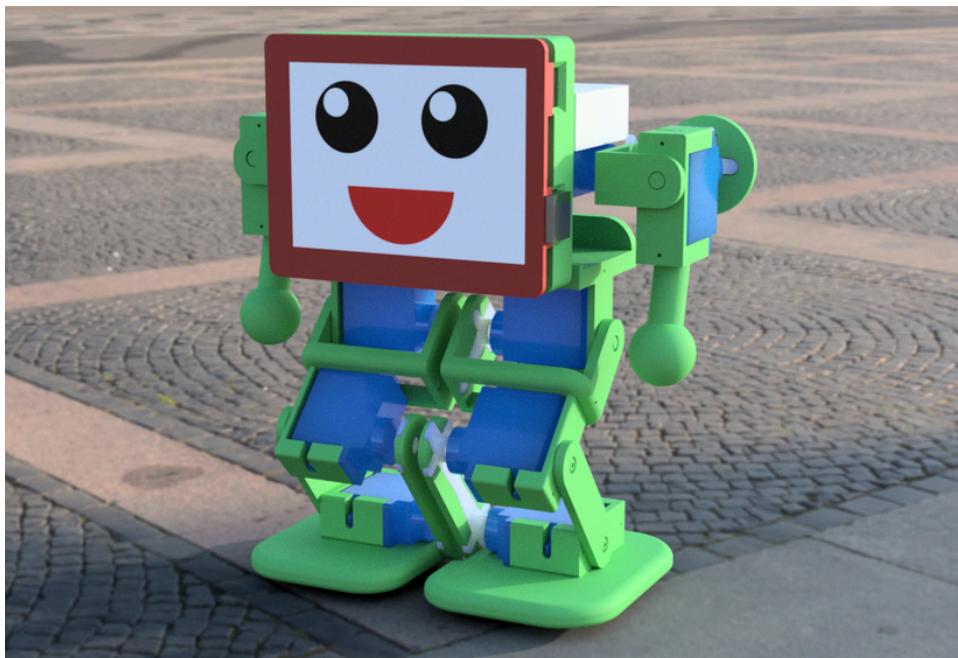


Ottis Robot, the Arduino Robot That Can Walk and Perform Visual and Audio Gestures

By [EduardoH20](#) in [CircuitsRobots](#)



Introduction: Ottis Robot, the Arduino Robot That Can Walk and Perform Visual and Audio Gestures

