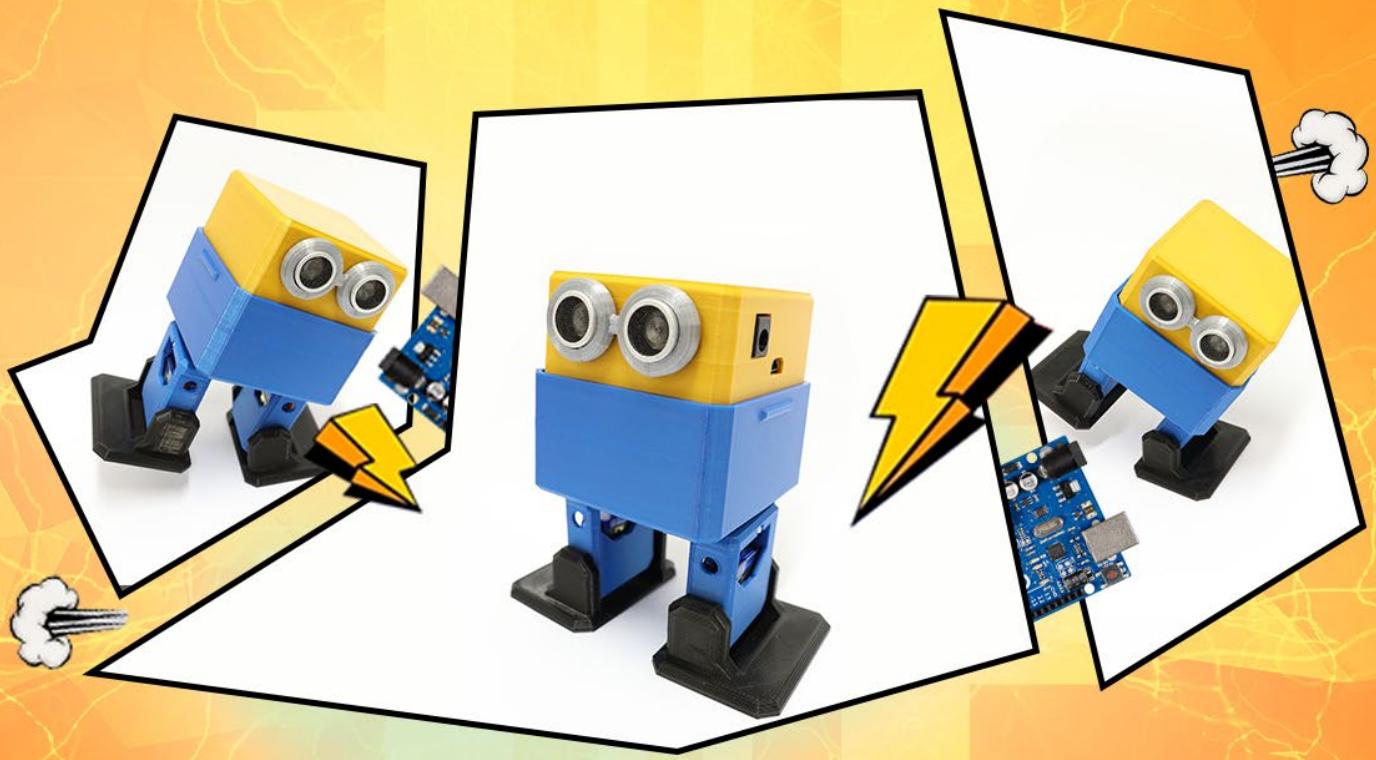




ELETROGATE

APOSTILA

# KIT DIY ROBÔ OTTO



A P O S T I L A K I T  
**ROBÔ OTTO**

V1.1 – Outubro/2024

Olá, **Maker!**

Primeiramente, gostaríamos de parabenizá-lo(a) pelo interesse neste material.

Esta apostila exclusiva servirá como material de apoio teórico e prático para o [Kit Robô Otto](#), mas, mesmo que você ainda não tenha adquirido o seu Kit, ela poderá te ajudar a conhecer um pouco mais sobre o Universo Maker, de uma maneira didática e objetiva.

Vale ressaltar que, embora esta apostila proponha apenas um exemplo de código, o Arduino permite infinitas possibilidades para você manipular o robô, implementar novos movimentos e muito mais, utilizando apenas os componentes presentes nesse kit e desenvolvendo novas funcionalidades somente via programação.

Caso ainda não tenha adquirido seu Kit, acesse o nosso site e garanta o seu.

Desejamos bons estudos!

## Sumário

|                                 |    |
|---------------------------------|----|
| 1. LISTA DE COMPONENTES.....    | 4  |
| 2. CONFIGURAÇÕES INICIAIS ..... | 5  |
| 3. INSTALAÇÃO DOS DRIVERS.....  | 8  |
| 4. MONTAGEM DO ROBÔ .....       | 11 |
| 5. CONSIDERAÇÕES FINAIS .....   | 39 |

## 1. L I S T A D E C O M P O N E N T E S



O Kit Robô Otto é composto pelos seguintes itens:

- Arduino Nano + Cabo
- Shield de Expansão para Arduino Nano 3.0
- Módulo Sensor de Distância Ultrassônico HC-SR04
- 4 Micro Servos 9g SG90 (com parafusos e acessórios)
- Buzzer Ativo 5V
- Mini Chave Gangorra
- Módulo Regulador de Tensão Step Down Buck DC-DC Mini 360 3A
- Jumpers – Fêmea/Fêmea – 20 unidades de 10cm
- 2 Capacitores Eletrolíticos de 1000uF/16V
- Partes impressas, sendo: Cabeça, Corpo, Pernas (2x) e Pés (Direito e Esquerdo).

