

# “Otto DIY Robot”

**Md. Imam Mahdi**

B.E. Student, Electronics and Communication Engineering  
University Institute of Technology, The University of Burdwan

Burdwan, India

Email: mahdibipu@gmail.com

**Abstract**— *An open-source modular robot with an epic personality. It means the hardware is easily discerned so that others can make it, is also Arduino compatible, 3D printable and customizable. Otto was inspired by another robot instructible BoB the BiPed and programmed using code from another open-source biped robot called Zowi. Otto robot walks, dances, makes sounds and avoids obstacles, easy to program and expand or customize, you will learn the logical connection between code and action, and by assembling it, you will understand how its components and electronics work. The motive of making the project cost efficient and user friendly is taken into account and achieved. The project is comprised of components such as an Arduino Nano, Arduino Nano Shield, Ultrasonic Sensor, Servo Motors, and Otto robot body parts.*

**Keywords**— **component:** Otto Robot Body Parts, Arduino Nano, Arduino Nano Shield I/O, USB-A to Mini-USB Cable,

SG90 Micro-servo motor, Ultra-sound sensor HC-SR04, Buzzer, Female/Female jumper wire, Toggle switch, 4 x AA Battery holder AA Battery.

## 1. INTRODUCTION

An open-source modular robot with an epic personality, a robot companion for learning STEM & Creativity.

Otto is an opensource robot; it means the hardware is easily discerned so that others can make it, is also Arduino compatible, 3D printable and customizable and with a social impact mission to create an inclusive environment for all kids.

Otto was inspired by another robot instructible BoB the BiPed and programmed using code from another open-source biped robot called Zowi.

Otto's differences are in the assembled size (12cm x 7cm x12cm), cleaner integration of components and expressions. Using off the shelf and 3D printed parts, simple electronics connections (almost no welding required), and basic coding skills. Otto is design using Autodesk 123D Design initially.

Otto robot walks, dances, makes sounds and avoids obstacles, easy to program and expand or customize, you will learn the logical connection between code and action, and by assembling it, you will understand how its components and electronics work.

## 2. Apps and Online service

- Arduino IDE
- Arduino Web Editor
- Microsoft Visual Studio Code Extension for Arduino

### 3. Review of Otto Robot

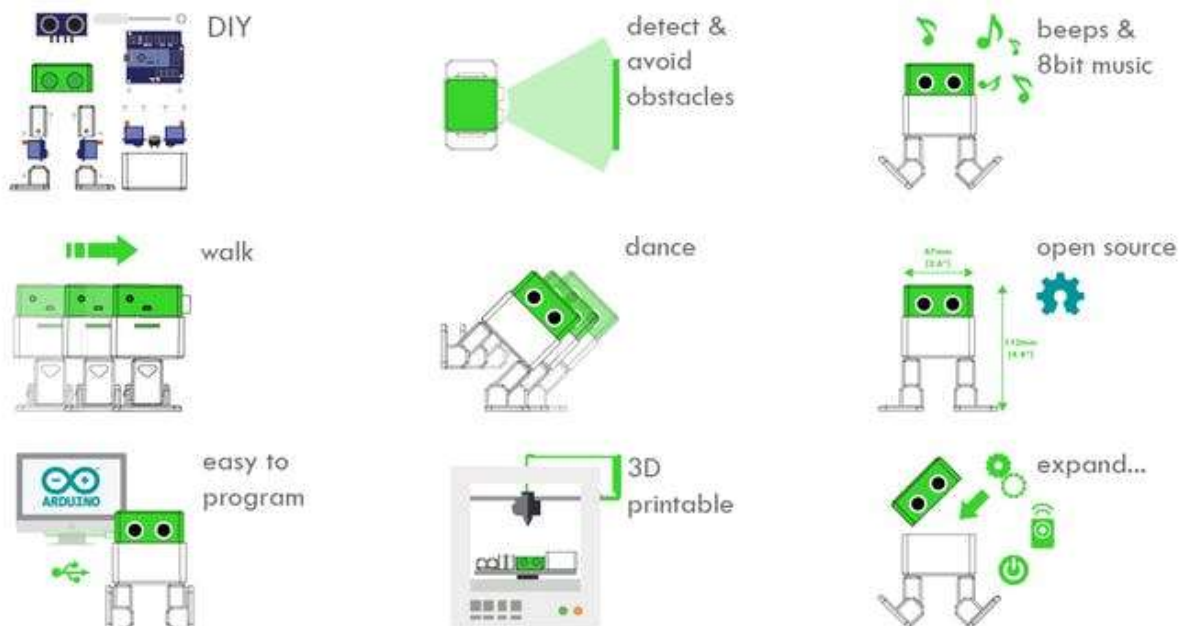
#### # What is Otto:

Otto is an opensource robot; it means the hardware is easily discerned so that others can make it, is also Arduino compatible, 3D printable and customizable.  
An interactive robot that anyone can make!