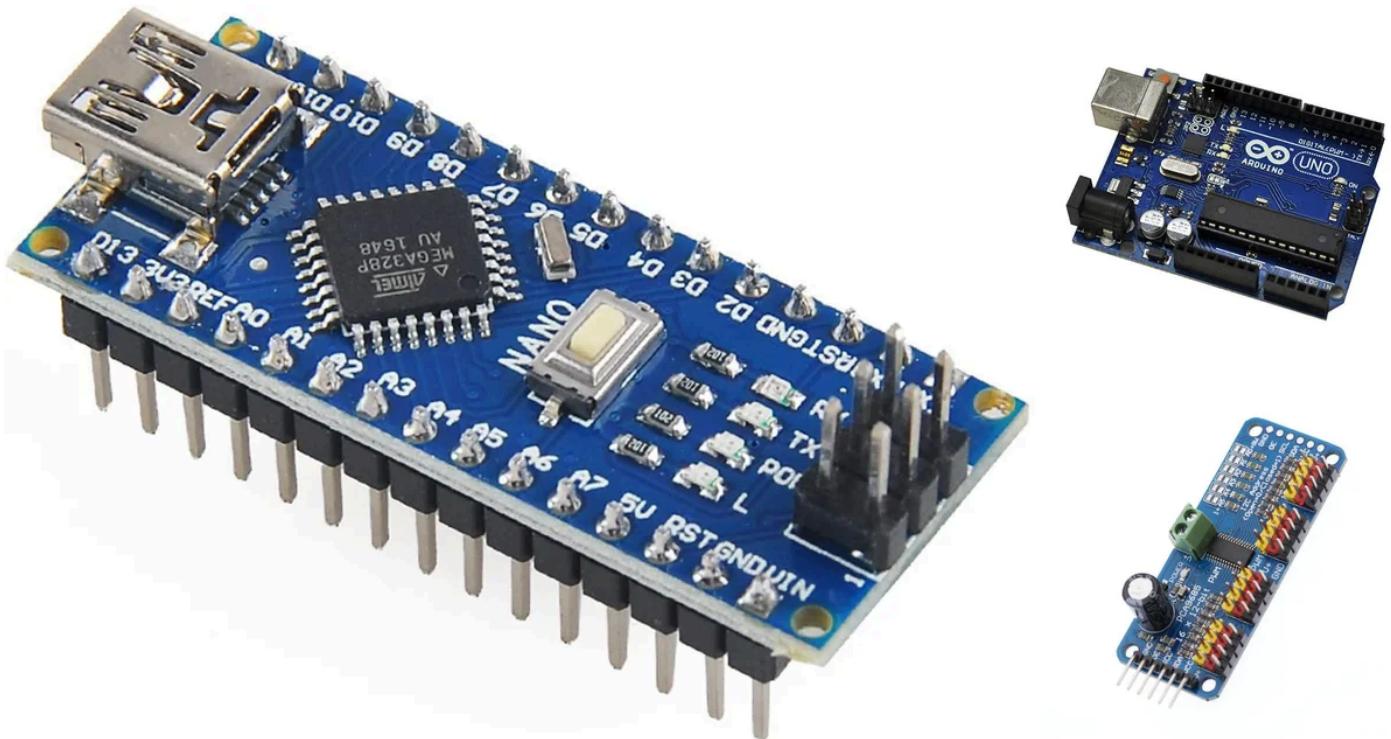


[Fusion Projects »](#)

Hello, good day! My name is Eduardo, since I was very young I have been interested in robots and all kinds of mechanical devices, I have studies in electronics and I am currently an engineering student, I love the arduino platform and I have done many electronic and robotics projects, you can see some of them on my facebook page [Edu Electronics](#). I will also design in 3D with Fusion 360, which is a very versatile software for all kinds of mechanical projects, I made this robot based on the human form, I just wanted to make it a little more "human", adding visual gestures with a display TFT and sounds,

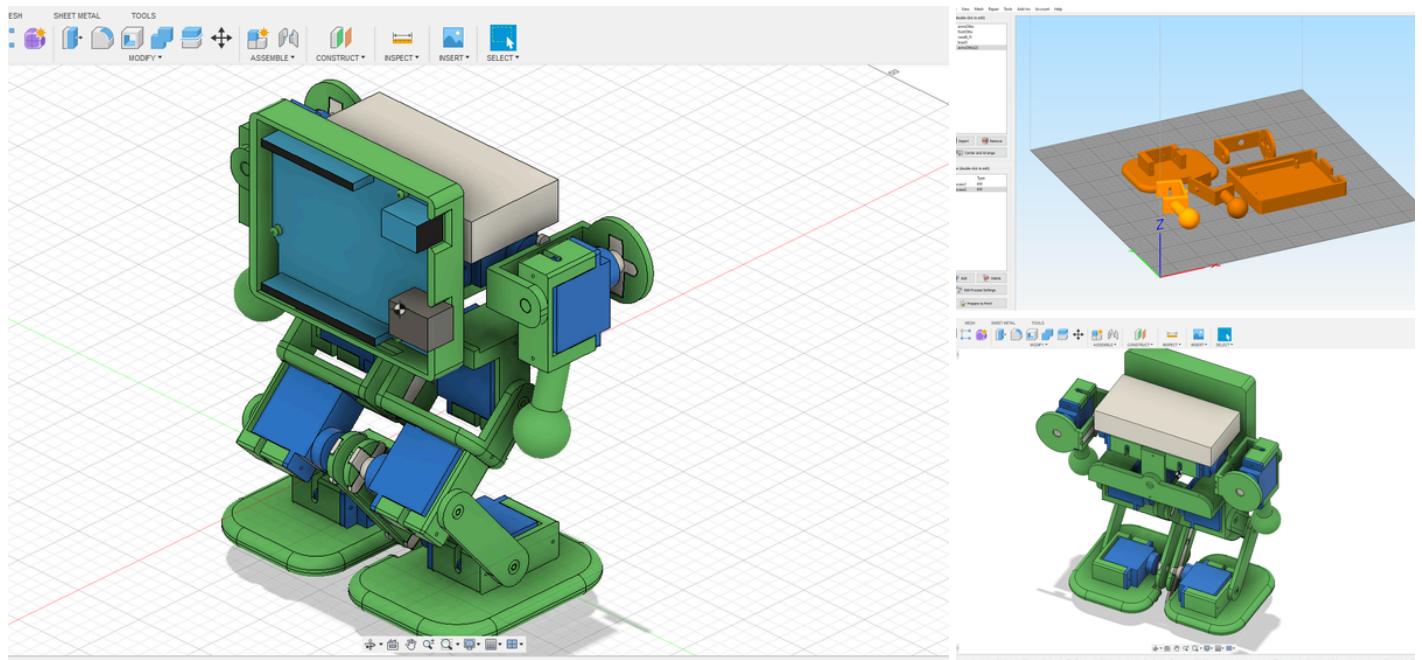
The creation of this robot requires materials that are easy to get, such as arduinos boards, cheap servos and the TFT display designed specifically for arduino uno, let's see what this robot can do