Além dos componentes presentes no kit, você precisará de:

- Bateria 9V Alcalina
- Chave Phillips ou Fenda pequenas

- Alicate de corte
- Cola quente ou fita dupla face tipo VHB
- Computador com acesso à internet

## 2. CONFIGURAÇÕES INICIAIS

Antes de iniciarmos a montagem, vamos fazer as instalações dos programas e drivers necessários em nosso computador para que possamos programar o Arduino.

A Arduino IDE é a plataforma oficial para programação do Arduino. Vamos começar sua instalação fazendo o download do instalador, que pode ser encontrado no [site oficial – clicando aqui](#).

Recomendamos sempre que você instale a versão estável mais recente disponível para seu sistema operacional. Como estamos utilizando o Windows 10 em nosso computador, vamos selecionar a opção **Windows Win 10 and newer, 64 bits**:

### Downloads



The screenshot shows the Arduino IDE 2.3.3 download page. On the left, there's a logo and a brief description of the features of the new release. Below that, there's a link to the documentation and a note about nightly builds. At the bottom, there's a link to the source code on GitHub. On the right, there's a sidebar titled "DOWNLOAD OPTIONS" with links for Windows (highlighted in red), Windows MSI installer, Windows ZIP file, Linux (AppImage 64 bits X86-64), Linux (ZIP file 64 bits X86-64), macOS (Intel, 10.15: "Catalina" or newer, 64 bits), and macOS (Apple Silicon, 11: "Big Sur" or newer, 64 bits). There's also a link to the Release Notes.

A tela seguinte nos propõe fazer uma contribuição para a plataforma, pode clicar em “Just Download” – Aqui é de suma importância que seu navegador não traduza a página, pois essa opção não é exibida quando traduzida.

### Download Arduino IDE & support its progress

Since the 1.x release in March 2015, the Arduino IDE has been downloaded **87.253.218** times — impressive! Help its development with a donation.

\$3

\$5

\$10

\$25

\$50

Other

**CONTRIBUTE AND DOWNLOAD**

or

**JUST DOWNLOAD**

Na tela seguinte, clique novamente em “Just Download”.

### Stay in the Loop: Join Our Newsletter!

As a beginner or advanced user, you can find inspiring projects and learn about cutting-edge Arduino products through our **weekly newsletter!**

email \*

I confirm to have read the [Privacy Policy](#) and to accept the [Terms of Service](#) \*

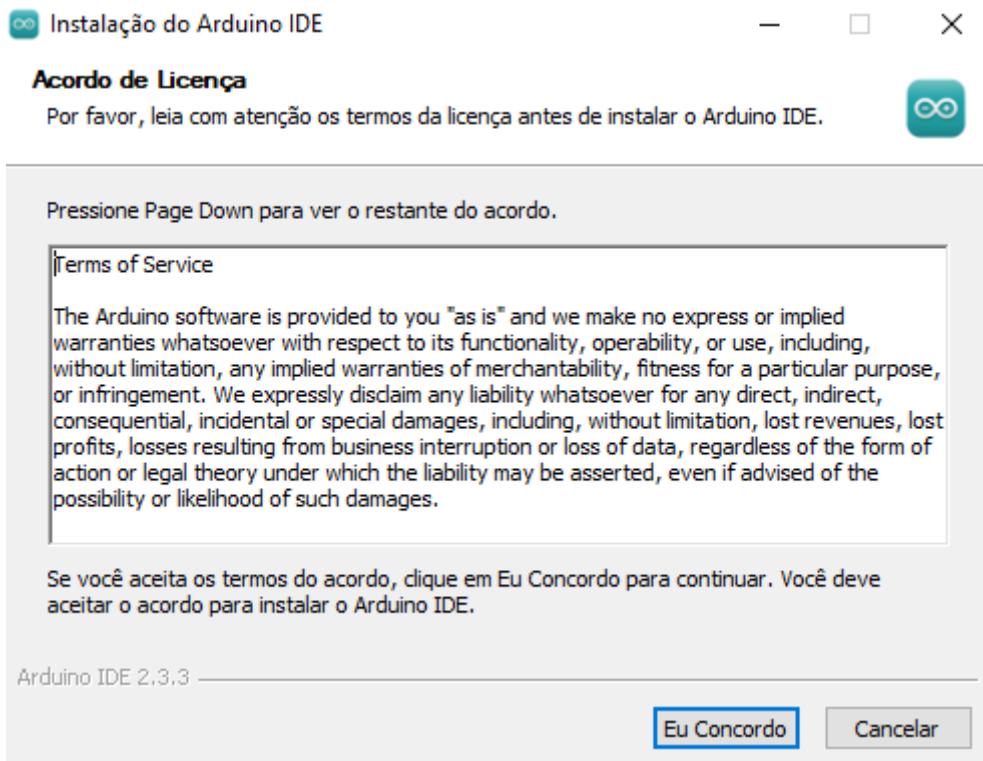
I would like to receive emails about special deals and commercial offers from Arduino.