#### # What can Otto do?

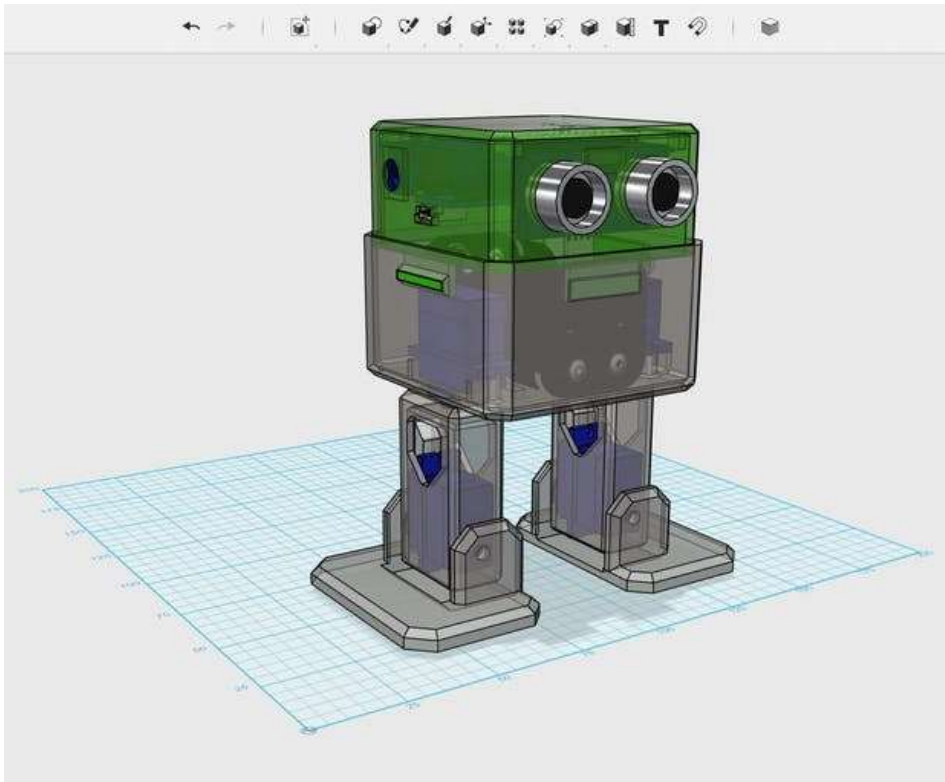
Otto walks, dances, makes sounds and avoids obstacles.



# CC-BY-SA:

Otto's differences are in the assembled size (11cm x 7cm x12cm), cleaner integration of components and expressions. Using off the shelf and 3D printed parts, simple electronics connections (almost no welding required), and basic coding skills, you will be able to build your own cute Otto friend in as little as one hour! or even less

Otto was design using Autodesk 123D Design, now [Thinker CAD](#) software you can modify it for customization or further improvements!



This step by step focuses on how to build the Otto DIY simple version; yes, there are more advanced [bipedal robots like Otto DIY+](#) or Tito.

### 3.1 3D Printing

Only need to 3D print 6 parts in total:

1. 3D printed head.
2. 3D printed body.
3. 3D printed leg x2.
4. 3D printed foot x2

( [download all .stl files](#), ) these files for 3D printing the body parts.

If you do not have a 3D printer you can always got to your local maker space.

If you don't have or is difficult to get the part you can just buy our full kits visit [www.ottodiy.com](http://www.ottodiy.com) .

### 3.2 3D print settings:

Otto is very well designed for 3D printing, so won't give you trouble if you follow this common parameter:

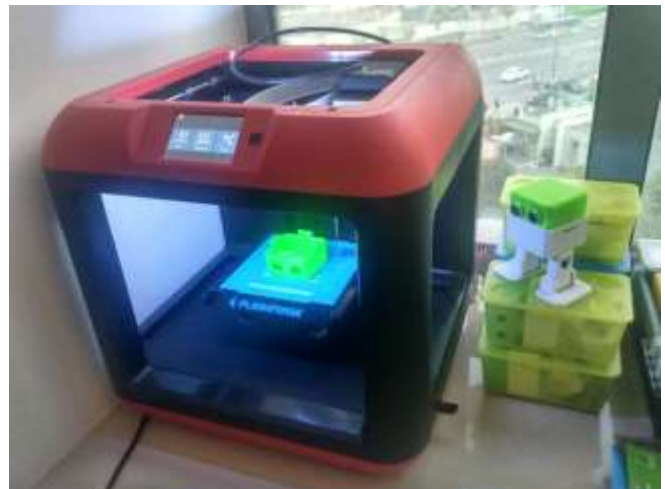
- Recommended to use a FDM 3D printer with PLA material.
- No need supports or rafts.
- Resolution: 0.20mm
- Fill density 20%

For slicing and generating the g code for the machine free slicer software like Cura or in our case Flash-Print that comes with the Flash-Forge Finder 3D printer.

All parts in total use 115gr approx.

equivalent to 14.5m of regular PLA 1.75mm thickness, it should take around 8 hours to

3D prints a full set of parts for one Otto even half that time with a proper 3D printer and optimized tune settings for your machine.