Supplies



- Arduino UNO
- Arduino Nano
- Servo driver PCA9685
- 2 - batteries Li-Ion 1500mAh
- Step Up Module mt3608
- Hc-05 bluettooth module
- 11 - micro servos SG90
- 2.4" TFT LCD Display Shield Touch Panel ILI9341
- 3D print parts
- Soldering iron, wire, double side tape...

Step 1: Print the 3D Files



The first step is to download and print the parts that I have designed, I leave here all the necessary files in STL format, I will also give you the editable file for Fusion 360, with this file you can customize your robot or improve it if you know how to use the fusion 360 software!

STL's

<https://drive.google.com/file/d/1-sLCTwfKZNly46tAgkTpEZaJGwMDmr0t/view?usp=sharing>

Fusion file:

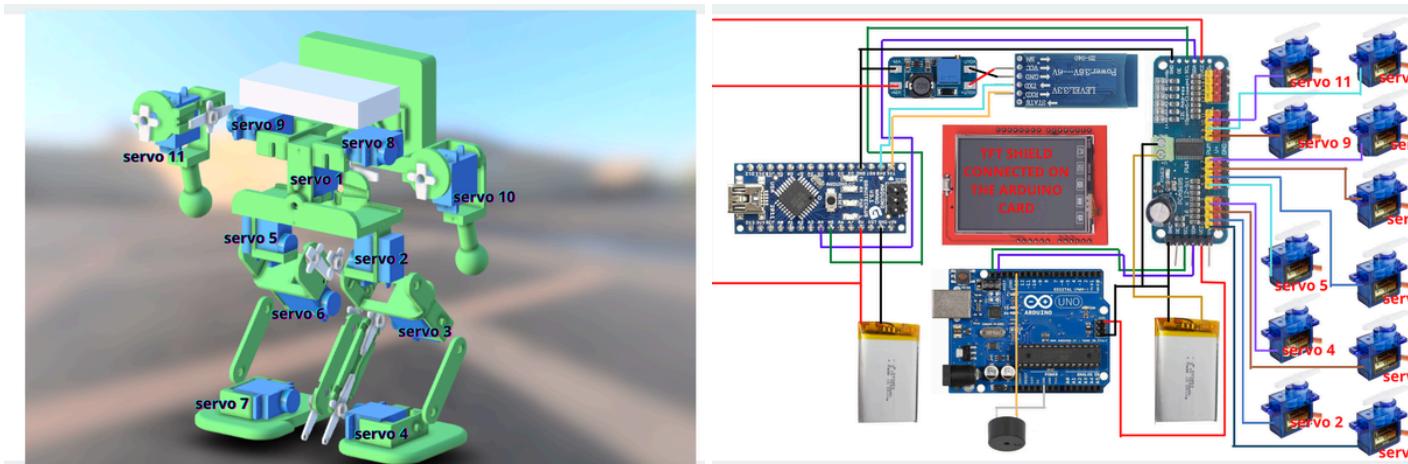
https://drive.google.com/file/d/1WEbP7r_DAYPbivqsT8Qq1ESWr7_RfWHe/view?usp=sharing