**SUBSCRIBE & DOWNLOAD**

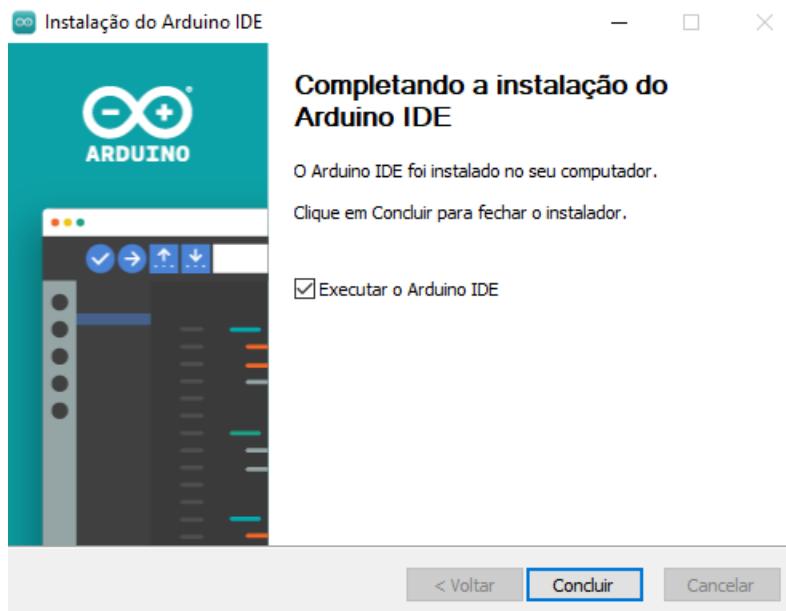
or

**JUST DOWNLOAD**

Após esses passos, o download irá iniciar automaticamente. Quando terminar, execute o arquivo baixado e clique em “Eu concordo” na janela exibida.



Nas telas seguintes, basta clicar em “Próximo” e “Instalar”. Ao término, a tela abaixo será exibida e você poderá fechar o instalador desmarcando a caixinha “Executar o Arduino IDE” (não vamos abri-la agora) e clicando em concluir.



### 3. INSTALAÇÃO DOS DRIVERS

O Arduino Nano utiliza o chip CH340C, um conversor USB-Serial genérico. Ele é responsável por fazer o Arduino se comunicar com o computador, porém a Arduino IDE não oferece suporte nativo, logo precisamos instalá-lo manualmente.

Para isso, acesse o [site oficial – clicando aqui](#) e faça o download conforme seu sistema operacional.



The CH340 chip is used by a number of Arduino compatible boards to provide USB connectivity, you may need to install a driver, don't panic, it's easier than falling off a log, and much less painful.

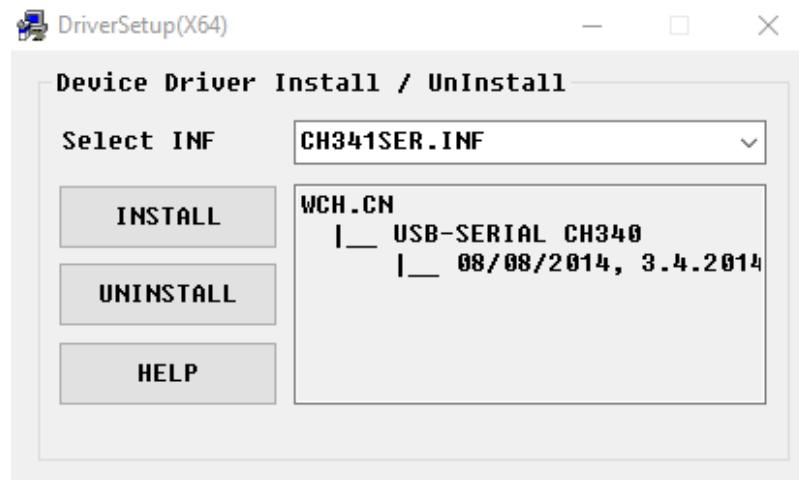
#### Windows

(Manufacturer's Chinese Info Link)

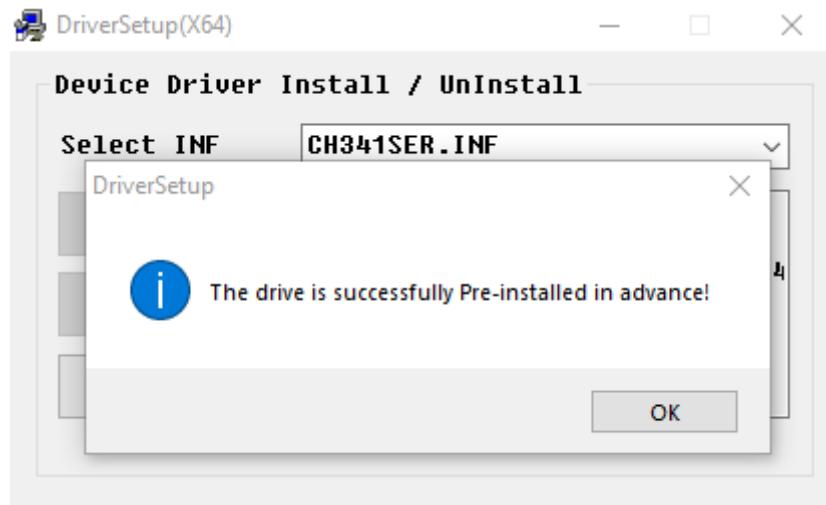
- Download the [Windows CH340 Driver](#)
- [Unzip the file](#)
- Run the installer which you unzipped
- In the Arduino IDE when the CH340 is connected you will see a COM Port in the Tools > Serial Port menu, the COM number for your device may vary depending on your system.

Em nosso caso, vamos fazer a instalação da versão para **Windows**, mas também é possível instalá-lo no MacOS e no Linux. Clique no link referente ao seu sistema operacional e aguarde o download ser concluído.

