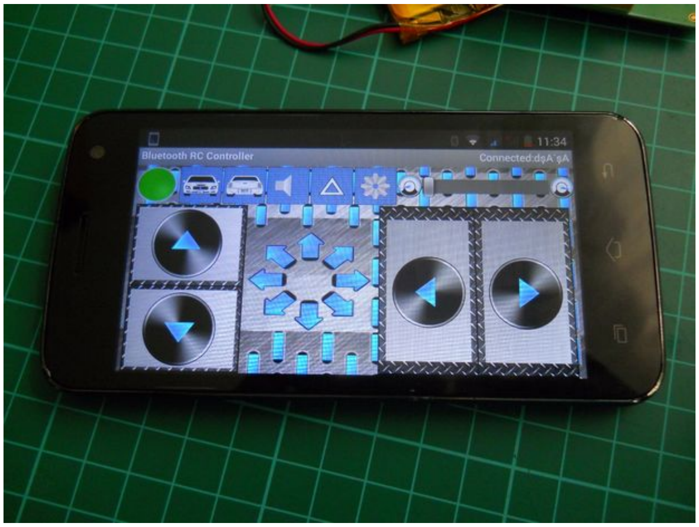
Step 18: A Quick Test

Use your smartphone with the downloaded app then connect to the microcontroller. Press any button that controls a motor. If worked go and build the sphere.



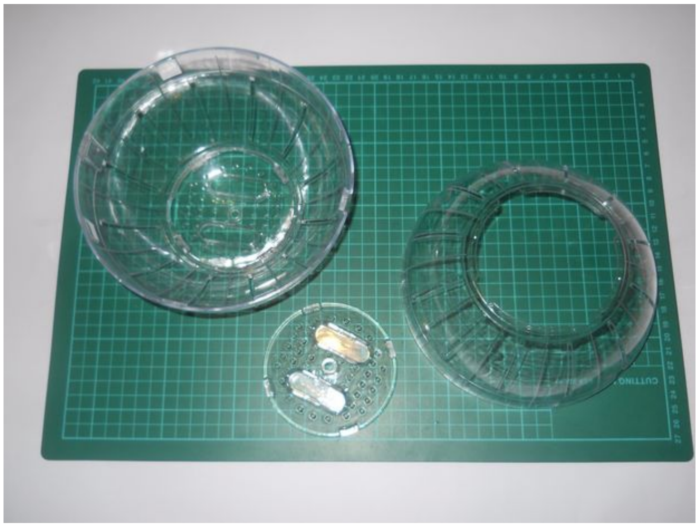


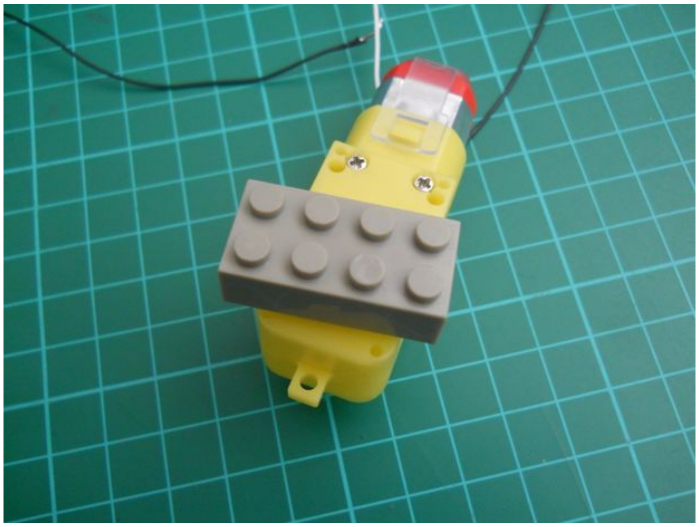


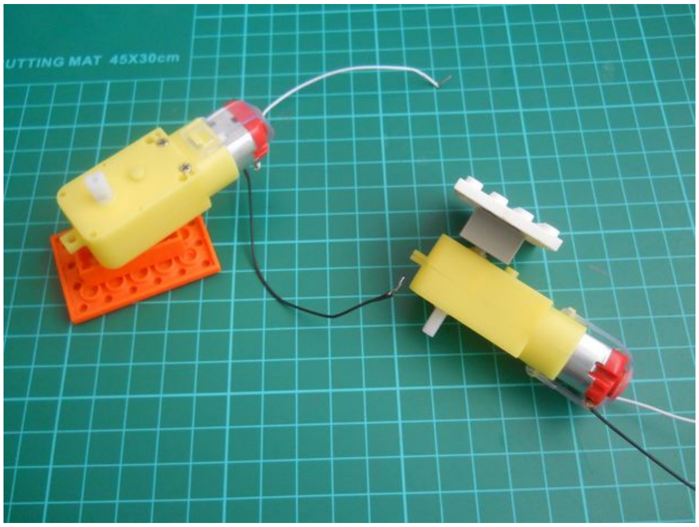


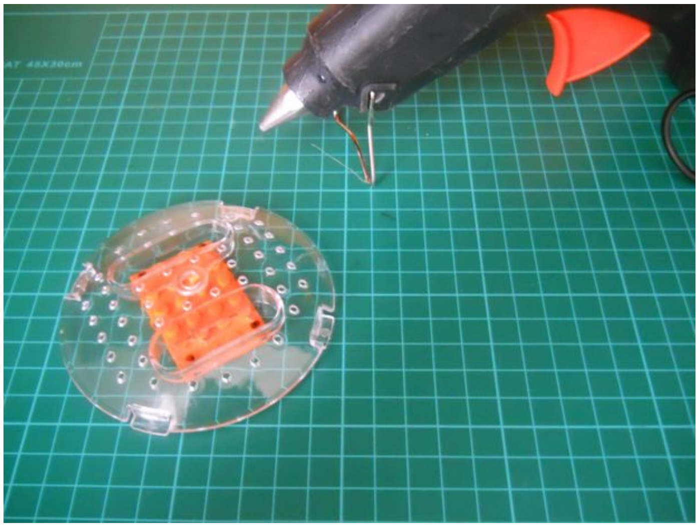
Step 19: The Mechanism

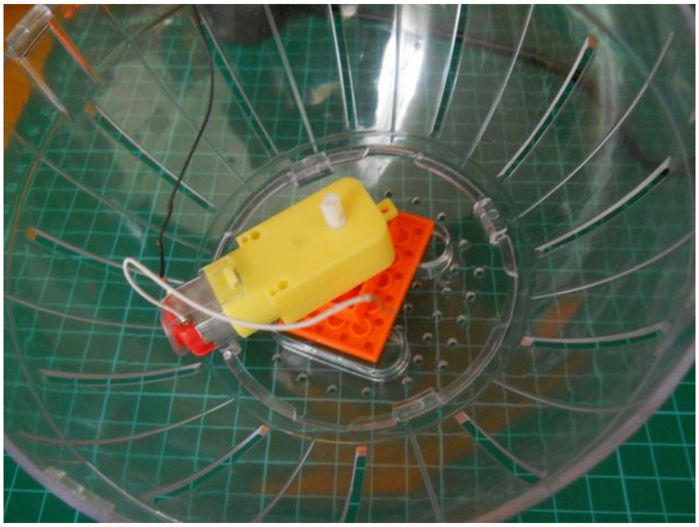
I took apart the hamster ball, then using the Lego parts I suited the motor to the ball. Why use Lego parts? Well it's very eaasy to take apart the droid is is needed. With a glue gun Iconnected the larger Lego part to the sphere and smaller is connected directly to the motor without glue.

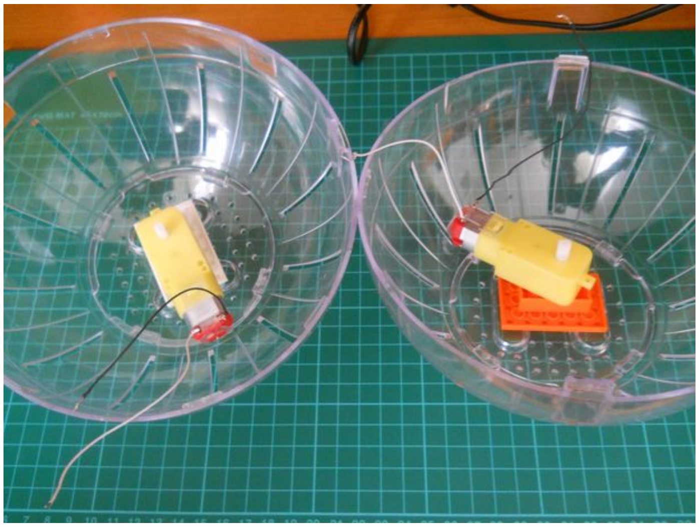






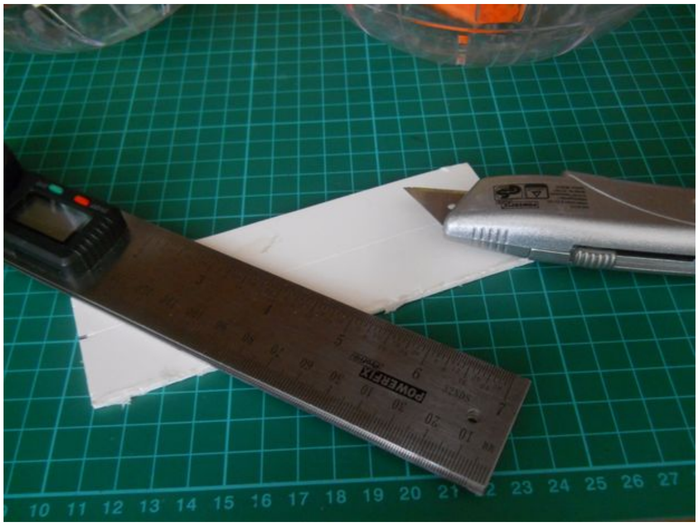


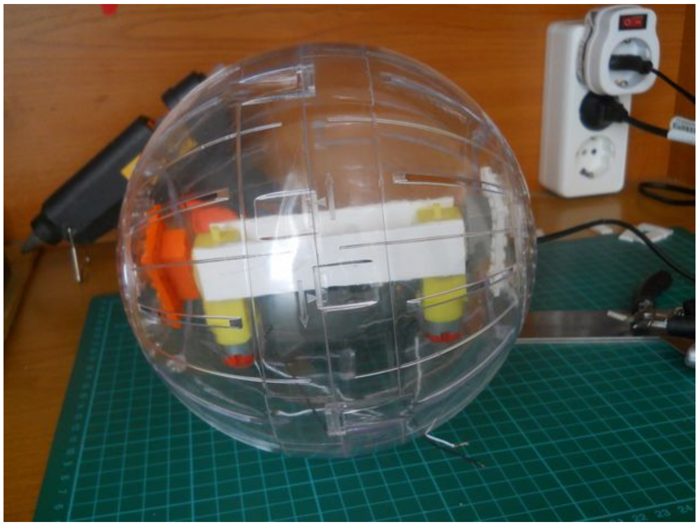


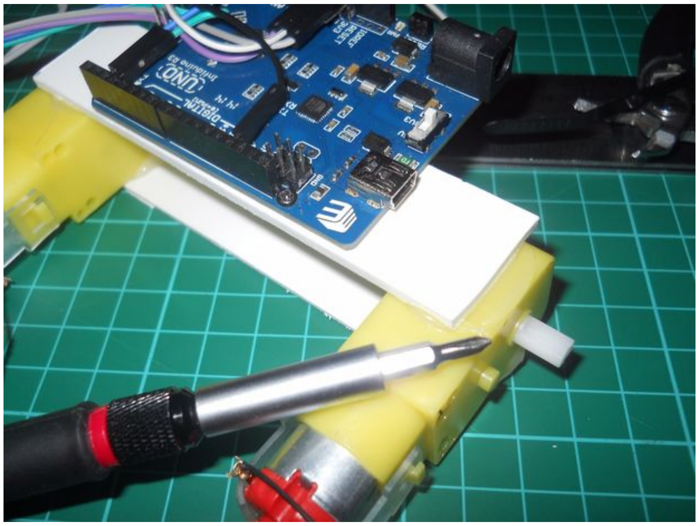


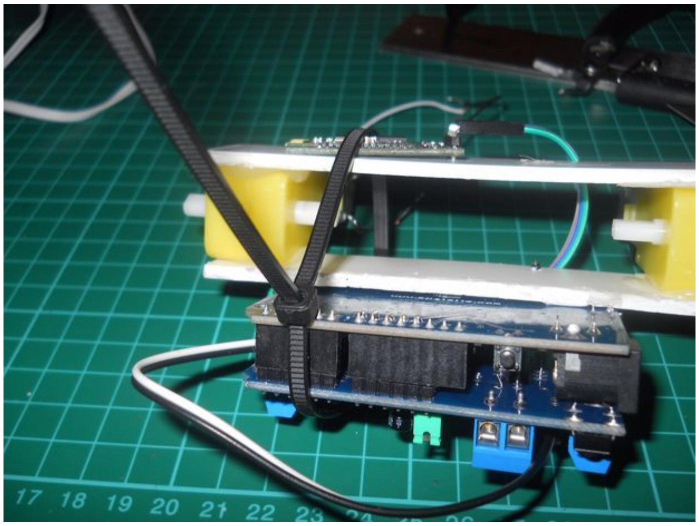
Step 20: The Body in the Sphere

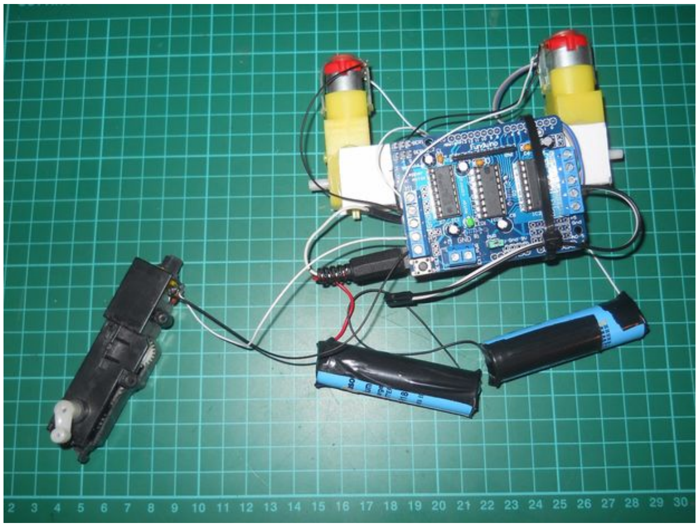
This is a tricky part: Cut two pieces of plastic and glue to the motors. The diameter of my sphere is 17cm so I used a 13cm long plastic piece. It was perfect for my droid. While working I removed the motor shield for a short time. Using ome screws and a zip tie I siuted the microcontroller to the body. The zip tie holds the the Arduino and the bluetooth module in place. The batteries will be installed later.