Step 2: Modify the TFT Shield



You have to modify the connection of a pin in the TFT module, it is necessary to disconnect the lcd_reset pin from the shield and make a bridge with a cable to the reset pin of the arduino, otherwise the screen will not be able to reset and the information will not be updated in LCD

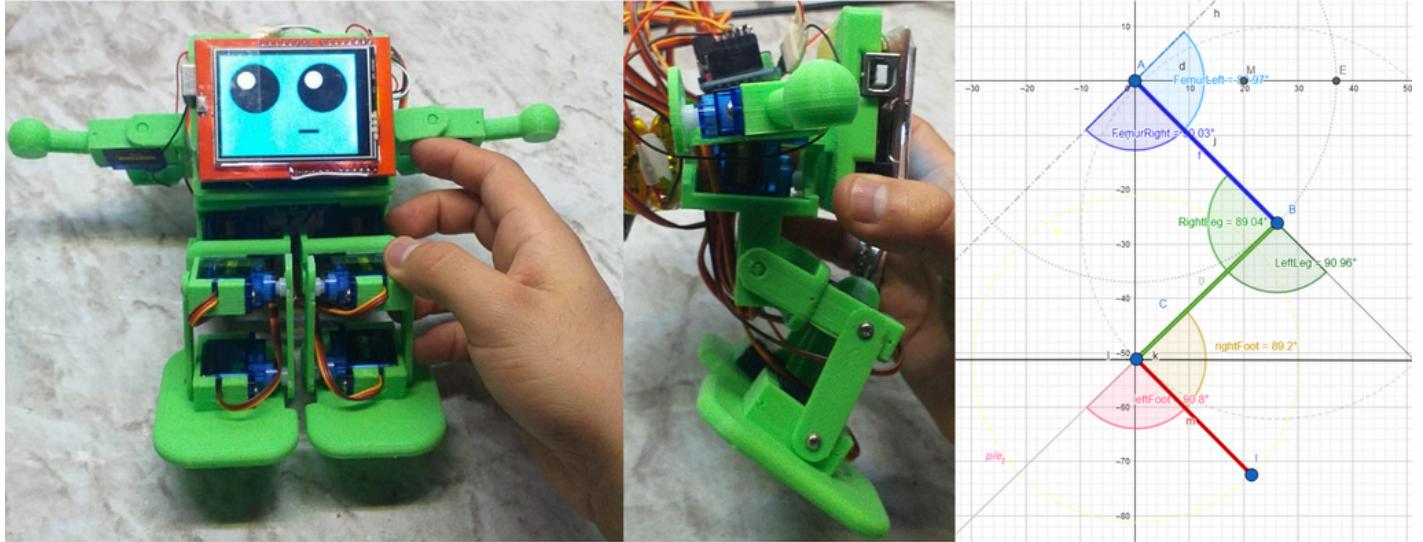
Step 3: Connect Everything!



You have to connect everything, but do not assemble the joints with the shafts of the servomotors , connect all modules and Arduino like the Schematics, you can see the 3d model to download it or use the explosion animation to see it in detail!

<https://a360.co/3bTHBQW>

Step 4: Upload the Home Calibration Code



Now you could upload the arduino code to both cards, but before uploading the final codes you must upload the test code to the arduino nano, be careful, you must upload the arduino nano code with the bluethoot module disconnected, otherwise an error will appear, after uploading the test code you have to assemble the shaft of the servos, the robot must be in this position with the code test, that is to calibrate the positions of the servos, the test code has the position "home" for the robot.