### 3.3 Component Description:

#### # Arduino Nano:

Arduino is an open-source computer hardware and software company, project, and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical and digital world. The project's products are distributed as open-source hardware and software, which are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL), permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form, or as do-it-yourself (DIY) kits.

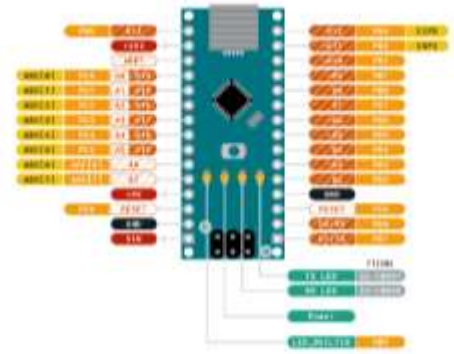
Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards or Breadboards (shields) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.

Arduino Nano is one [type of microcontroller](#) board, and it is designed by Arduino.cc. It can be built with a microcontroller like Atmega328.

This microcontroller is also used in [Arduino UNO](#). It is a small size board and also flexible with a wide variety of applications.

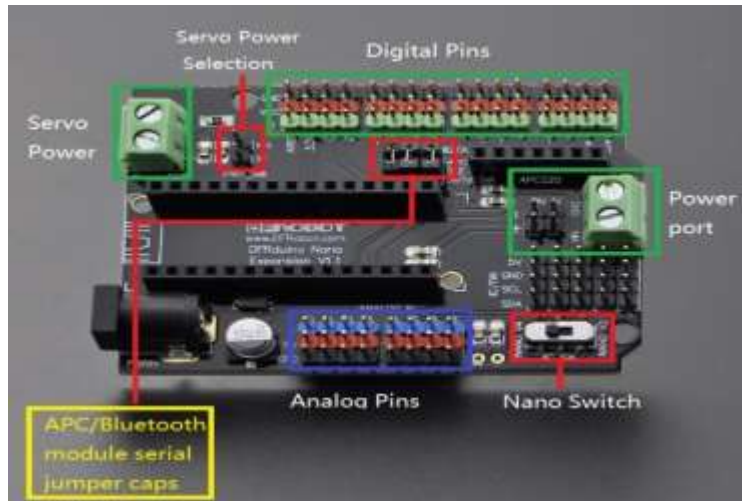
The features of an Arduino nano mainly include the following.

- ATmega328P Microcontroller is from 8-bit AVR family
- Operating voltage is 5V
- Input voltage (Vin) is 7V to 12V
- Input/Output Pins are 22
- Analog i/p pins are 6 from A0 to A5
- Digital pins are 14
- Power consumption is 19 mA
- I/O pins DC Current is 40 mA
- Flash memory is 32 KB
- SRAM is 2 KB
- EEPROM is 1 KB
- CLK speed is 16 MHz
- Weight-7g
- Size of the printed circuit board is 18 X 45mm
- Supports three communications like SPI, IIC, & USART



### # Arduino Nano Shield:

For Arduino Nano IO Expansion Shield is specifically designed to facilitate an easy connection between for Arduino Nano and many other devices.



In essence, it expands the Nano controller to link those devices in a simple and trouble-free manner. It is a perfect companion of Nano breadboard and compatible with both Nano v2.x and v3.x.

This expansion shield provides:

- 14 I/O Pin (servo type with GND, power and signal)
- 8 analog Pin with power output and GND
- 6 PWM Pin
- 1 Servo power input
- 5 I2C expansion Pin
- AREF output
- 3.3V output

### # Arduino Software part:

#### 1. **IDE:**

The Arduino integrated development environment (IDE) is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It originated from the IDE for the languages Processing and Wiring. It includes a code editor with features such as text cutting and pasting, searching and replacing text, automatic indenting, brace matching, and syntax highlighting, and provides simple one-click mechanisms to compile and upload programs to an Arduino board. It also contains a message area, a text console, a toolbar with buttons for common functions and a hierarchy of operation menus. The source code for the IDE is released under the GNU General Public License, version 2.

The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop,



that are compiled and linked with a program stub `main ()` into an executable cyclic executive program with the GNU toolchain, also included with the IDE distribution. The Arduino IDE employs the program *avrduide* to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.

## 2. Sketch:



A program written with the Arduino IDE is called a sketch. Sketches are saved on the development computer as text files with the file extension `.ino`. Arduino Software (IDE) pre-1.0 saved sketches with the extension `.pde`.

## 3. C/C++ program:

A minimal Arduino C/C++ program consist of only two functions:

setup (): This function is called once when a sketch starts after power-up or reset. It is used to initialize variables, input and output pin modes, and other libraries needed in the sketch.

loop (): After `setup ()` has been called, function `loop ()` is executed repeatedly in the main program. It controls the board until the board is powered off or is reset.