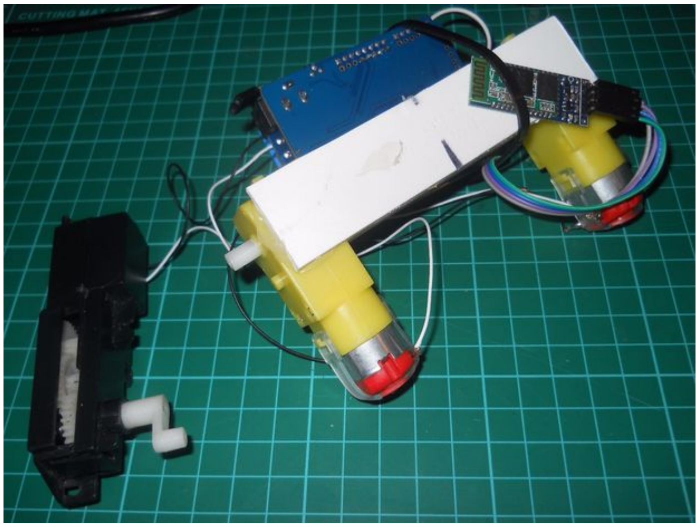


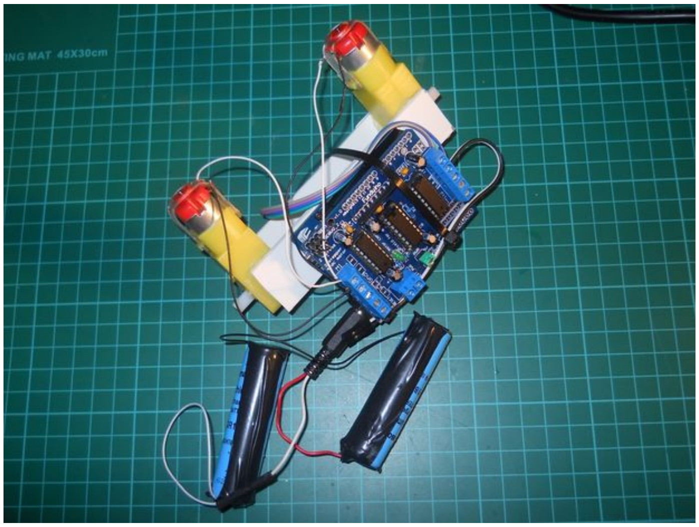






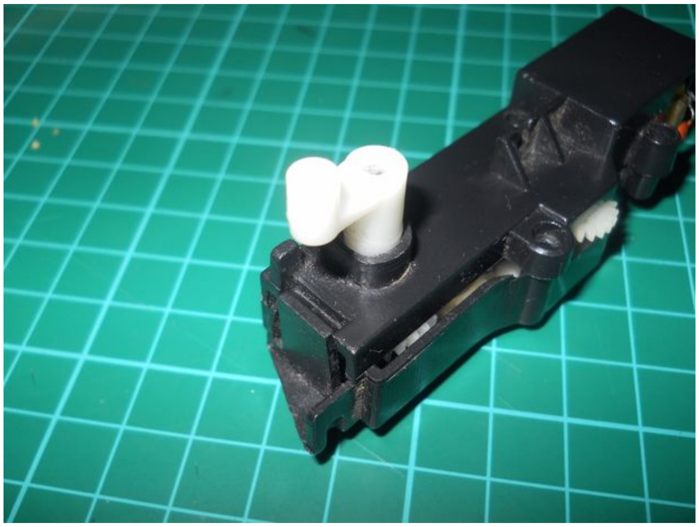


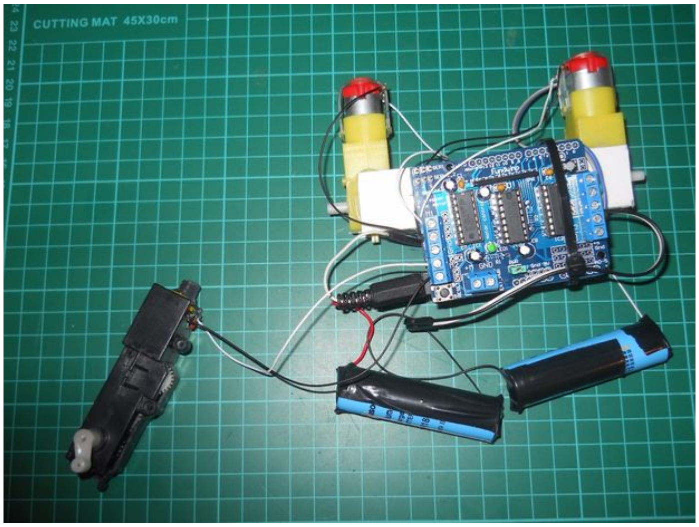


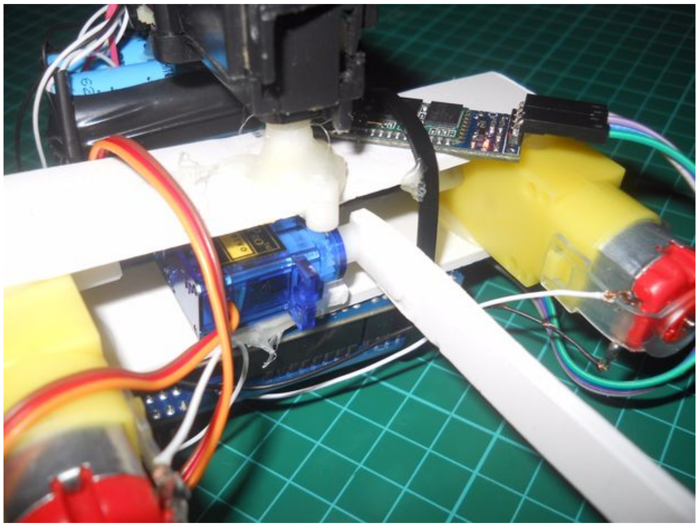


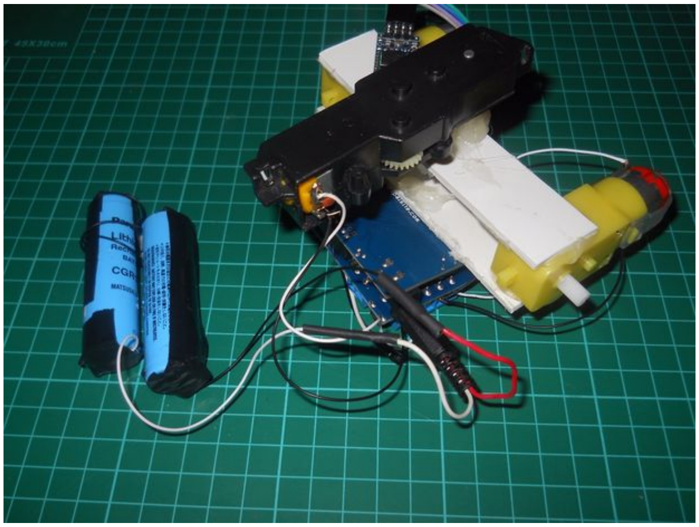
Step 21: Steering

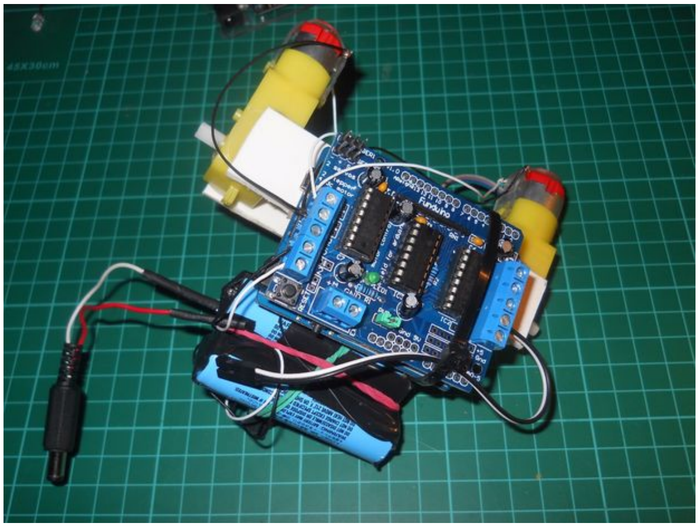
The time has come, use the Force or your glue gun and glue that third motor to the body. As you can see the motor and the gearing system is free, only the ending, the moving part is glued to the body. To that black motor system we'll connect the batteries. The batteries doubles as a power source and weight for steering.