You need this library for the servo driver

<https://github.com/adafruit/Adafruit-PWM-Servo-Driver-Library/blob/master/examples/pwmtest/pwmtest.ino>

Test code:

https://drive.google.com/file/d/13bqThnsI_gnkbV_8nVJOvAqs7p3Nr94A/view?usp=sharing

Step 5: Upload the Code

The image shows two side-by-side Arduino IDE windows. The left window is titled 'test_Ottis | Arduino 1.8.11' and contains the following code:

```
test_Ottis
1
2
3 #include <Wire.h>
4 #include <Adafruit_PWM_Servo_Driver.h>
5
6 Adafruit_PWM_Servo_Driver pwm = Adafruit_PWM_Servo_Driver();
7
8 int pwmAngulo_PI = 90;
9 int pwmAngulo_TI = 90;
10 int pwmAngulo_PiI = 90;
11
12 int pwmAngulo_PD = 90;
13 int pwmAngulo_TD = 90;
14 int pwmAngulo_PiD = 90;
15
16 int ang_cintura = 90;
17 int L_Shldr = 90;
18 int R_Shldr = 90;
19
20 int L_Arm = 90;
21 int R_Arm = 90;
22
23 //-----
24
25 void setup() {
26     Wire.begin();          // Conexión al Bus I2C MAESTRO
27     Serial.begin(115200);
28
29 //pinMode(led,OUTPUT);
30     Serial.println("16 channel PWM test!");
31
32     pwm.begin();
33     pwm.setPWMFreq(60);
34
35 }
36
37 //-----
38
39 void loop() {
40 }
```

The right window is titled 'Ottis_UNO | Arduino 1.8.11' and contains the following code:

```
Ottis_UNO
1 #include <Adafruit_TFTLCD.h>
2 /*
3     S_CONNECTION      S_DISCONNECTION    S_BUTTON_PUSHED
4     S_MODE1           S_MODE2           S_MODE3
5     S_SURPRISE        S_OHOH           S_OHOH2
6     S_CUDDLY          S_SLEEPING       S_HAPPY
7     S_SUPER_HAPPY     S_HAPPY_SHORT    S_SAD
8     S_CONFUSED         S_FART1          S_FART2
9     S_FART3           S_JUMP           20
10
11 */
12 #include <SPFD5408_Adafruit_GFX.h>      // Core graphics library
13 #include <SPFD5408_Adafruit_TFTLCD.h> // Hardware-specific library
14 #include <SPFD5408_TouchScreen.h>
15
16 #define LCD_CS A3 // Chip Select goes to Analog 3
17 #define LCD_CD A2 // Command/Data goes to Analog 2
18 #define LCD_WR A1 // LCD Write goes to Analog 1
19 #define LCD_RD A0 // LCD Read goes to Analog 0
20
21 #define LCD_RESET 2 // Can alternately just connect to Arduino's
22
23 // Assign human-readable names to some common 16-bit color values
24 #define BLACK 0x0000
25 #define BLUE 0x001F
26 #define RED 0xF800
27 #define GREEN 0x07E0
28 #define CYAN 0x07FF
29 #define MAGENTA 0xF81F
30 #define YELLOW 0xFFE0
31 #define WHITE 0xFFFF
32
33 Adafruit_TFTLCD tft(LCD_CS, LCD_CD, LCD_WR, LCD_RD, LCD_RESET);
34 //-----
35 #define BUZZER_PIN 13
36 //-----
37 #include <Wire.h>
38 int valor=0;
```

If the robot is already in the "home" position with the test code, then you can upload the final codes, the code for the Arduino Nano controls the servo motors with the help of the library for the PCA9685 driver, the Arduino UNO code controls the gestures performed with the display and the buzzer, the arduinos communicate via i2c protocol, the Nano receives the commands from an android application via bluethoot and sends data to the UNO to perform the corresponding gestures

libraries (Arduino UNO):

Cute buzzer

sounds:https://drive.google.com/file/d/1hGTMd_5iqw0oo43KZcQ1XjjoreXarMkw/view?usp=sharing