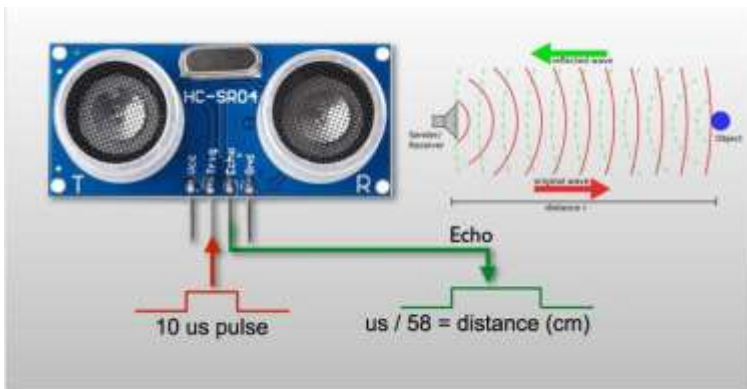
# Ultra-sound sensor HC-SR04:

The HC-SR04 uses non-contact ultrasound sonar to measure the distance to an object, and consists of two ultrasonic transmitters (basically speakers), a receiver, and a control circuit. The transmitters emit a high frequency ultrasonic sound, which bounce off any nearby solid objects, and the receiver listens for any return echo. That echo is then processed by the control circuit to calculate the time difference between the signal being transmitted and received. This time can subsequently be used, along with some clever math, to calculate the distance between the sensor and the reflecting object!

The HC-SR04 sensor works best between 2cm – 400 cm (1" - 13ft) within a 30-degree cone, and is accurate to the nearest 0.3cm

The **features of the HC-SR04 sensor** include the following:

- The [power supply](#) used for this sensor is +5V DC
- Dimension is 45mm x 20mm x 15mm
- Quiescent current used for this sensor is <2mA
- The input pulse width of trigger is 10uS
- Operating current is 15mA
- Measuring angle is 30 degrees
- The distance range is 2cm to 800 cm
- Resolution is 0.3 cm
- Effectual Angle is <15°
- Operating frequency range is 40Hz
- Accuracy





# SG90 Micro-servo motor:

Servo motors (or servos) are self-contained electric devices that rotate or push parts of a machine with great precision.

The simplicity of a servo is among the features that make them so reliable. The heart of a servo is a small direct current (DC) motor, similar to what you might find in an inexpensive toy. These motors run on electricity from a battery and spin at high **RPM** (rotations per minute) but put out very low **torque**. An arrangement of gears takes the high speed of the motor and slows it down while at the same time increasing the torque. The gear design inside the servo case converts the output to a much slower rotation speed but with more torque (big force, little distance).

Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but smaller. You can use any servo code, hardware or library to control these servos.

**Features:**

- Weight: 9 g
- Dimension: 22.2 x 11.8 x 31 mm approx.
- Stall torque: 1.8 kgf.cm
- Operating speed: 0.1 s/60 degree
- Operating voltage: 4.8 V (~5V)
- Dead band width: 10  $\mu$ s
- Temperature range: 0 °C – 55 °C

Position "0" (1.5 ms pulse) is middle, "90" (~2 ms pulse) is all the way to the right. -90" (~1 ms pulse) is middle, "90" (~2 ms pulse) is all the way to the left.



#### # Buzzer:

A buzzer is a small yet efficient component to add sound features to our project/system. It is very small and compact 2-pin structure hence can be easily used on [breadboard](#), Perf Board and even on PCBs which makes this a widely used component in most electronic applications.

#### **Buzzer Features and Specifications:**

- Rated Voltage: 6V DC
- Operating Voltage: 4-8V DC
- Rated current: <30mA
- Sound Type: Continuous Beep
- Resonant Frequency: ~2300 Hz
- Small and neat sealed package
- Breadboard and Perf board friendly



#### # Toggle Switch:

A toggle switch is a type of electrical switch that is actuated by moving a lever back and forth to open or close an electrical circuit. There are two basic types: maintained contact and momentary toggle switches.

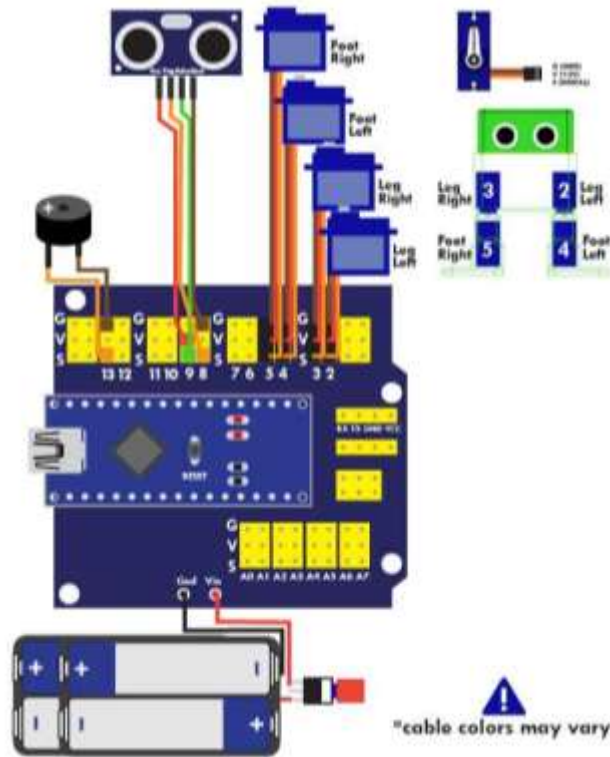
A maintained switch changes its position when actuated and will remain in that position until actuated again, such as an ON/OFF function. A momentary toggle switch is actuated only when someone is operating the switch.