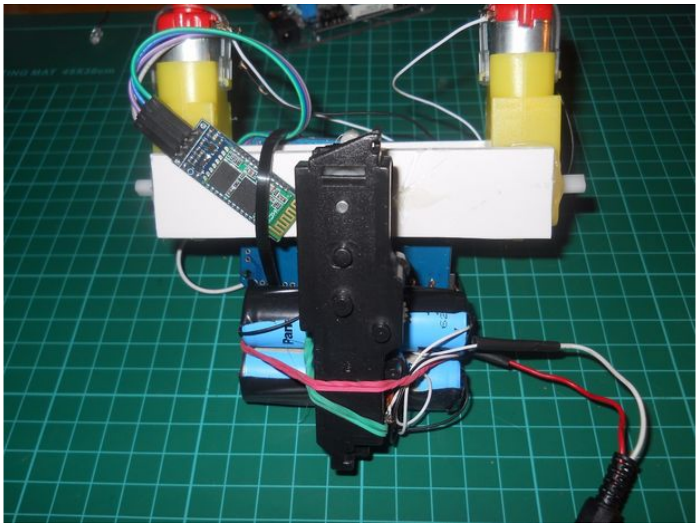


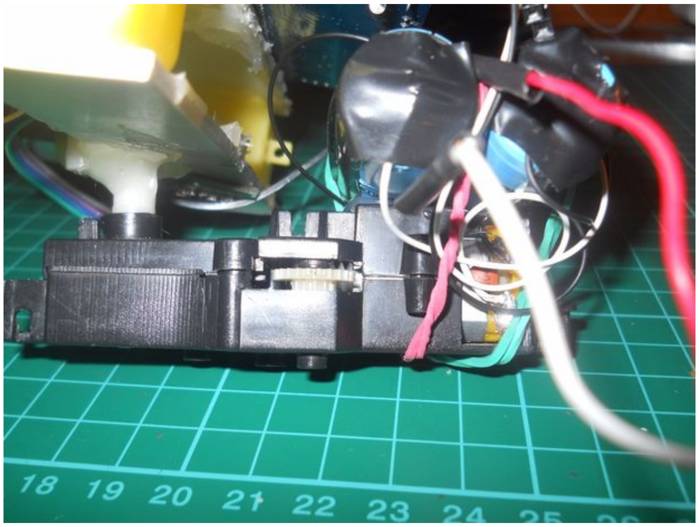








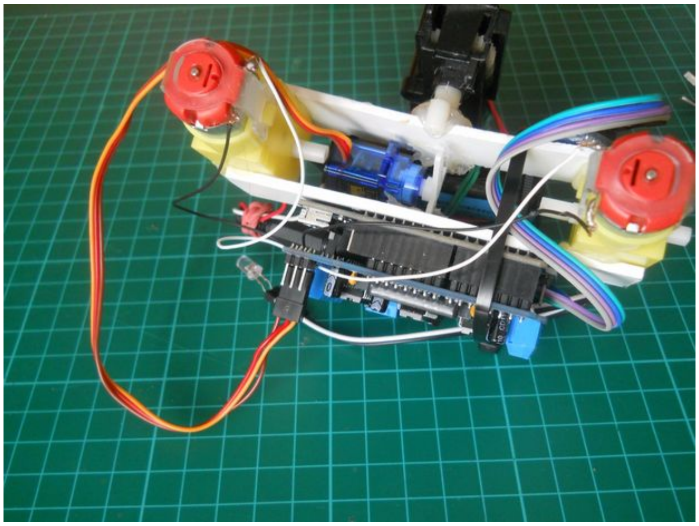


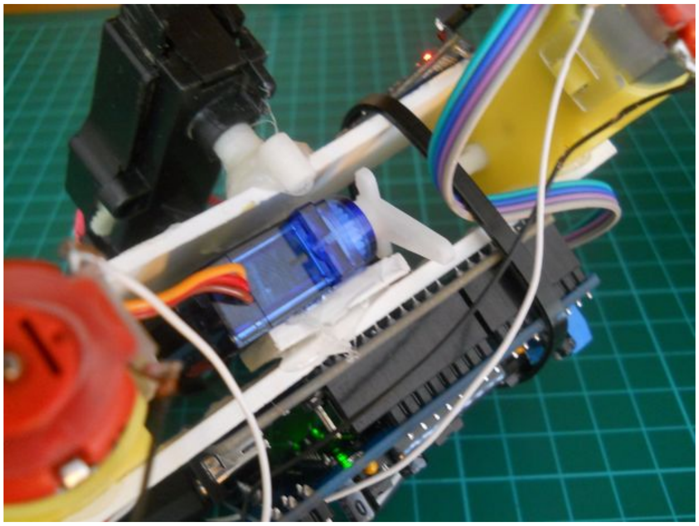


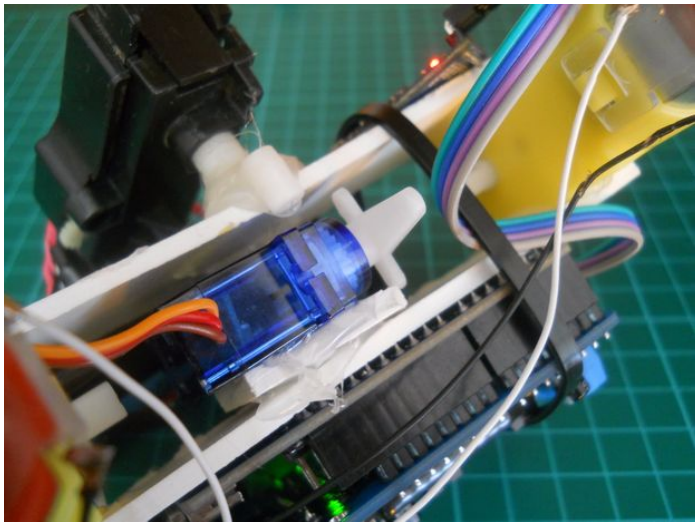
Step 22: The Servo

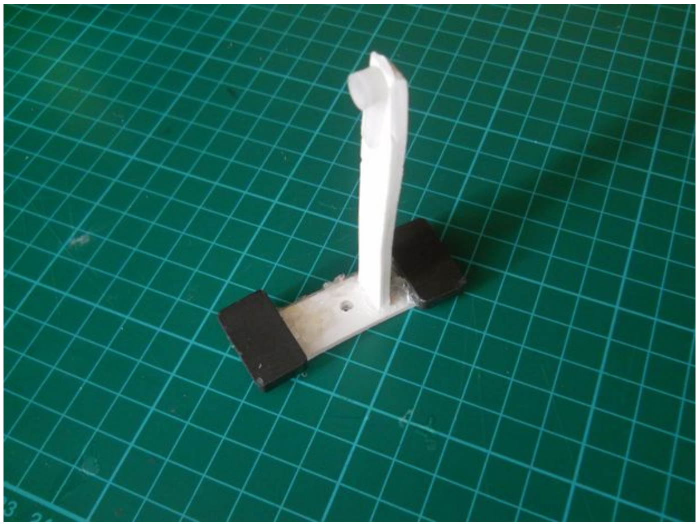
The role of the servo motor is to hold or tilt the head on the sphere. I used hot glue agian and glued the motor in the center of the body. Then with some plastic and two magnets I created a holding element. This stick will hold the head on the top in one place.

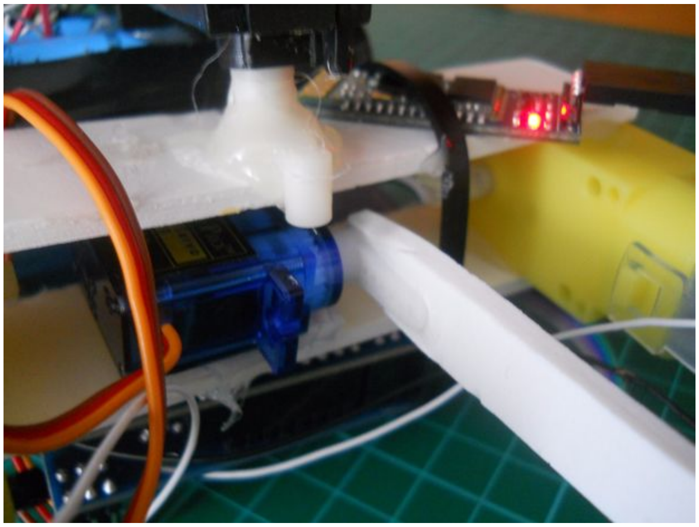
The head can be used to adjust the speed of the droid. If is placed in the front of the droid, will move much more faster.

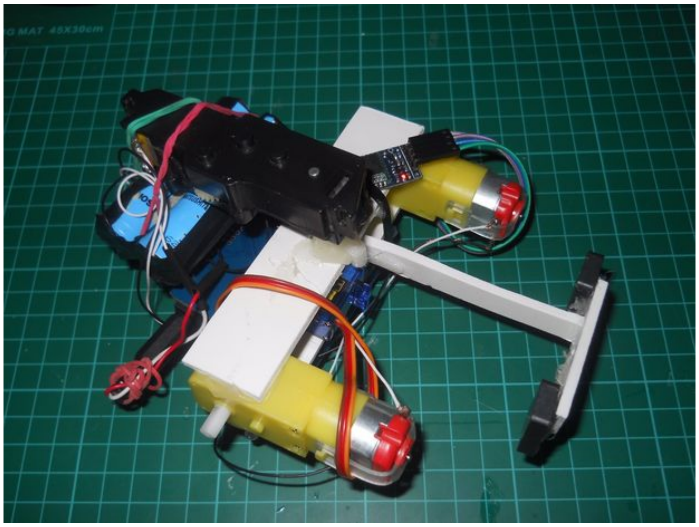








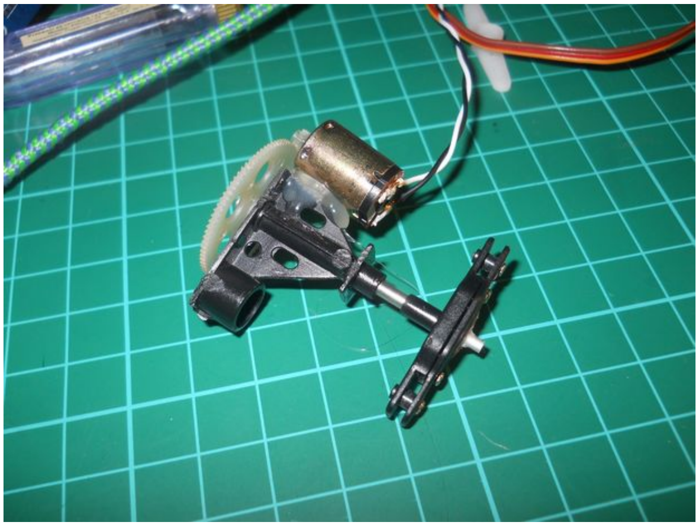


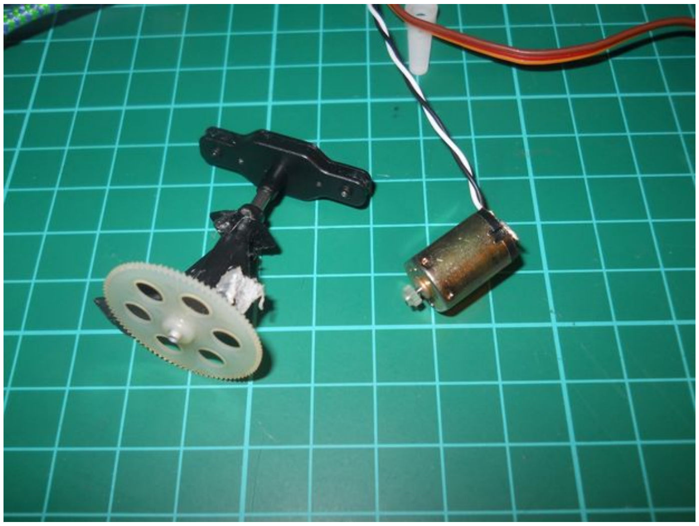


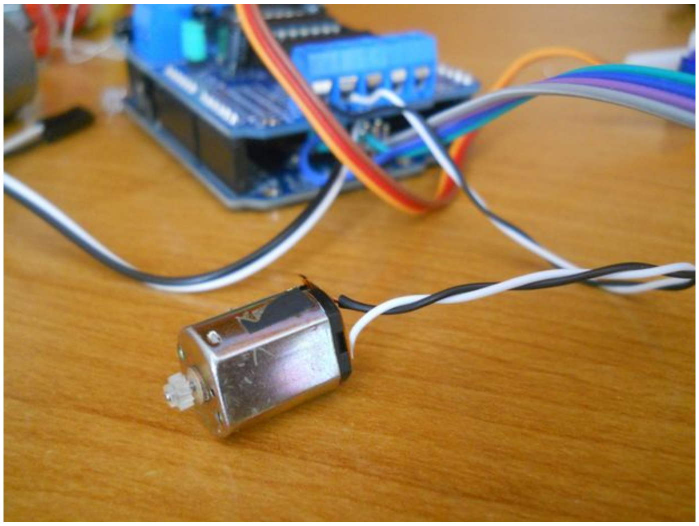
Step 23: Optional Upgrade (head spin)

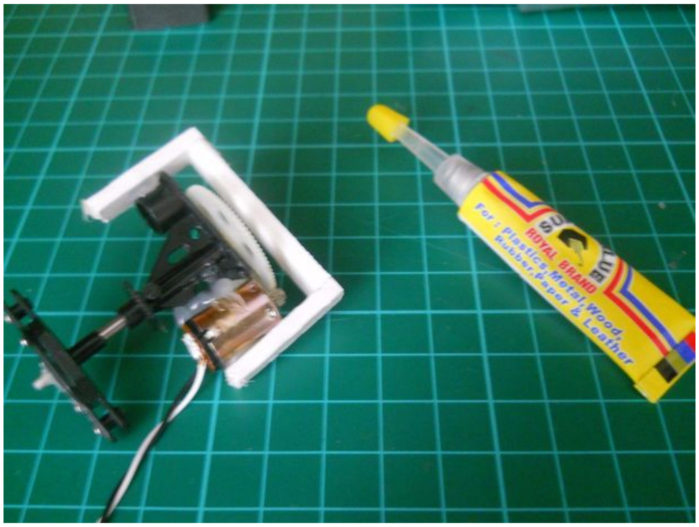
If you have a small motor like on the pictures., you can add an extra function to your droid. I made a plastic frame for it and glued the magnets on the top moving part. Attach this to M3 and to the servo motor.

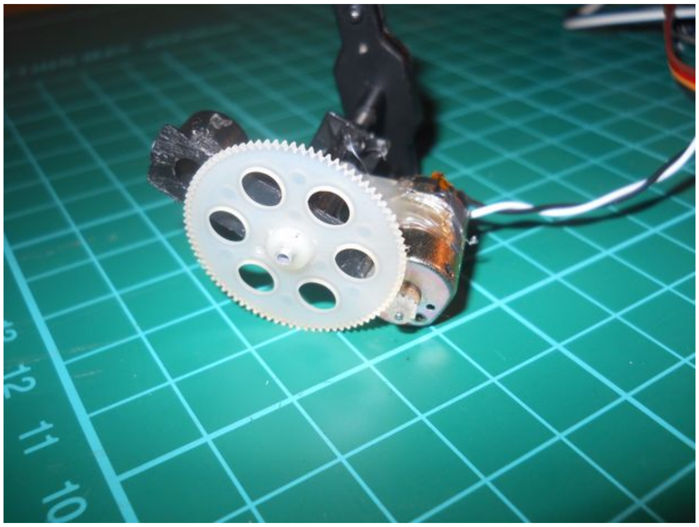
Note: I continue the Instructable without this, unfortunately my motor burned out.