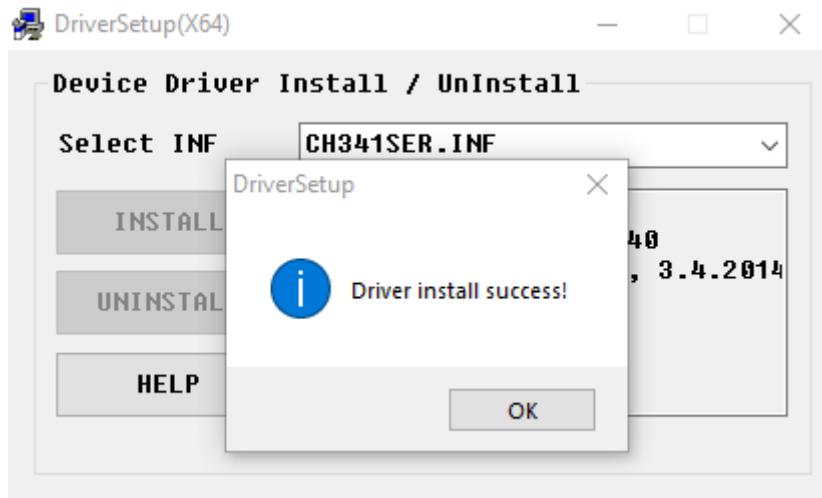
O arquivo baixado será em formato compactado (.zip). Abra-o e dê um duplo clique no arquivo *CH34x\_Install\_Windows\_v3\_4.EXE*, para executar o instalador. Após, a janela abaixo será exibida e você só precisa clicar em **Install**, para iniciar a instalação.



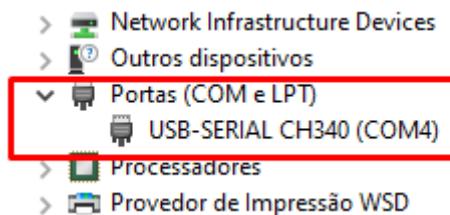
Quando o processo de instalação terminar, a mensagem abaixo será exibida:



Clique em OK e conecte o Arduino ao computador utilizando o cabo USB que acompanha o kit. Em seguida, clique novamente em **Install** e aguarde o processo terminar, sendo exibida a seguinte mensagem:



Com o driver instalado, vamos apenas confirmar se a placa está sendo reconhecida corretamente pelo computador. Para isso, abra o Gerenciador de Dispositivos do Windows (basta abrir o menu iniciar e pesquisar por “gerenciador de dispositivos”). Nele, procure o item “Portas COM e LPT” e veja se a placa está sendo exibida como na imagem abaixo:



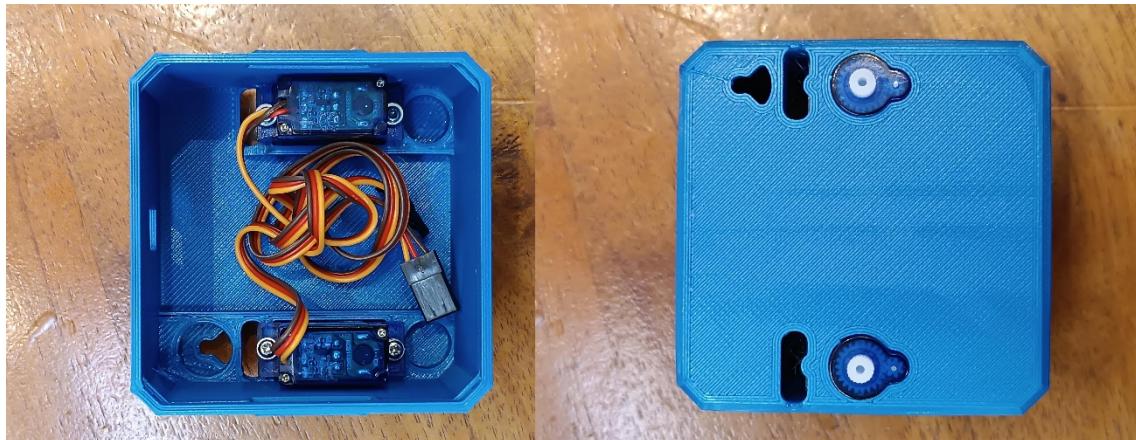
A partir desse ponto, já temos todos os recursos instalados para programar o Otto, restando apenas configurar a Arduino IDE – processo que será mostrado mais adiante. A partir de agora, vamos mostrar o passo a passo de montagem do robô.

#### 4. MONTAGEM DO ROBÔ

Para começar a montagem do Otto, vamos separar os seguintes itens:

- Corpo
- 2 servos e seus acessórios.

Com eles em mãos, vamos iniciar a montagem do corpo, posicionando os servos responsáveis por movimentar as pernas do robozinho. Para montá-los, comece posicionando os servos na parte interna da peça, como pode ser visto nas imagens a seguir:



Note que os servos são presos por 2 parafusos cada. Esses parafusos você encontra no saquinho que acompanha cada servo – nele você encontra três parafusos, onde 2 são para fixação do servo (parafusos maiores) e um para fixação do braço no eixo (parafuso menor), utilize os maiores nessa etapa.

Com essa etapa finalizada, vamos iniciar a preparação das perninhas do robô. Para isso, vamos utilizar dois braços (as peças branquinhas) em formato de cruzeta, os dois parafusos pequenos que restaram da etapa anterior e outros dois de fixação, que você pode pegar do kit de outro servo.