Toggle switches with different actuator styles and amperage ranging from 3 to 20 amps, as well as different load-carrying capabilities and locking combinations.



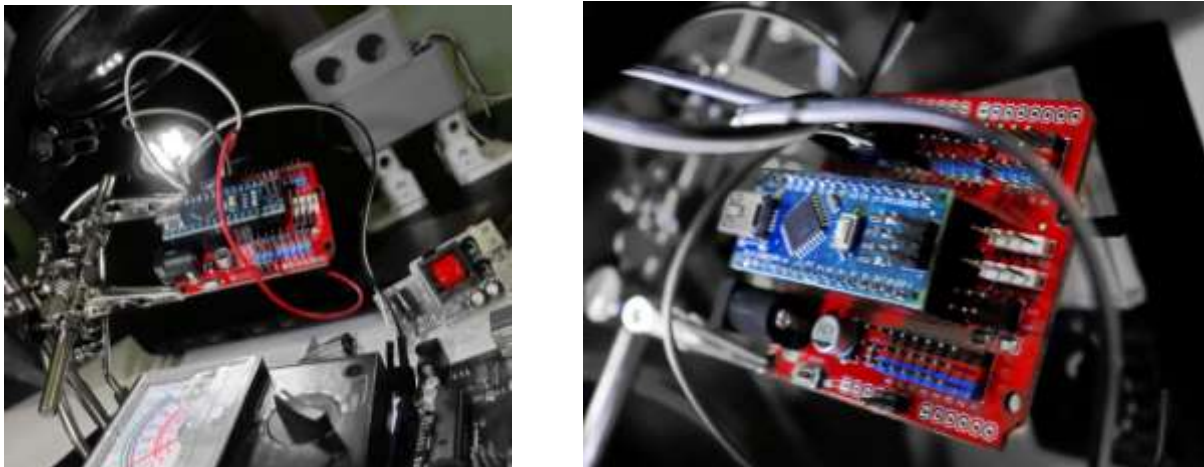
#### 4. Circuit Diagram:

Servos, ultrasound sensor, buzzer, and battery connections, follow the cables. Use the DuPont cables.



**Fig 1:** Schematic diagram of Otto

Arduino Nano attach with Arduino Nano Shield:



**Fig 2:** Arduino Nano attached with Arduino Nano Shield

#### 5. Body & Parts Assembly

First gather all parts and tools.



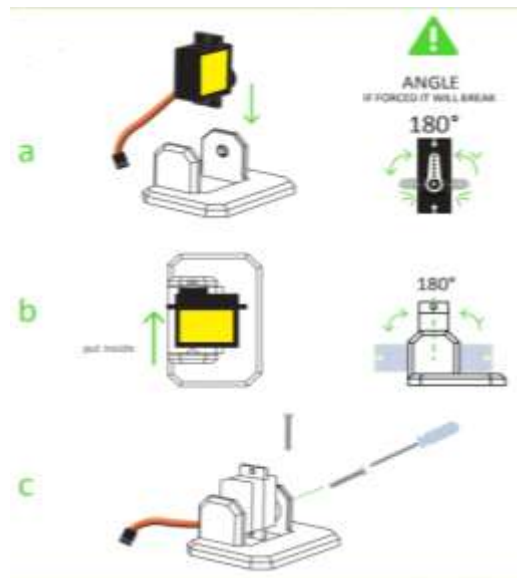
# Foot Servos Assembly:

Put the micro servo inside feet and then push it inside, if is to hard maybe need to clean more the area with a cutter.

Is very important to check that the servo is able to rotate at least 90 degrees to each side.

After checking the movement use only the small screw to fix it.

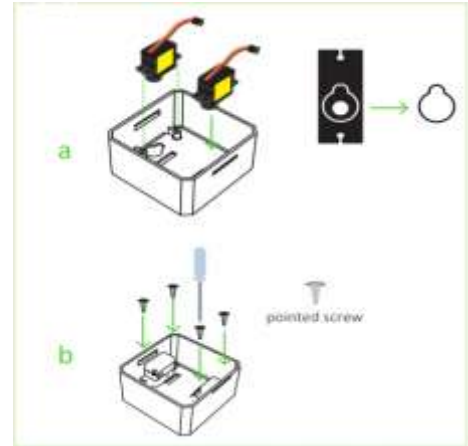
Same process for the other foot.