TFT display: <https://github.com/JoaolopesF/SPFD5408>

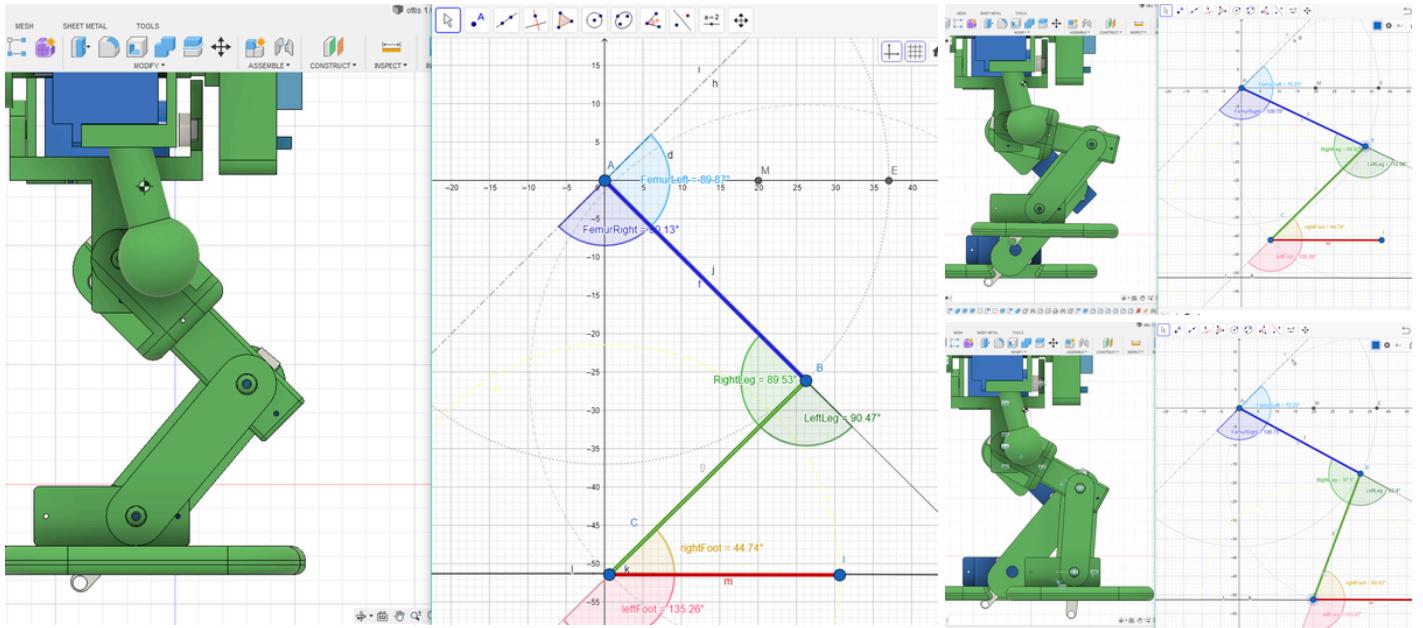
Arduino Nano

code:<https://drive.google.com/file/d/1wFzN4lqWscweqq6aMt2G9hzuLyIDkZWx/view?usp=sharing>

Arduino UNO

code:<https://drive.google.com/file/d/1xcpe3QGNugKbf9jeuONmJVWDFSzwoTBT/view?usp=sharing>