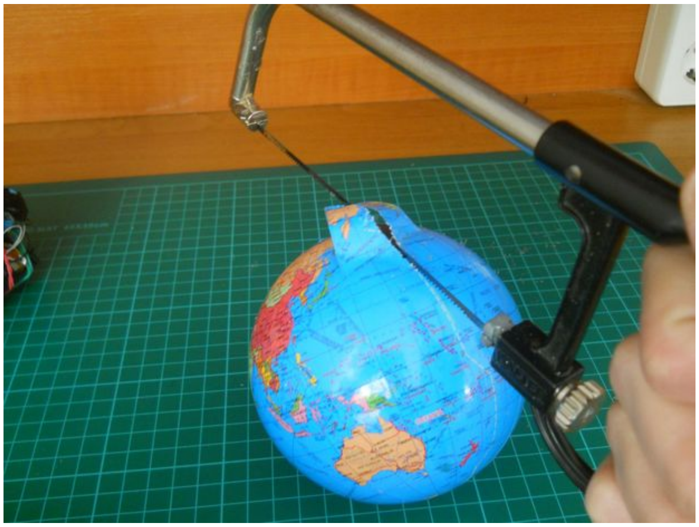


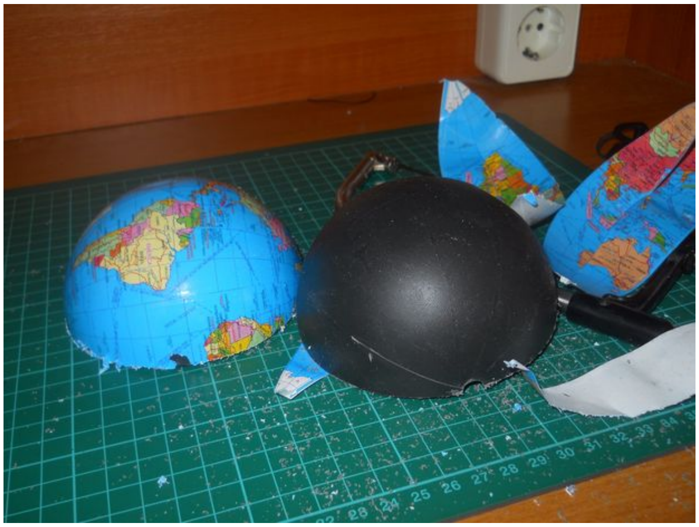


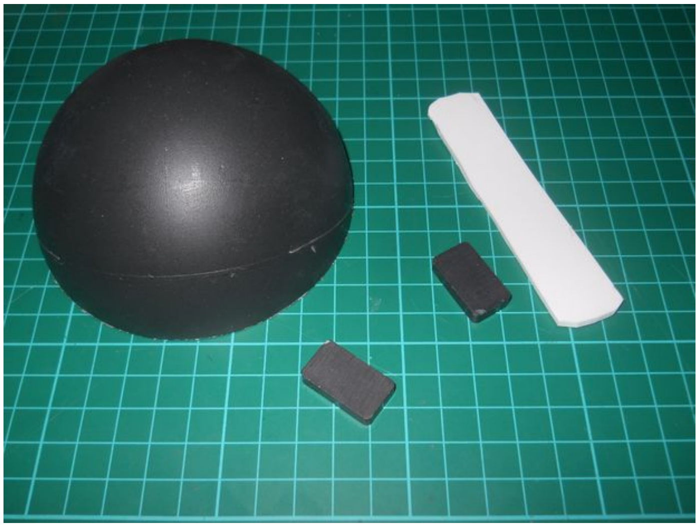


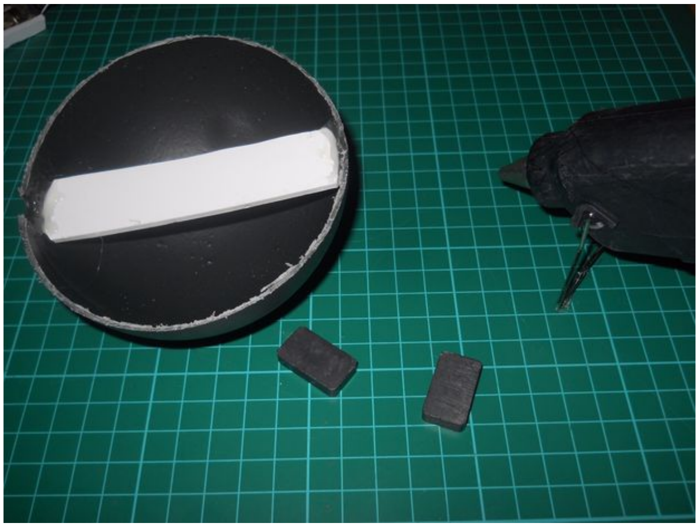
Step 24: The Head

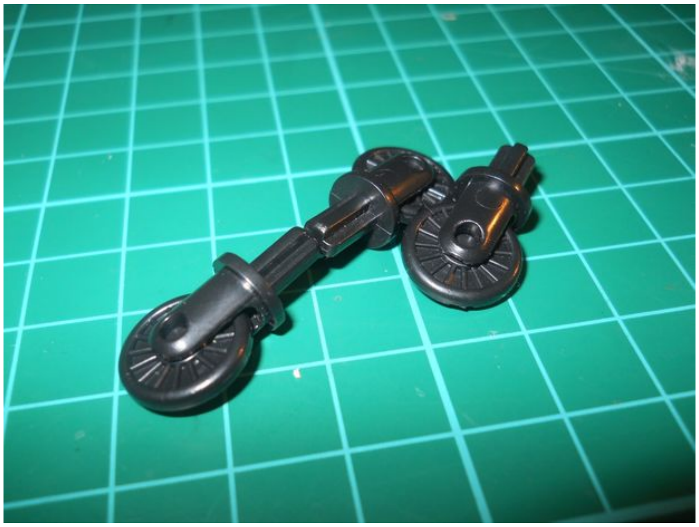
Cut the globe in half, then clean from the paper. You can use now anything as a holding element, I used plastic again and glued together with the magnets. If you want you can add Lego wheels for better motion. The wheels make the rotation much more easier.

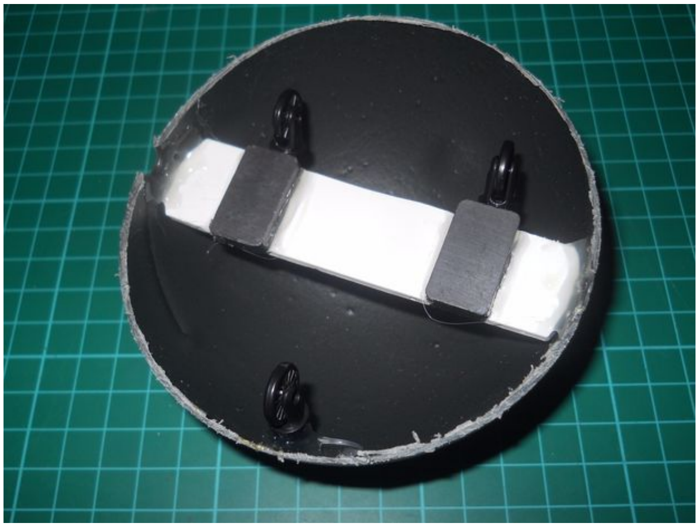


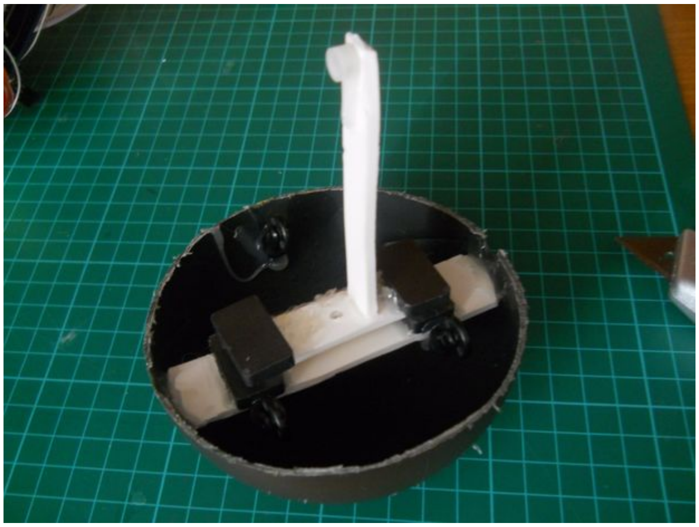








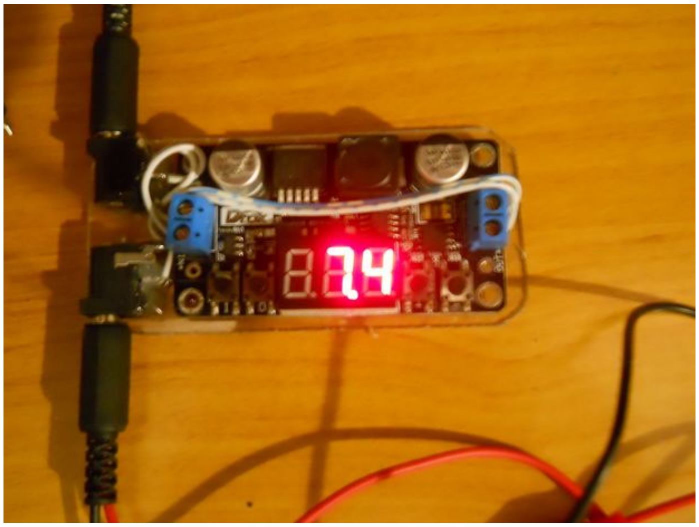


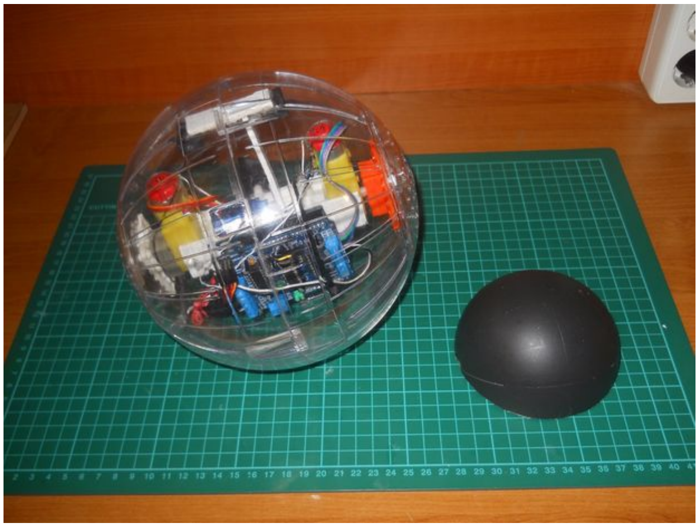


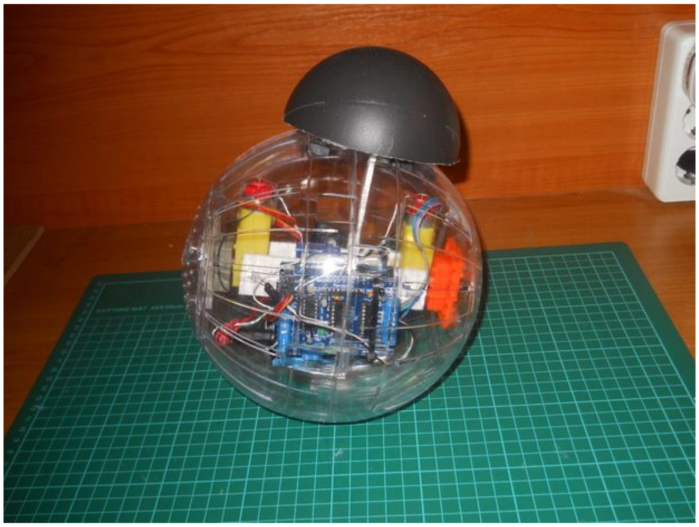
Step 25: Quick Test (Again)

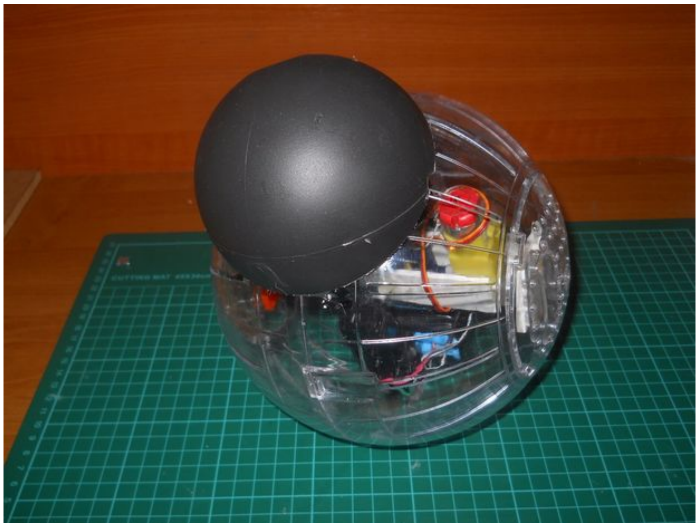
Use the power of the Dark side or electricity and charge up the batteries. On the first picture you can see that I charged up my battery pack using a small variable power supply. After 2 hours the battery was charged up with 400mA charging current.

If your droid is ready try it out. Put the head on the ball then try to control with the phone. First time was tricky, but after about five minutes was very easy. The head moved very



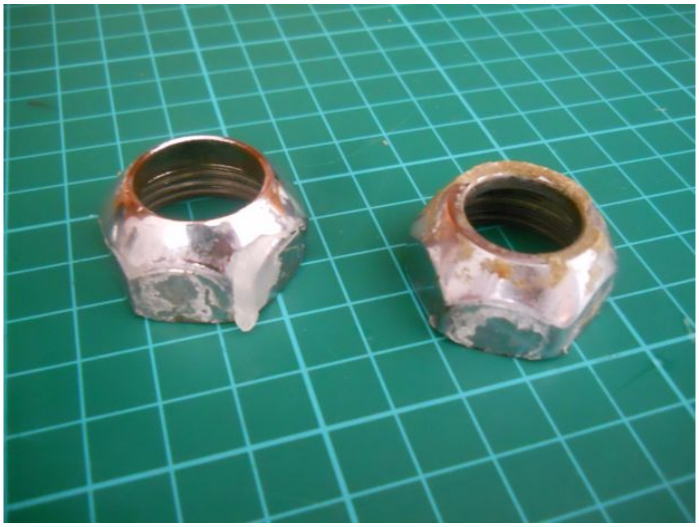


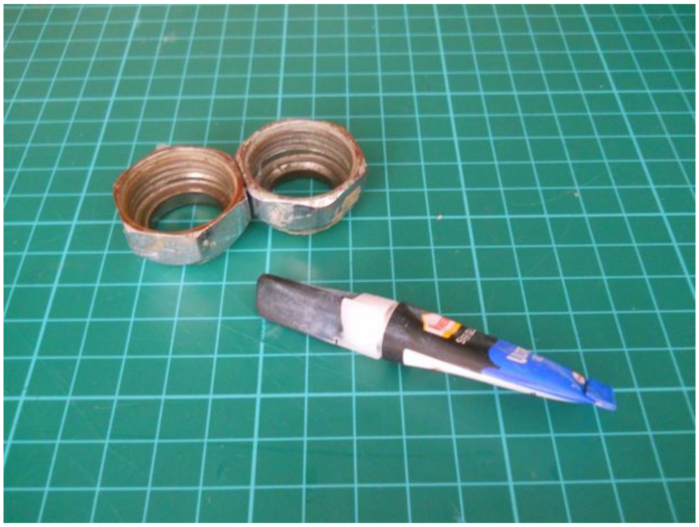


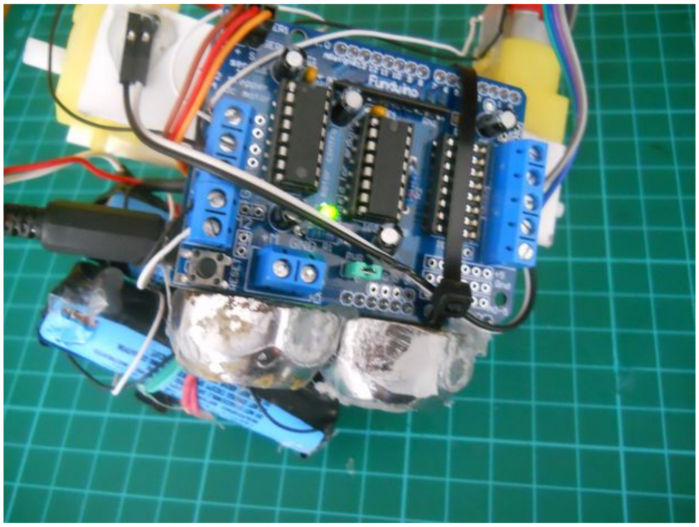


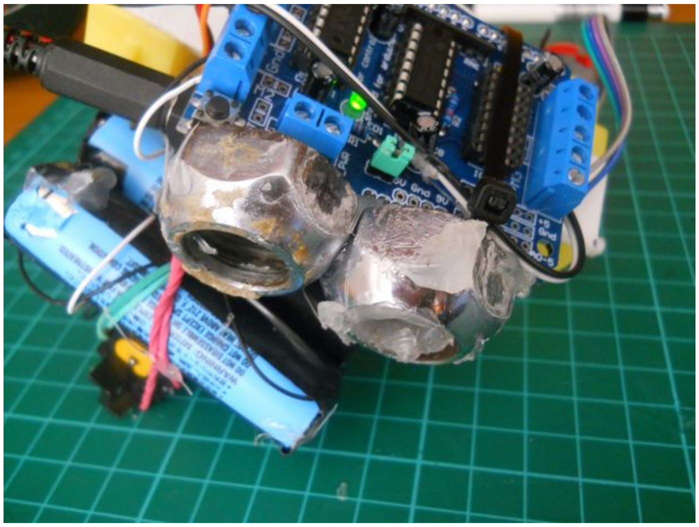
Step 26: Adding Some Extra Weight For Better Steering