# Fix Servos to Body:

Take the other 2 micro servos put them in the defined locations in the 3D printed body

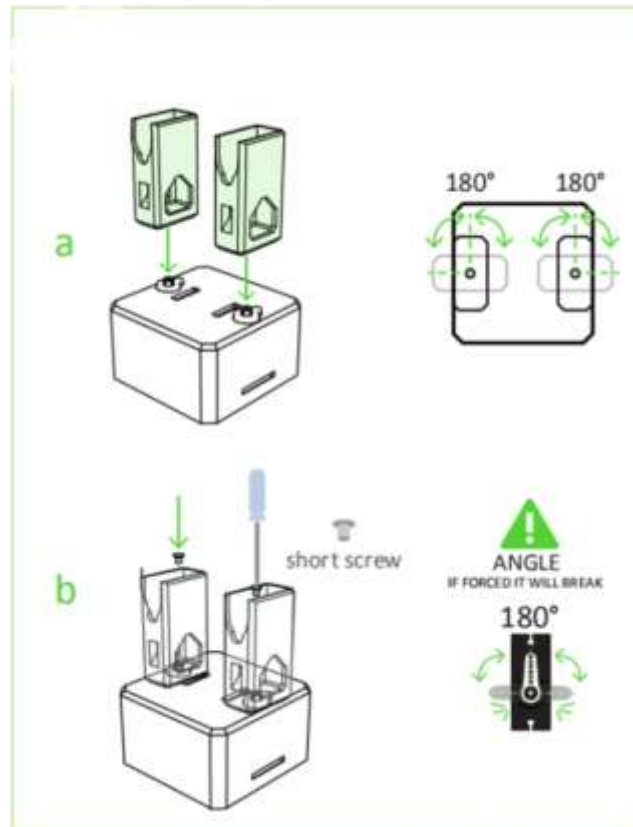
and fix them only with the pointed screws.



# Fix Legs to Body:

Connect the legs to the hub of the micro servo, important like the foot servos you must check the legs are able to rotate 90 degrees each side respect to the body.

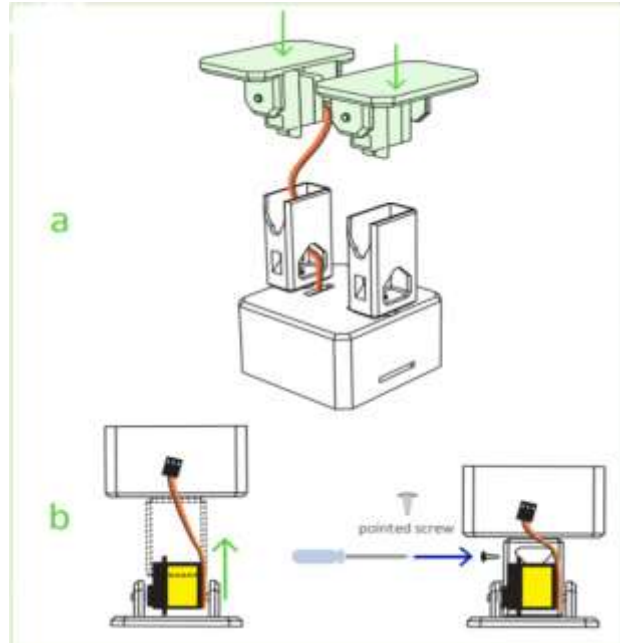
After verifying the alignment fix them using the small screws to the hole inside the leg.



# Fix Foot to Legs:

Taking care of the cables as showed in the illustration you should put the cables inside the slots of the body passing through the hole of the legs.

Once they are in right position use the pointed screws to fix them from the back.