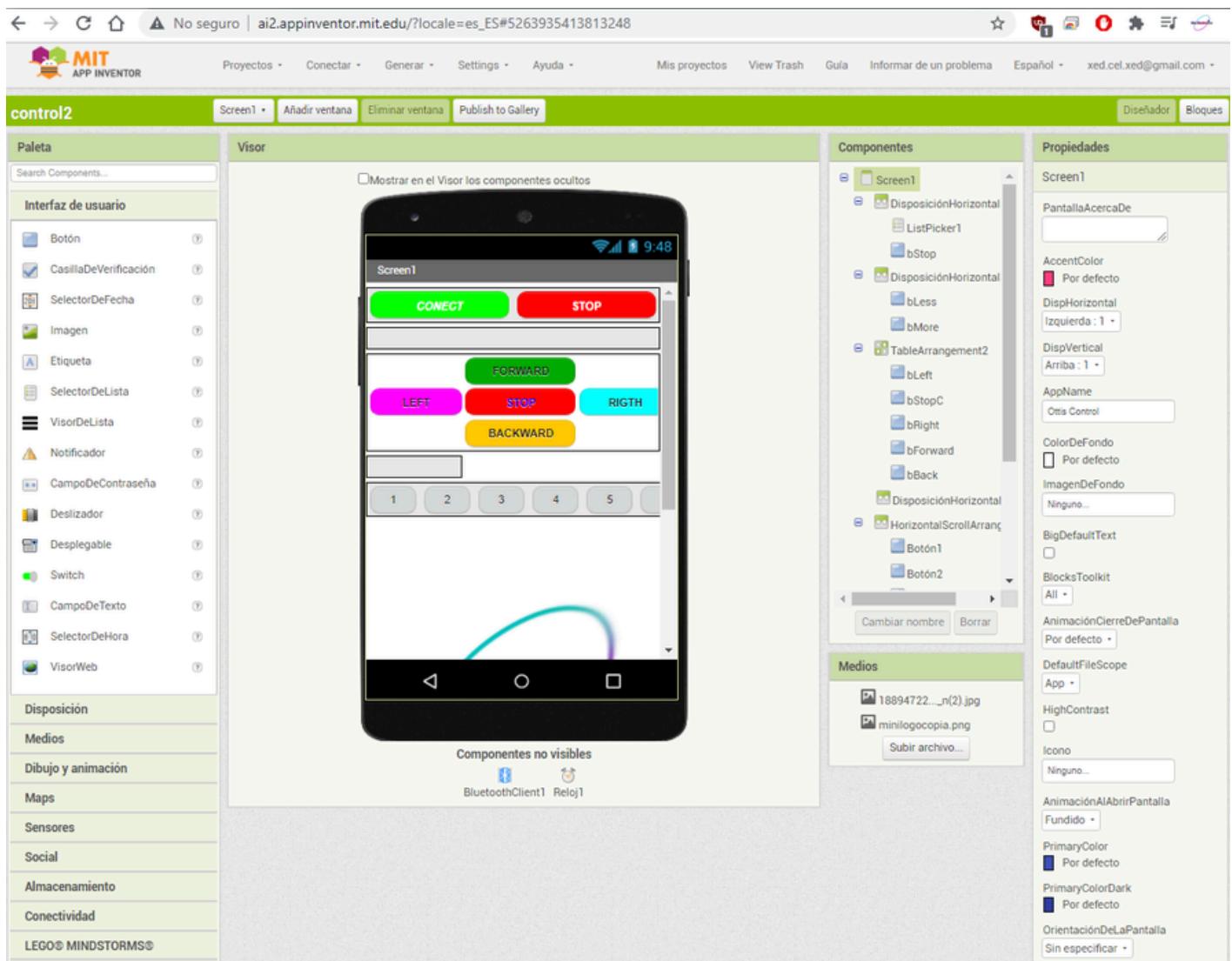
Step 6: See How I Program the Cinematics



You can use this program called Geogebra, which is free, to simulate the angles, create movements and with these angles you could create new sequences of movements by writing them in arduino with some appropriate programming structure

<https://www.geogebra.org/suite/v4mc3xay>

Step 7: Android App



I program an android application with the famous MIT APP INVENTOR platform, you can download the project and edit it (.aia file) or download and install the apk file to your cell phone to use the application

.aia file: https://drive.google.com/file/d/1jCSM6QCcCwRWdgh8FqEp1j8IOUoKr_BV/view?usp=sharing

apk file: https://drive.google.com/file/d/1jCSM6QCcCwRWdgh8FqEp1j8IOUoKr_BV/view?usp=sharing

Step 8: Enjoy!

Now you have everything you need to build your Ottis robot, feel free to improve or customize it with the Fusion 360 Software, in the near future I plan to add more gestures, because there are still many sounds that I do not use and it is possible to draw a large number of figures on the TFT display