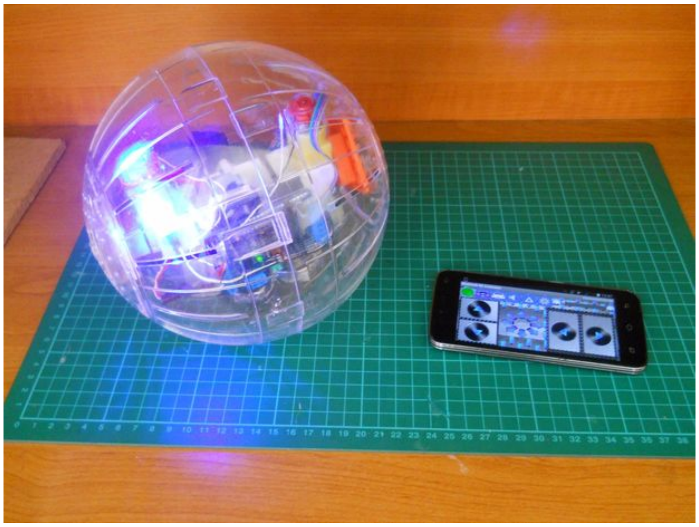
As you see a took some metal srews and simply glued on the electronics. Now my spere will move little bit faster, and it's much more easier to steer.





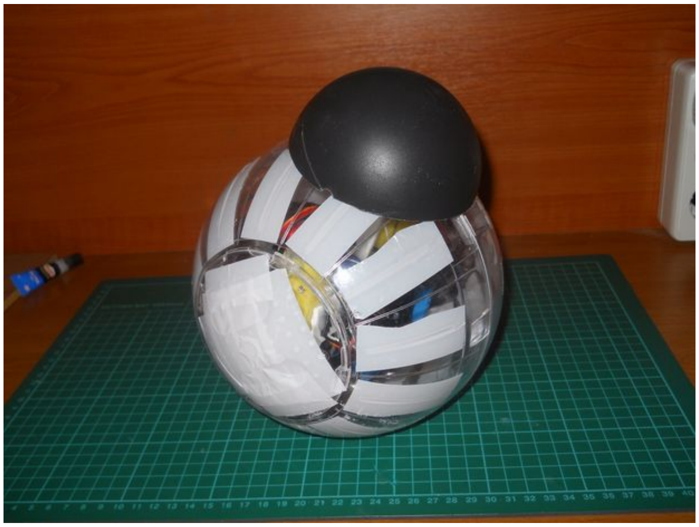


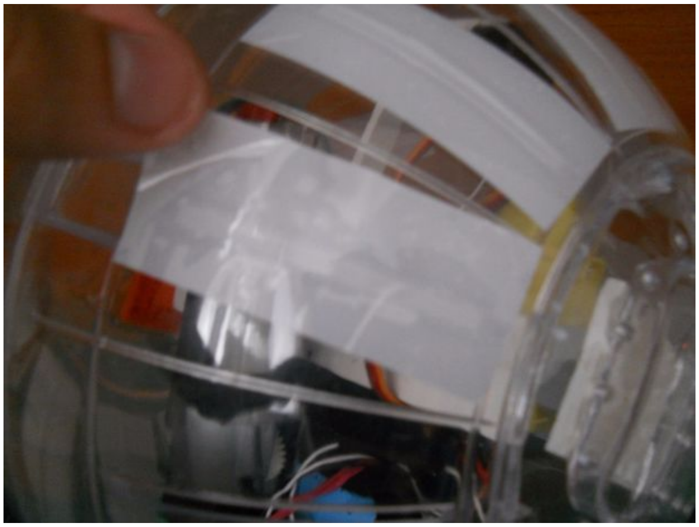


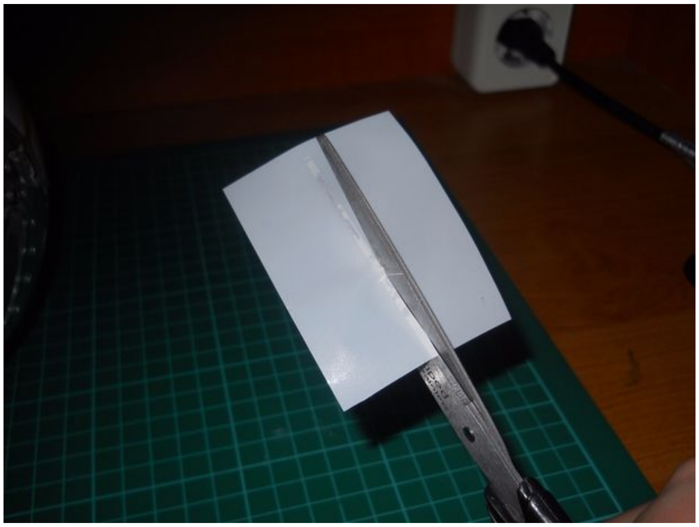


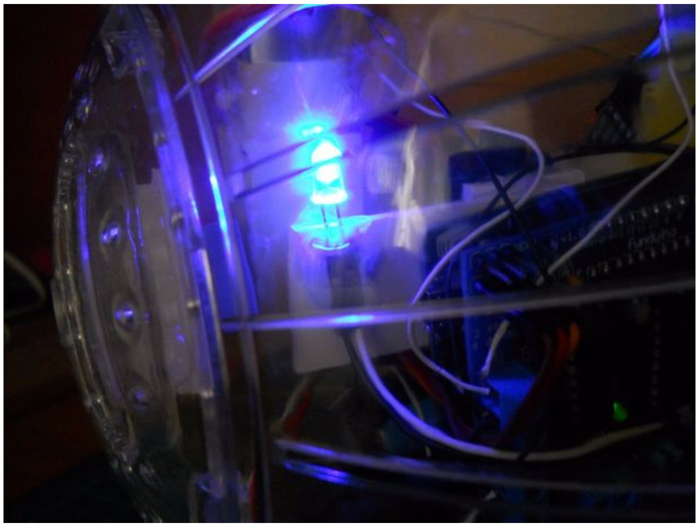
Step 27: Preparing for Paint

On this hamster ball were a lot of holes to ensure enough fresh air or the pet, but we don't need these holes now so I simlply neautralyzed them using some ductape. Worked well!





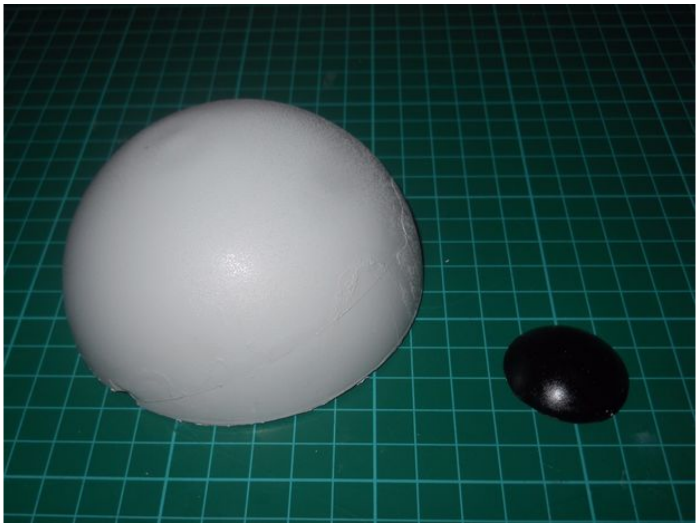




Step 28: Painting

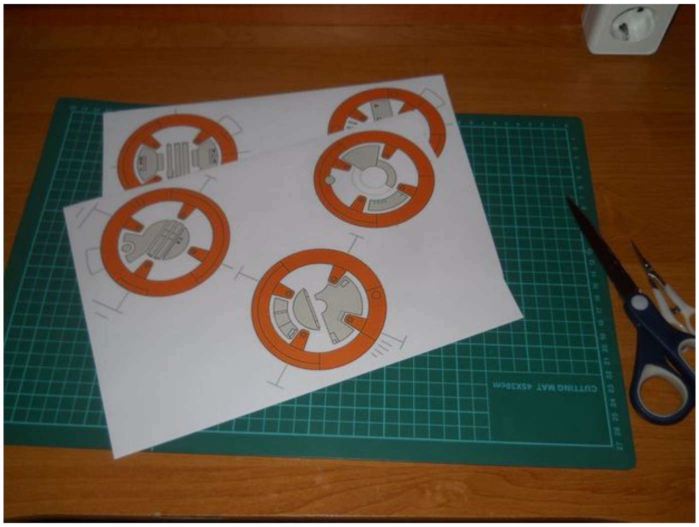
I took apart the board then I painted the entire chasics with a white paint spray. That was easy but I needed to paint two times to fix the errors caused by the ductspe. So remove the ductape, fill the holes woth silicon (from glue gun) then paint again.

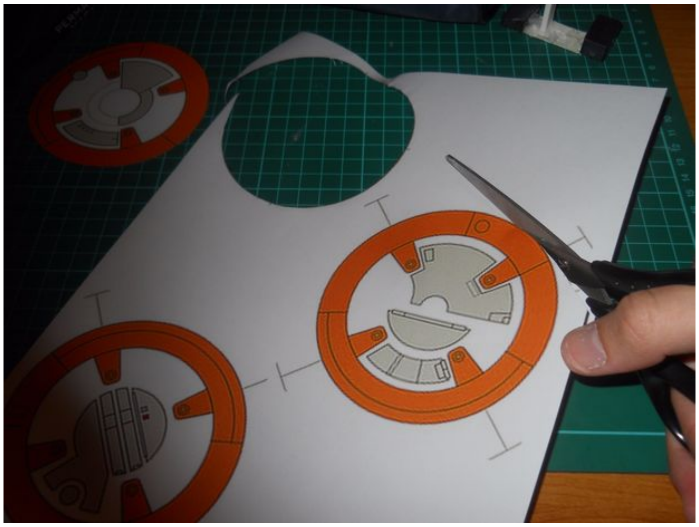
Note: The painting was the hardest part for me, but I fixed everything while working. So I survived...

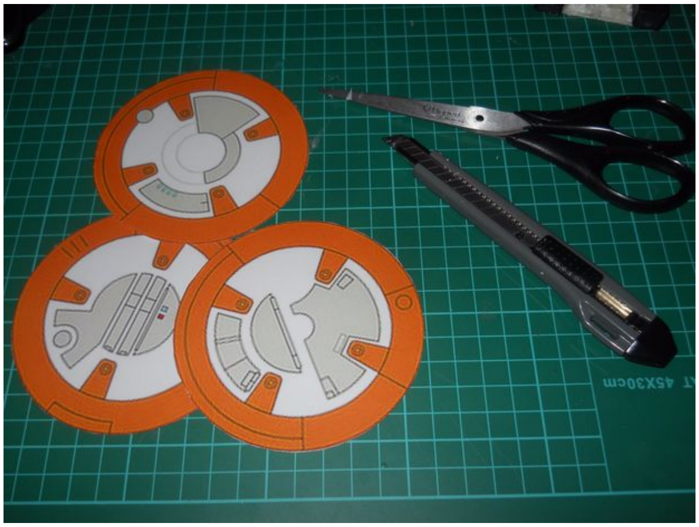


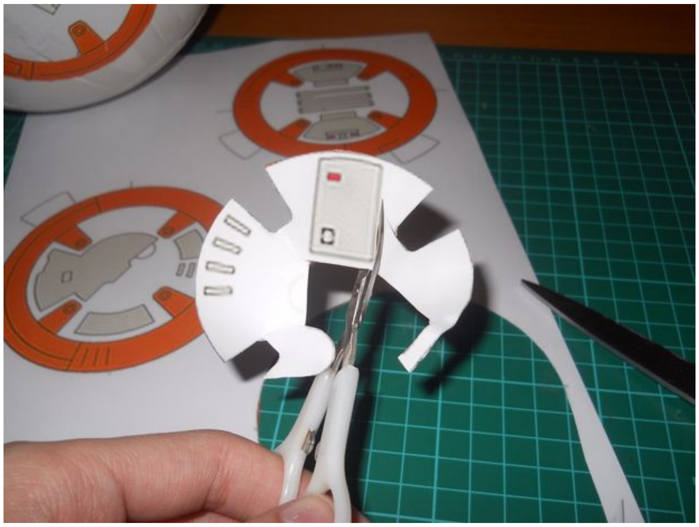
Step 29: The Pattern

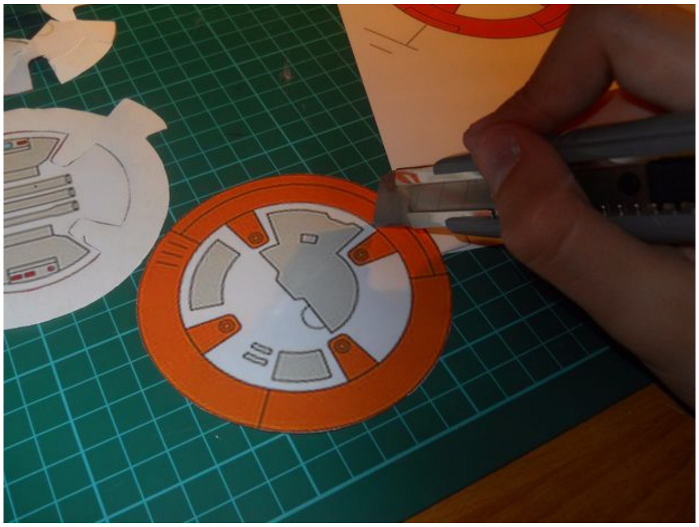
Download the attached .pdf file then print it on a self-adhesife plastic sheet just like me. I haven't at home a paper like this, so I went to the local printer office and I printed the patterns for about 0.8 bucks. After printing take a cutter or scissors and cut every single part. Of course you'll need all six circles.