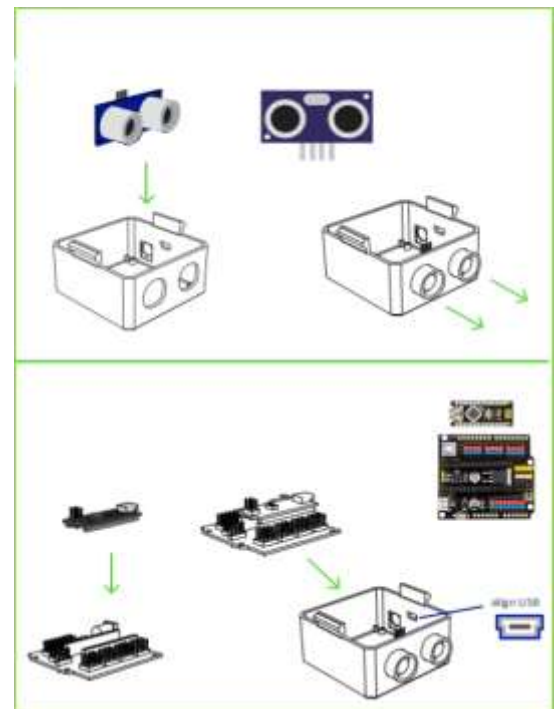


# Head Assembly:

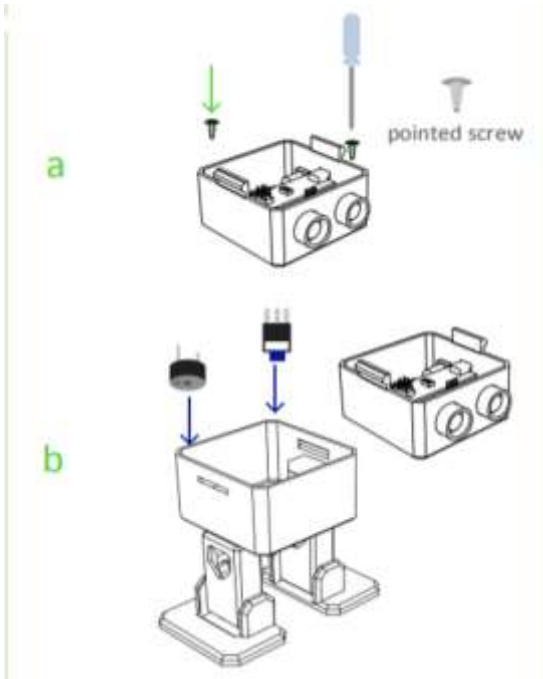
Start from the ultrasound sensor is important to pull out the eyes to the limit.

After putting the Arduino nano in the shield, optionally you can weld the battery holder positive cable to Vin in the board and negative to any GND.

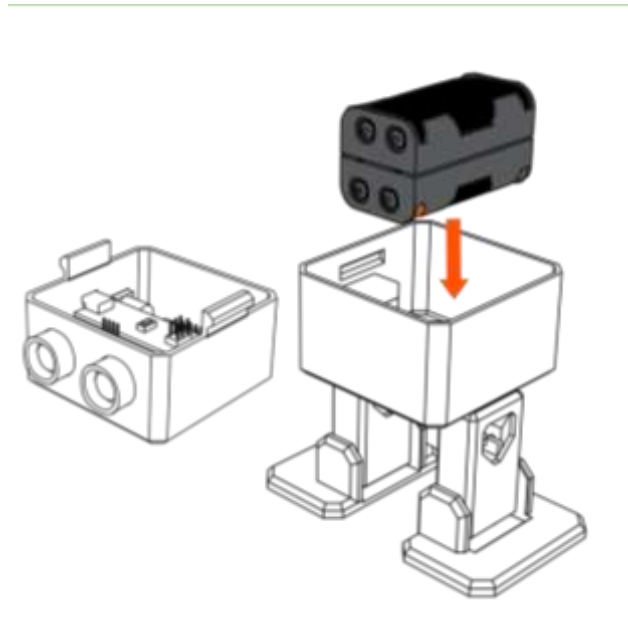
Insert diagonally the both boards together facing the USB connector to the hole in the 3D printed head, then use the last 2-pointed screws to fix it.