O primeiro passo é diminuir o tamanho das pontas do braço, para que ele encaixe perfeitamente na parte interna da perna. O tamanho final é próximo ao da peça mais à direita na imagem a seguir:



Após encaixado, a perna deve ficar como na imagem a seguir:



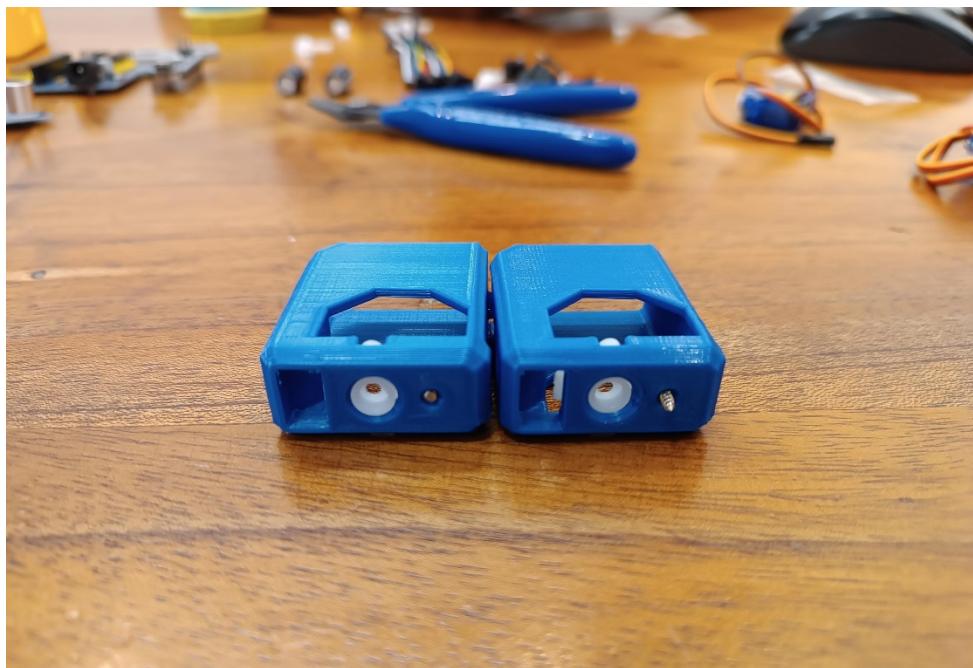
Repita o mesmo processo para a segunda perna:



Feito isso, precisamos prender o bracinho na peça impressa, para que ele não caia quando encaixarmos no servo. Você utilizará, para isso, um dos parafusos de fixação pego de um dos servos que ainda não foram usados.

Você deverá se guiar pelo furo presente no fundo da peça impressa, parafusando de dentro para fora. Corte o excesso com o alicate, deixando o conjunto como a peça mais à esquerda na imagem abaixo:

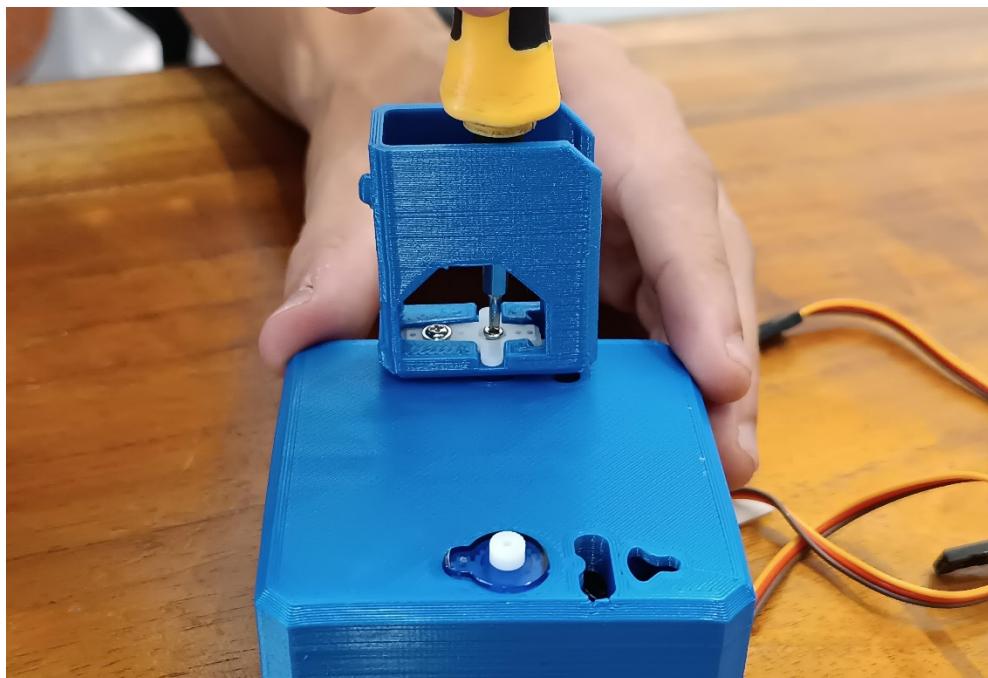
A P O S T I L A K I T  
**ROBÔ OTTO**



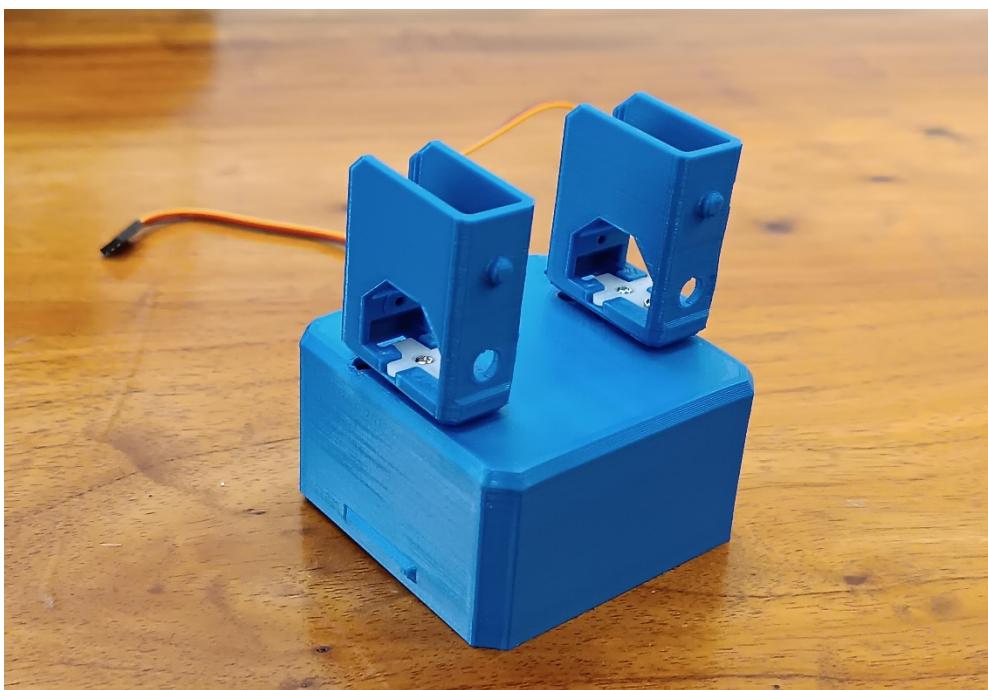
O próximo passo será encaixar as perninhas no corpo. A posição correta pode ser vista na imagem a seguir:



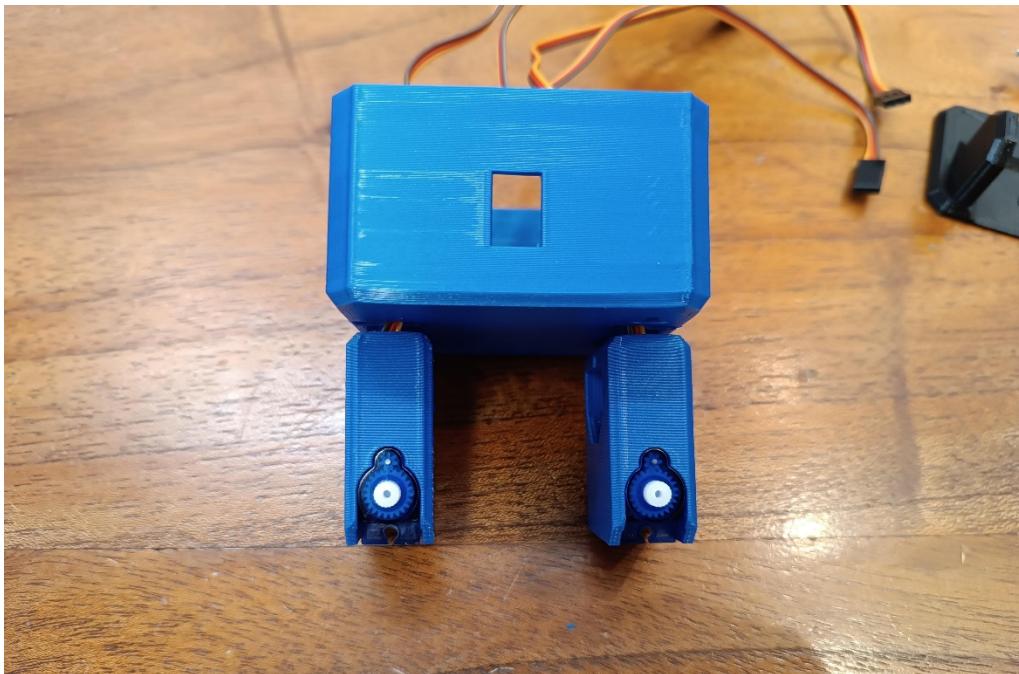
Encaixe a peça impressa no eixo do servo o mais reto possível e parafuse-o usando os parafusos pequenos que sobraram:



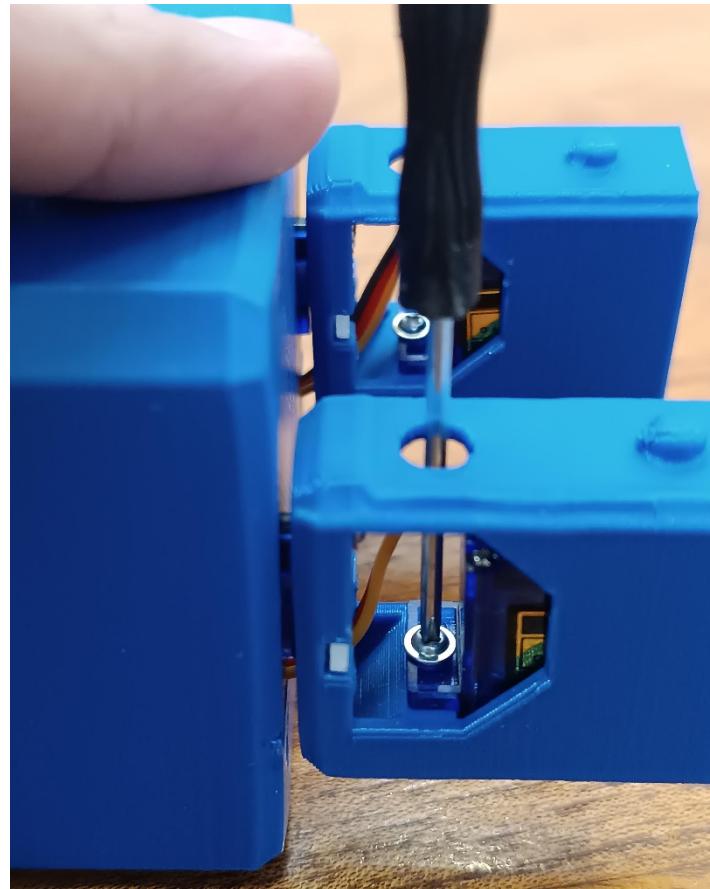
Ao término dessa etapa, você terá o corpo e as pernas montados da seguinte forma:



Concluída essa etapa, vamos à montagem dos pés. Vamos começar posicionando os servos responsáveis pela movimentação dessa parte e são posicionados na parte de dentro das pernas. Faça o encaixe conforme ilustrado na imagem abaixo:



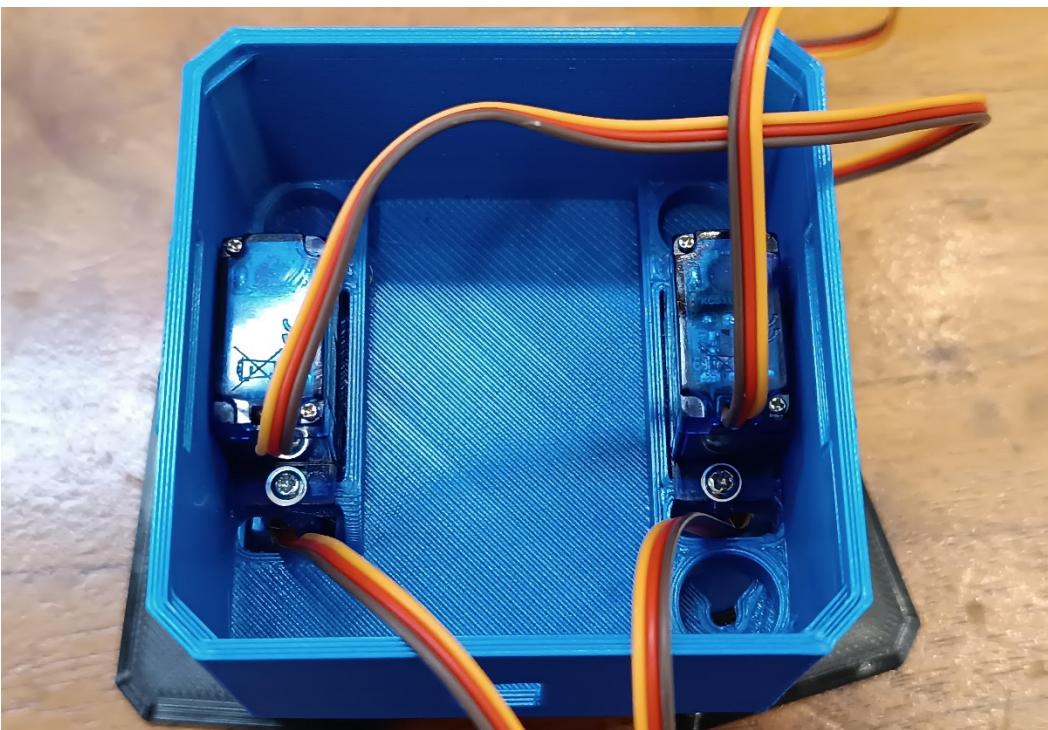
Com os servos no lugar, parafuse-os com os últimos dois parafusos de fixação que restaram, utilizando o furo na parte frontal da peça para encaixar a chave, como mostrado na imagem seguinte (repita esse processo para os dois lados):



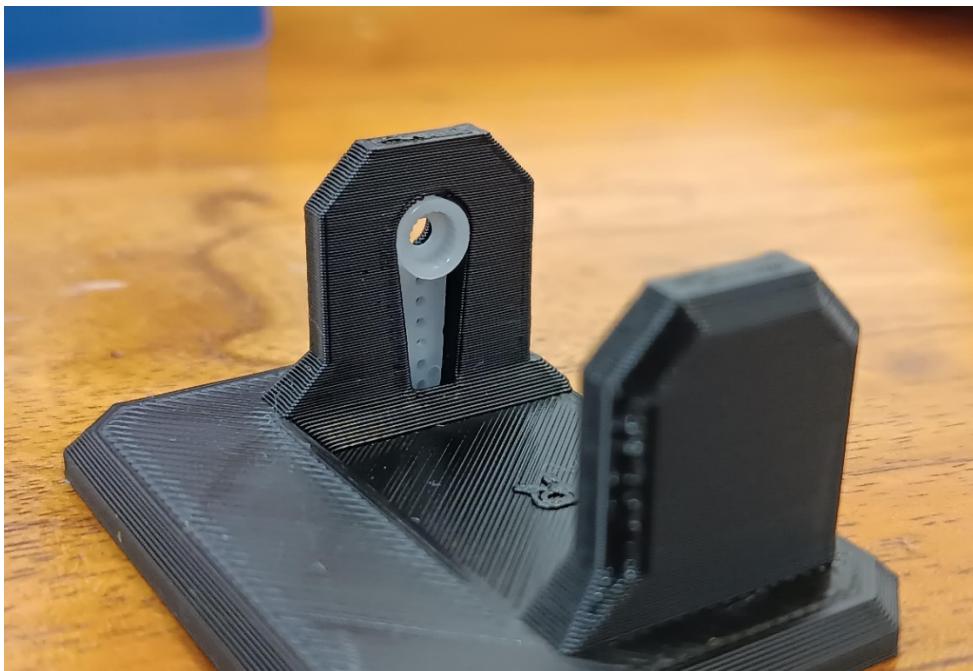
\*\*\*\*\*

**IMPORTANTE:** Os fios dos servos que acabamos de posicionar devem ser passados para dentro da parte superior, usando a fenda presente no fundo de cada uma das pernas que se alinham com os furos presentes no fundo do corpo.