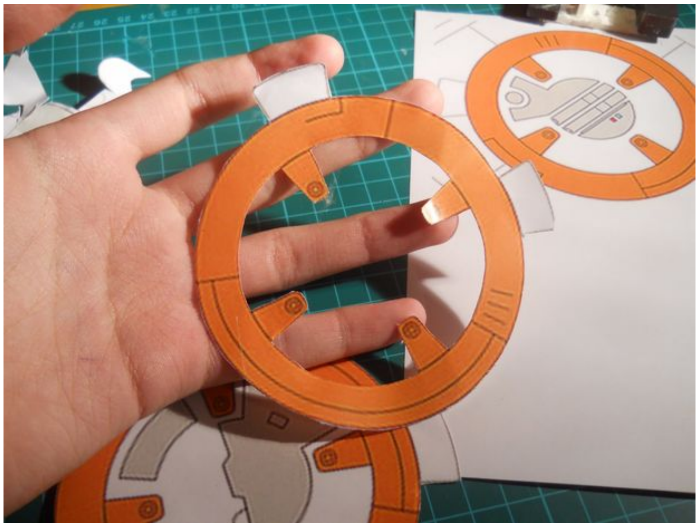










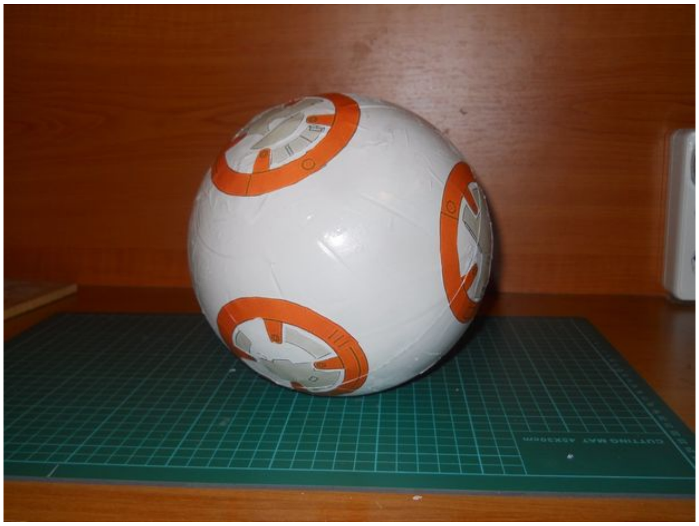


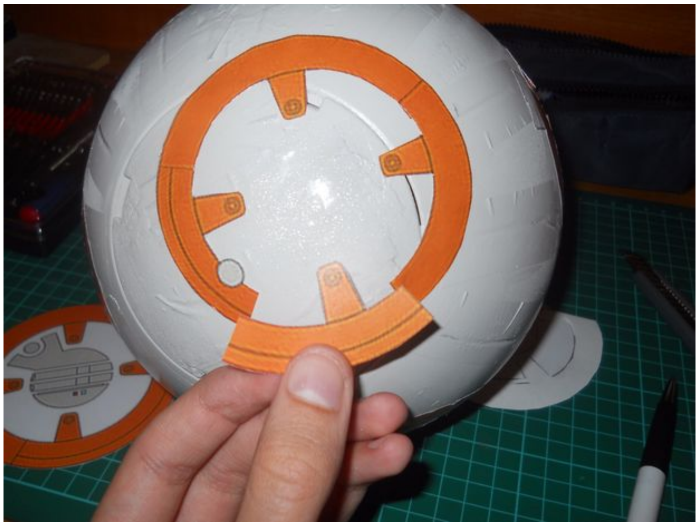
File Downloads

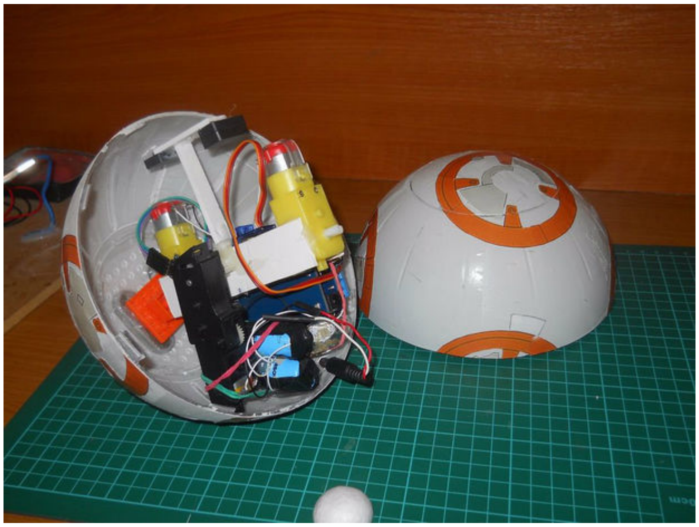
BB-8 Pattern.pdf (1 MB)
[NOTE: When saving, if you see .tmp as the file ext, rename it to 'BB-8 Pattern.pdf']

Step 30: Gluing the Pattern

When you are done with the cutting put the patterns on the droid. I cut every circles in four parts to beware the creases.

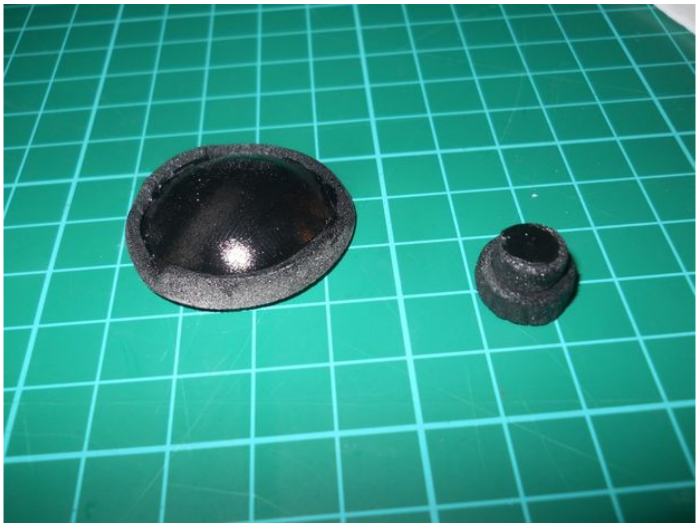


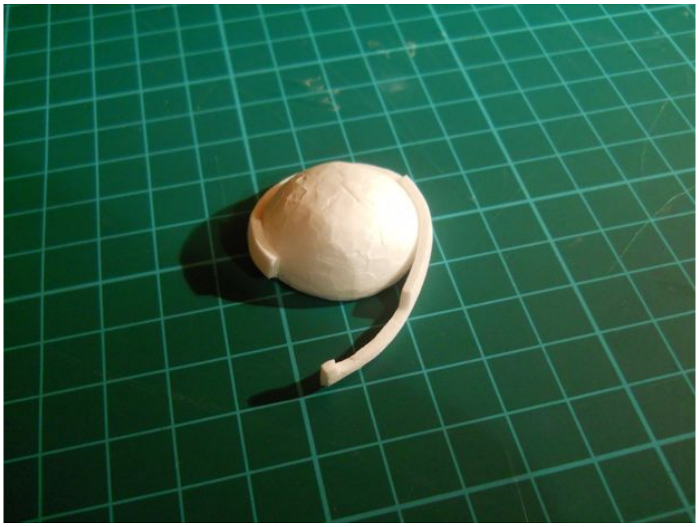


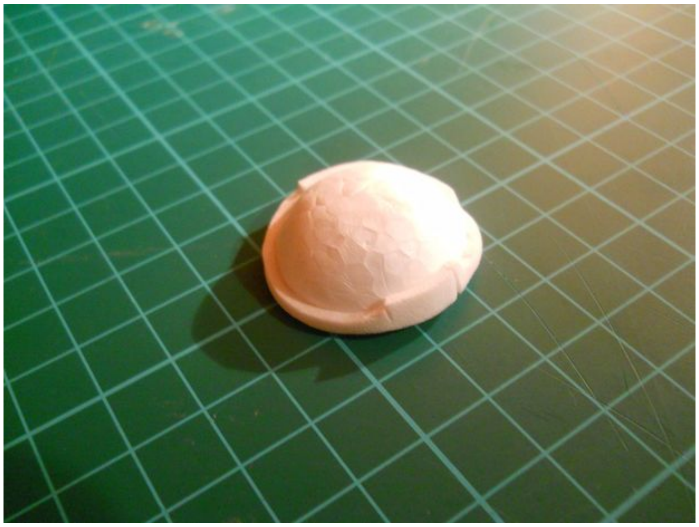


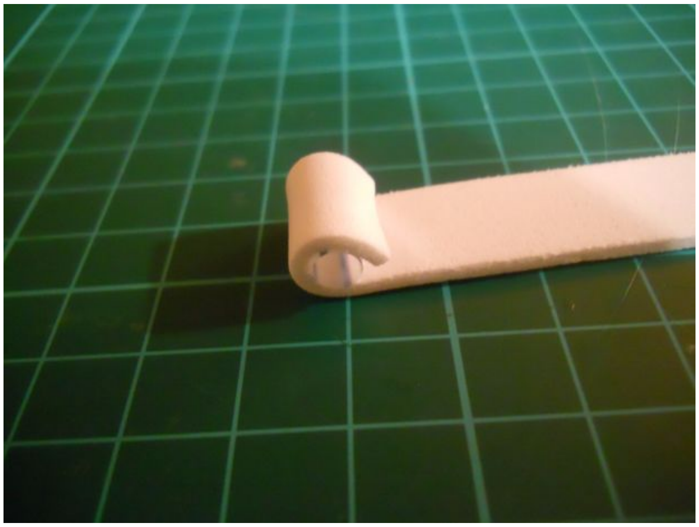
Step 31: The Eyes

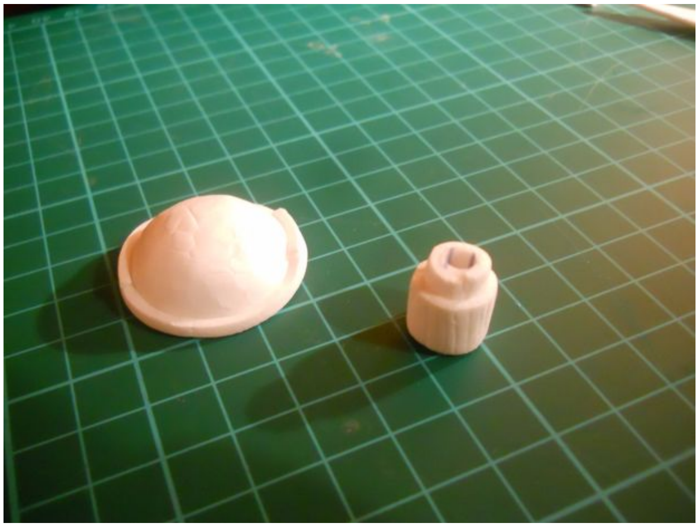
Using some styrofoam I made the two eyes and paintend them with a black paint spray. They look pretty nice.