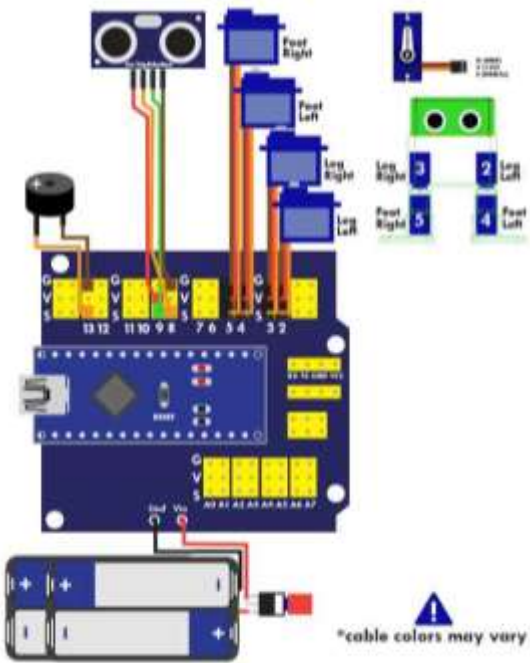




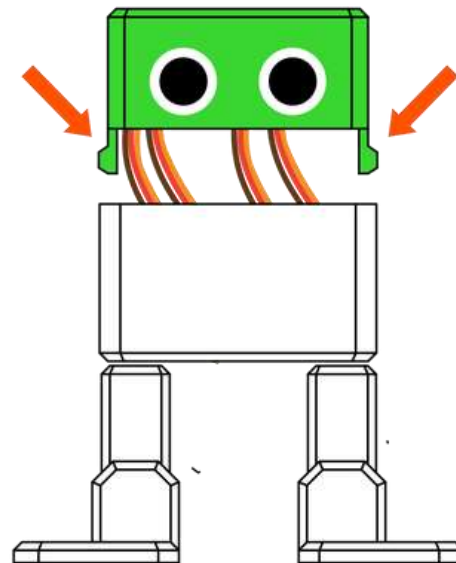
**Fig 3:** Switch & Buzzer



**Fig 4:** Battery Connection



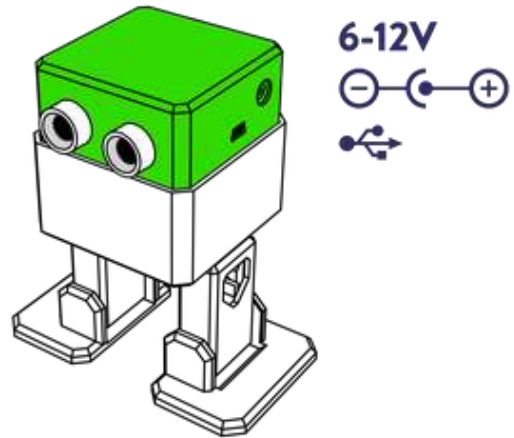
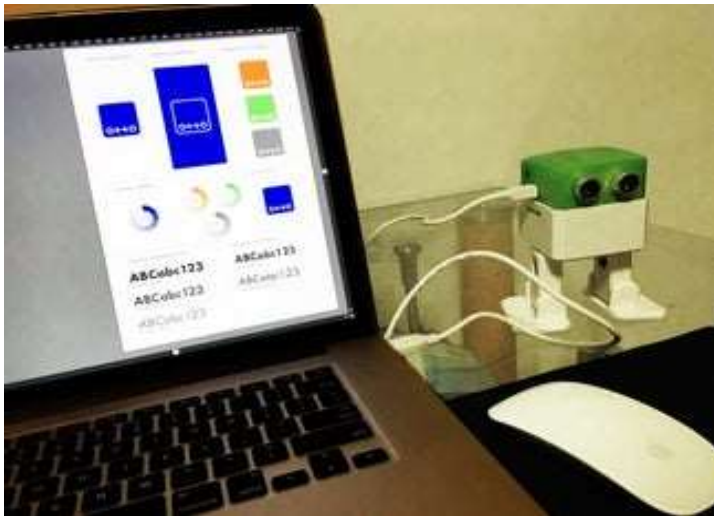
**Fig 5:** Full Circuit Diagram



**Fig 6:** Full Otto robot assembly

# Electric Connection:

By using on the switch and also you can use a USB cable to give Otto robot power.



**Fig 7:** Electric Connection

6. Prototype:

Below pictures shows the prototype of the system:



**Fig 8:** Prototype System

### 7. Application:

Otto is open-source user friendly project. Also, Arduino compatible, 3D printable and customizable, the perfect opportunity to build and have your very first robot, learn robotics, STEM and have fun, you will learn the logical connection between code and action, and by assembling it, you will understand how its components and electronics work.

Otto robot walks, dances, makes sounds and avoids obstacles, easy to program and expand or customize.

### 8. Result:

With the procedures mentioned, the implementation of the project “Otto DIY using the Arduino Nano Microcontroller” is successfully completed and implemented.

The project is cost efficient and user friendly because it can be used by anyone.

### 9. Limitations and Problems Encountered:

This project encountered certain difficulties that are described below.

Initially, when all the connections were done, the major problem was the connection between the Servo Motors and the Arduino Nano. If the servos are connected with wrong pin, then it won't move correctly.

A second problem was also encountered with installing Arduino Nano with Arduino Nano Shield and Ultra-Sound sensor in Head part.

### 10. Future work and Upgrade:

The current project presented the implementation of an inexpensive Otto DIY mini robot, within the framework of assistive technology. The system implementation is based on the Arduino microcontroller.

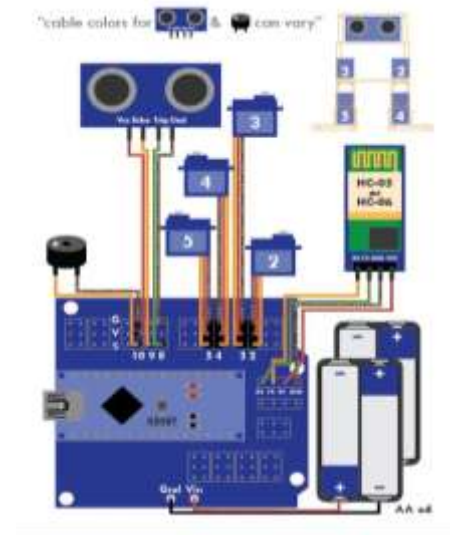


We can modify Otto parts for further improvements or create new accessories for new features. We can give them a new shape and face, by using 3D model.

We can also modify it by using Bluetooth module, to make it wireless. We can control it by using Smart phone.

It is same as the Otto robot just one Bluetooth module is added and the code is different for this.

The Circuit diagram is given here:



**Fig 22: Bluetooth Otto**

## 11. **Conclusion:**

The implementation of this project overall is successful. The motive of making the project cost efficient and user friendly is taken into account and achieved.

The project is comprised of components such as an Arduino Nano, Arduino Nano Shield, Ultrasonic Sensor, Servo Motors, and Otto robot body parts.

Furthermore, with the discussions and objectives presented, it can be concluded that the objectives of the project have been achieved. With this project we are familiar with Arduino nano, coding, and making 3D model.

## 12. **Reference:**

- Wikifactory ( <https://wikifactory.com/+OttoDIY/otto-diy> )
- <https://www.ottodiy.com/>
- Acrobotic from YouTube <https://www.youtube.com/watch?v=57SubBrt-HY>



## **Authors**

**Md Imam Mahdi**, B.E Student

**Email:** mahdibipu@gmail.com