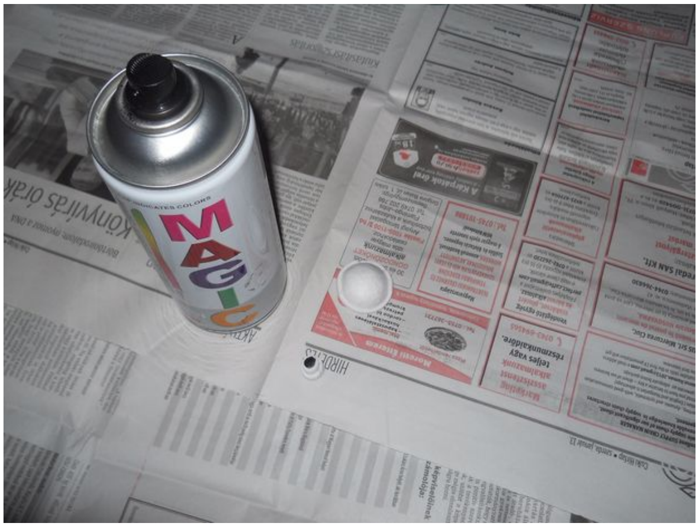


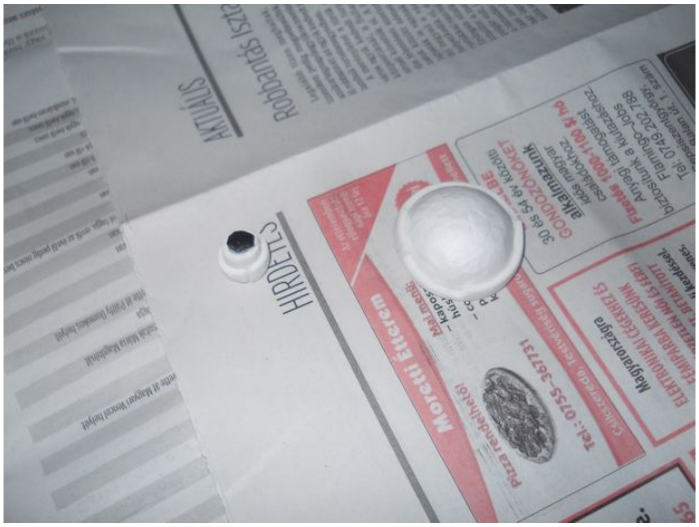






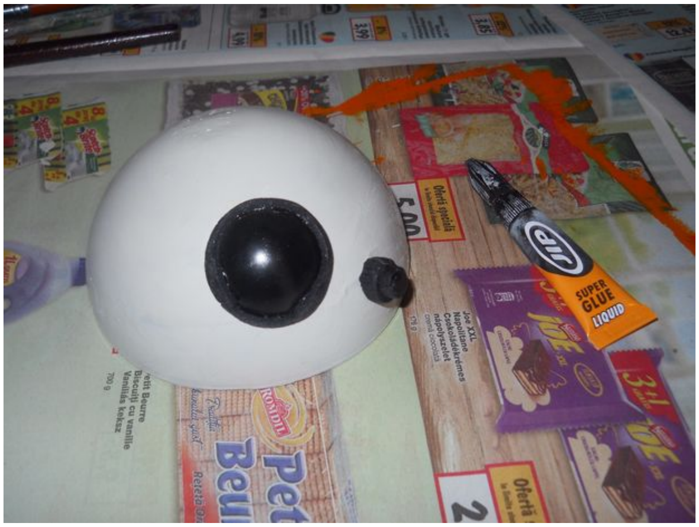


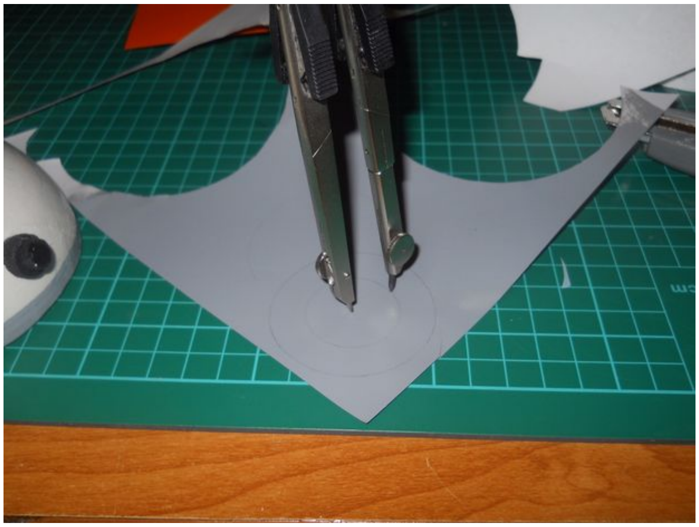

MAGIC

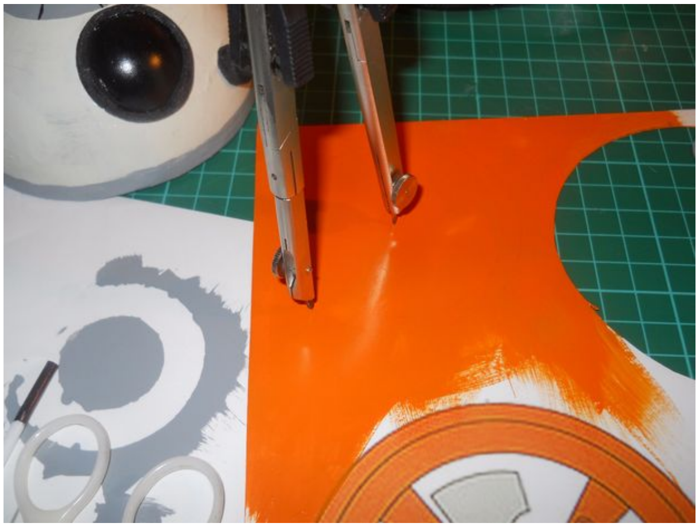


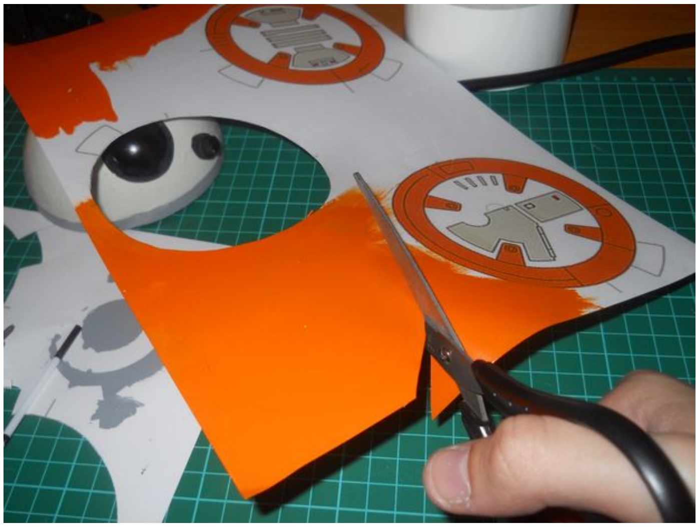
Step 32: The Head Pattern

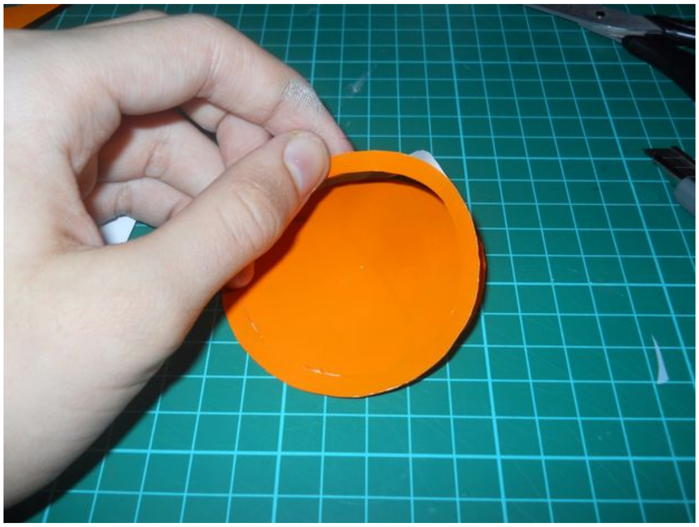
First glue the eyes on the head. Use super glue. I painted the remaing self-adhesive paper with orange and grey colors. Using a calliper I made the patterns that were needed for the head. Using a cutter I tried to cut perfect these patterns.

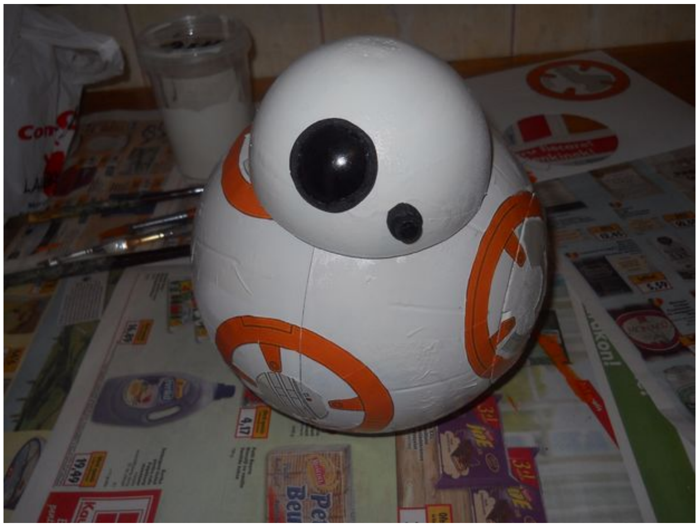






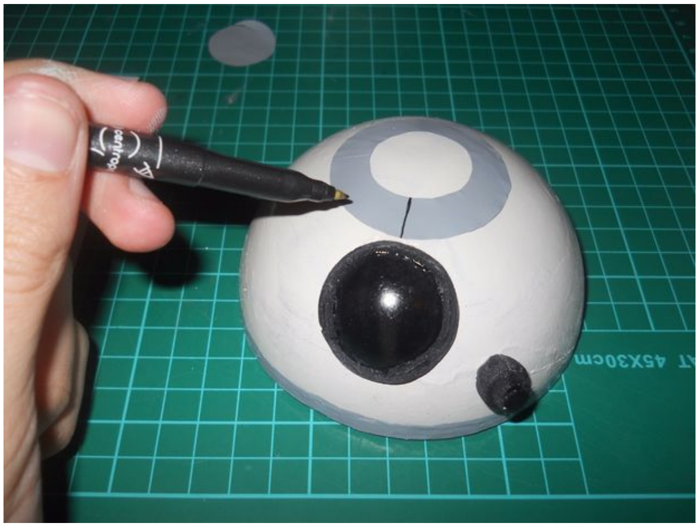


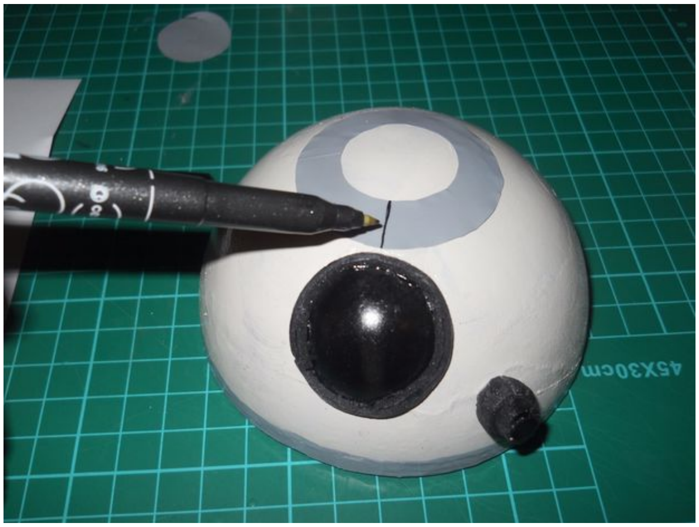


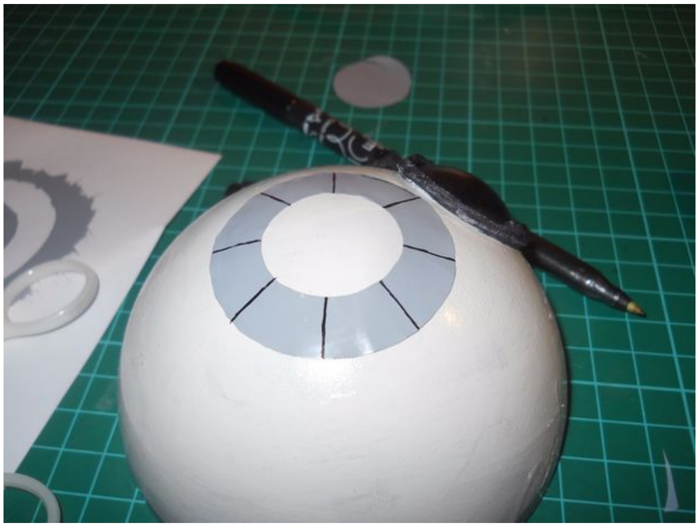


Step 33: Decoration of the Head

Put the grey circle on the head then using a marker make some lines on it.

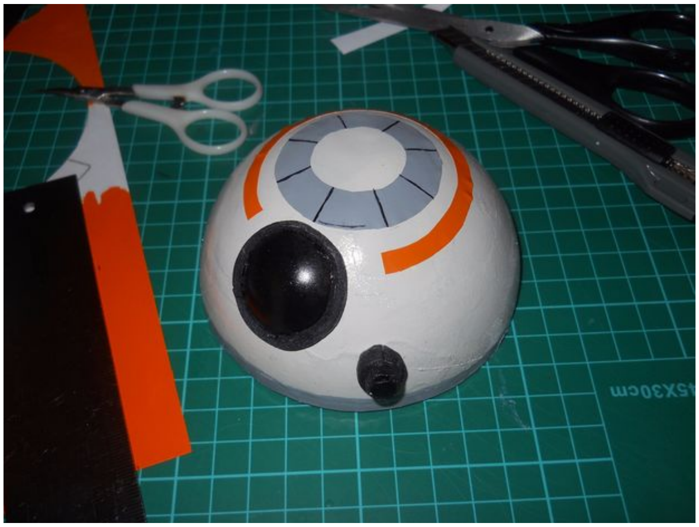


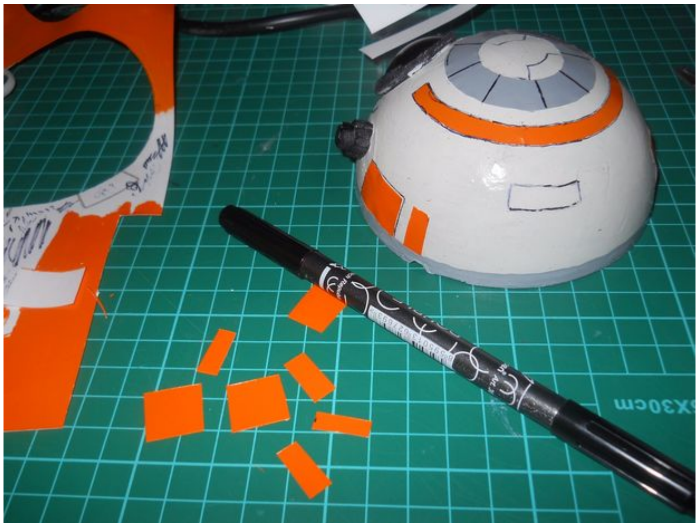


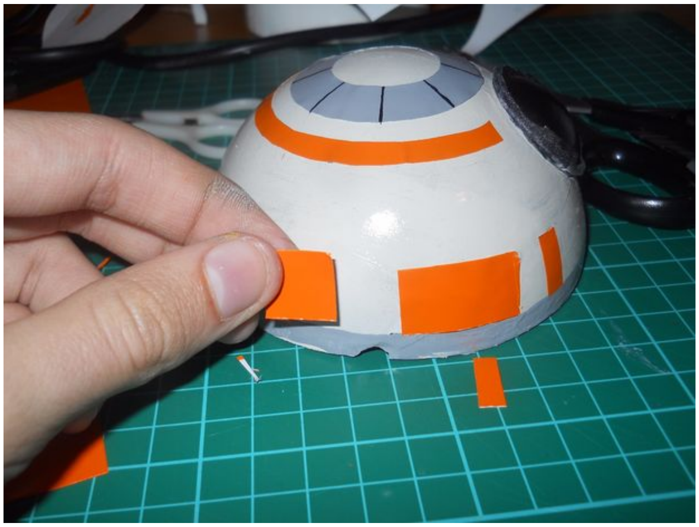


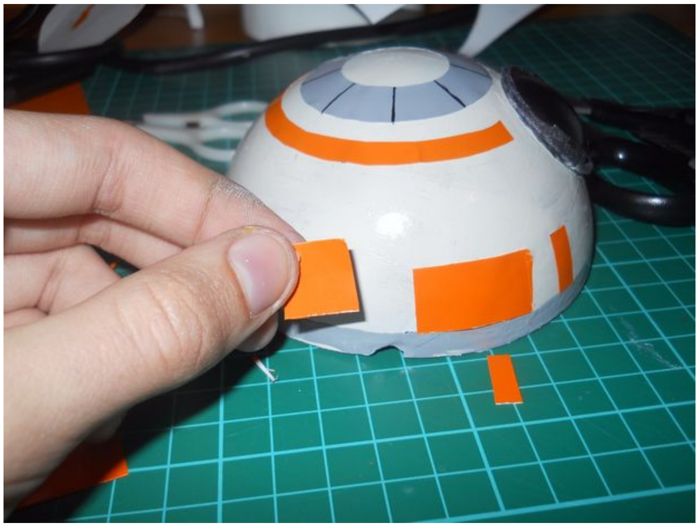
Step 34: Still Decorating...

I think this step doesn't need any extra explanations, just follow the pictures. I used two small white straws as antennas.



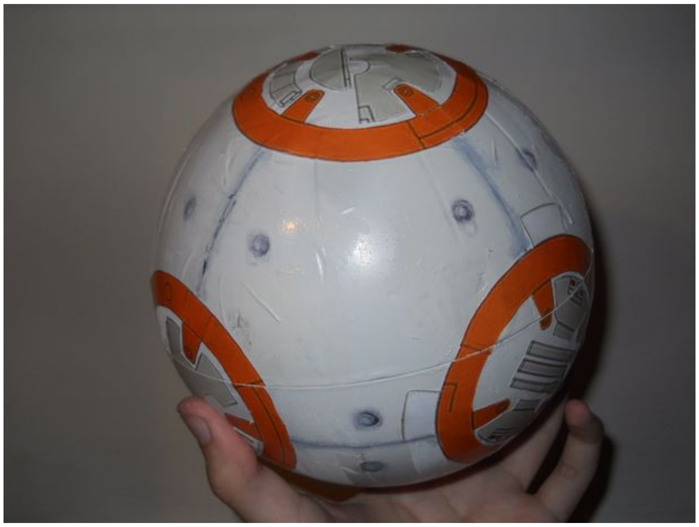


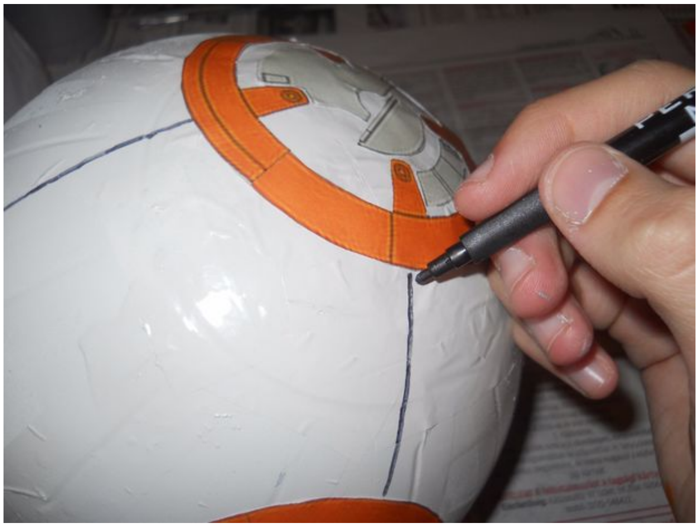




Step 35: Details On the Body

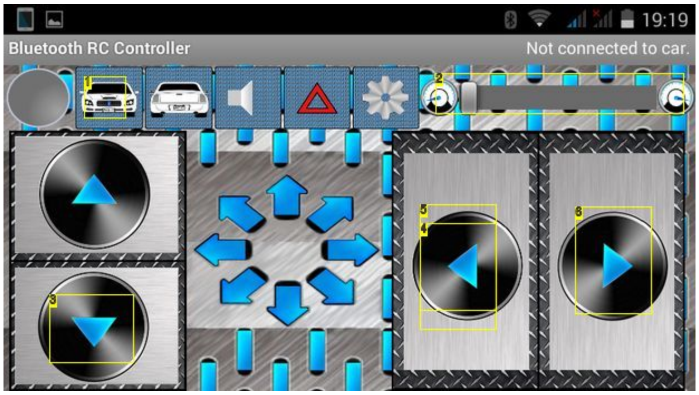
With a marker I made dottles and the lines that were on BB-8. Then I used a brush to make the lines paler to earn a dirty effect. In the movie he was very dirty.

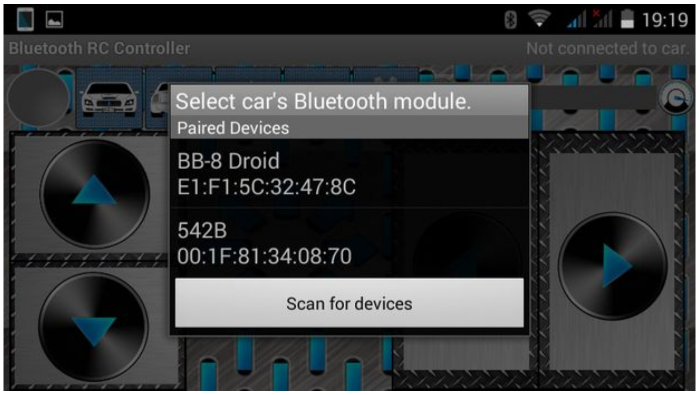




Step 36: How to use the app?

I can gaurantee that is very easy to use the app. Connect to your BB-8 then control it.





Select car's Bluetooth module.

Paired Devices

BB-8 Droid
E1:F1:5C:32:47:8C

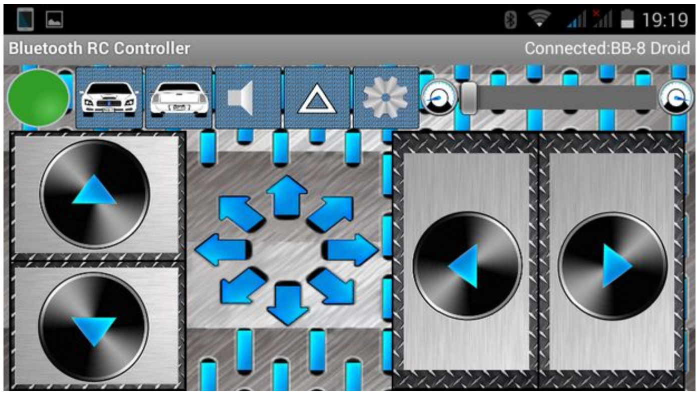
542B
00:1F:81:34:08:70

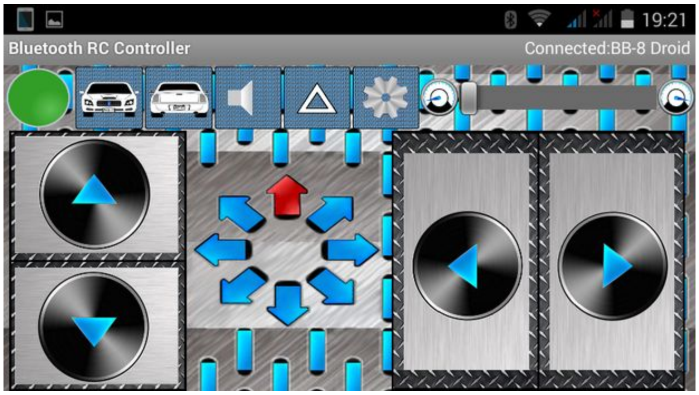
Scan for devices

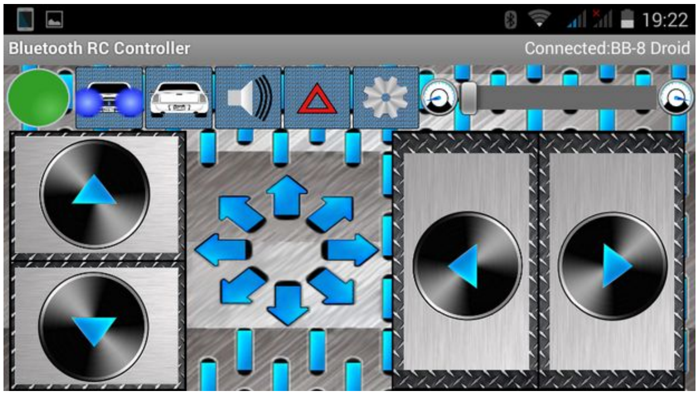
Image Notes

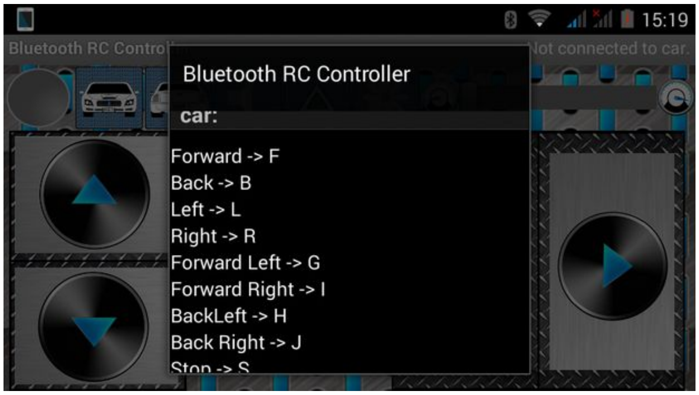
1. Light On/Off
2. Head Motion

3. Backward
4. Left
5. Left
6. Right









Bluetooth RC Controller

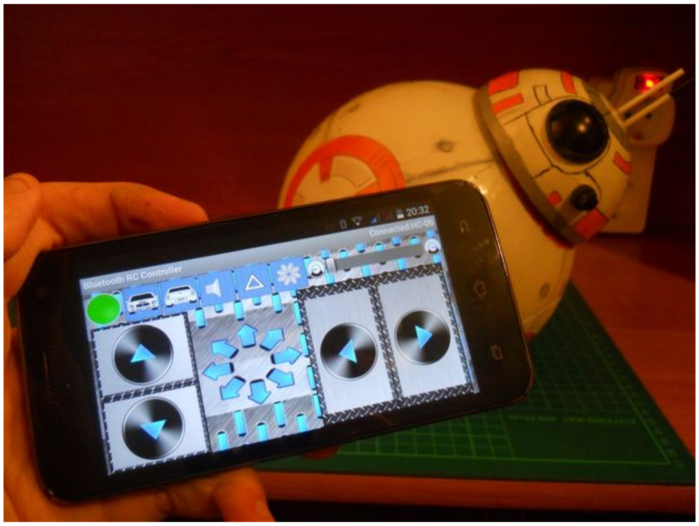
car:

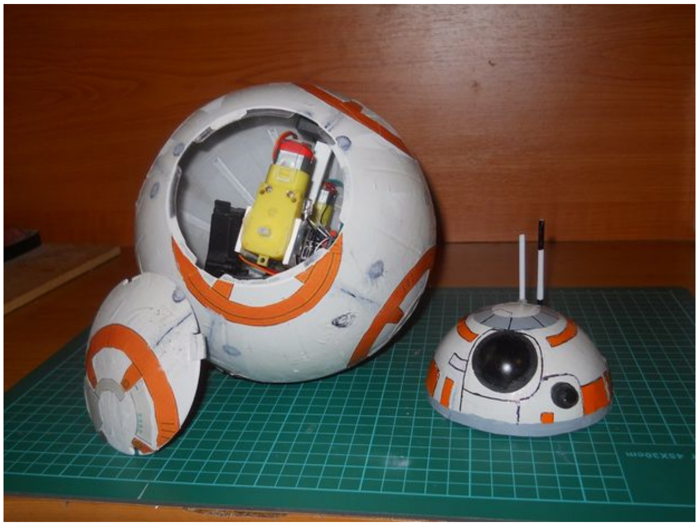
Forward -> F
Back -> B
Left -> L
Right -> R
Forward Left -> G
Forward Right -> I
BackLeft -> H
Back Right -> J
Stop -> S

Step 37: This is droid you are looking for!

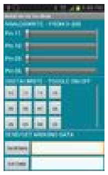
Thank you for watching my Instructable. I really hope you liked it. I'll add a video an some new explanations tomorrow. **If you liked give me a kindly vote! Have fun with this droid.**

May the Force be with you!





Related Instructables



ArduDroid: A Simple 2-Way Bluetooth-based Android Controller for Arduino by techbitar

1:1 Scale Replica BB-8 Prop by 1truedrum

3D BB-8 From Star Wars | Bead Sprites (Perler/Hama Beads) by Shendijiro_Ura

Mid-size BB8 With 257 Lego Parts by DanielS296

Star Wars Cross Stitch: Droid Booties by hatstand4510

DIY Star Wars Christmas Ornament by inspiretomake

Comments