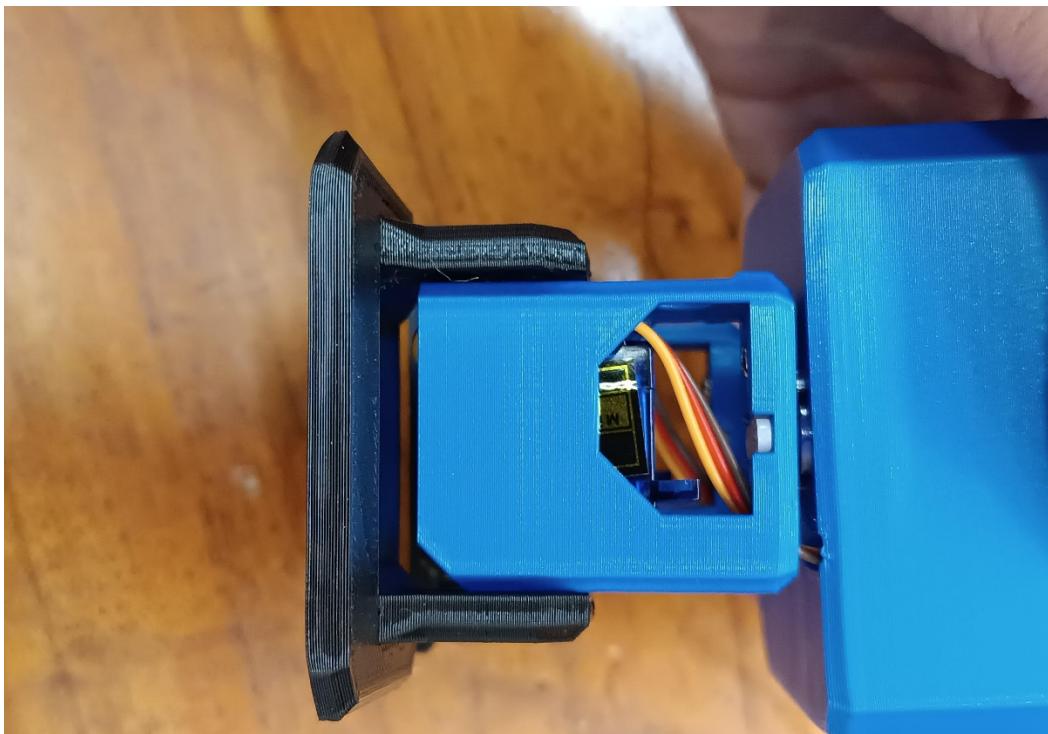
\*\*\*\*\*



Agora, vamos iniciar a montagem dos pés. Eles utilizam somente os dois bracinhos pequenos, que acompanham o kit. Você deve posicioná-los na parte de dentro de cada pé dessa forma:



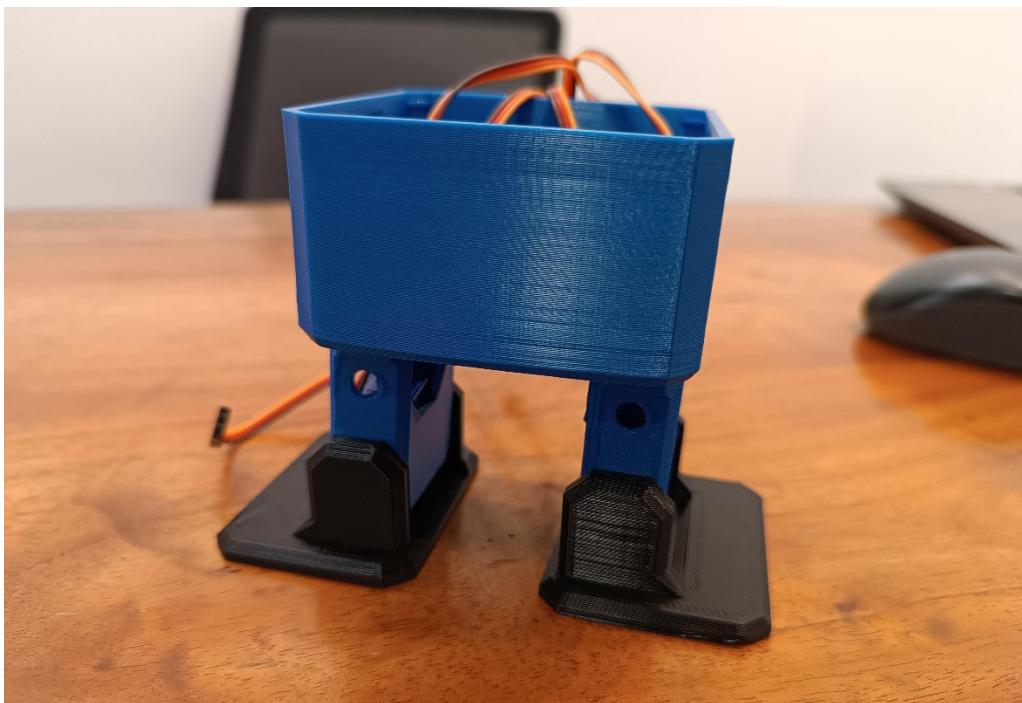
A próxima etapa será encaixar os pés nas pernas. Esse processo exige um pouco de cautela, pois precisamos forçar um pouco as laterais do pé para que ele encaixe tanto no eixo do servo quanto na trava presente na perna, ficando dessa forma:



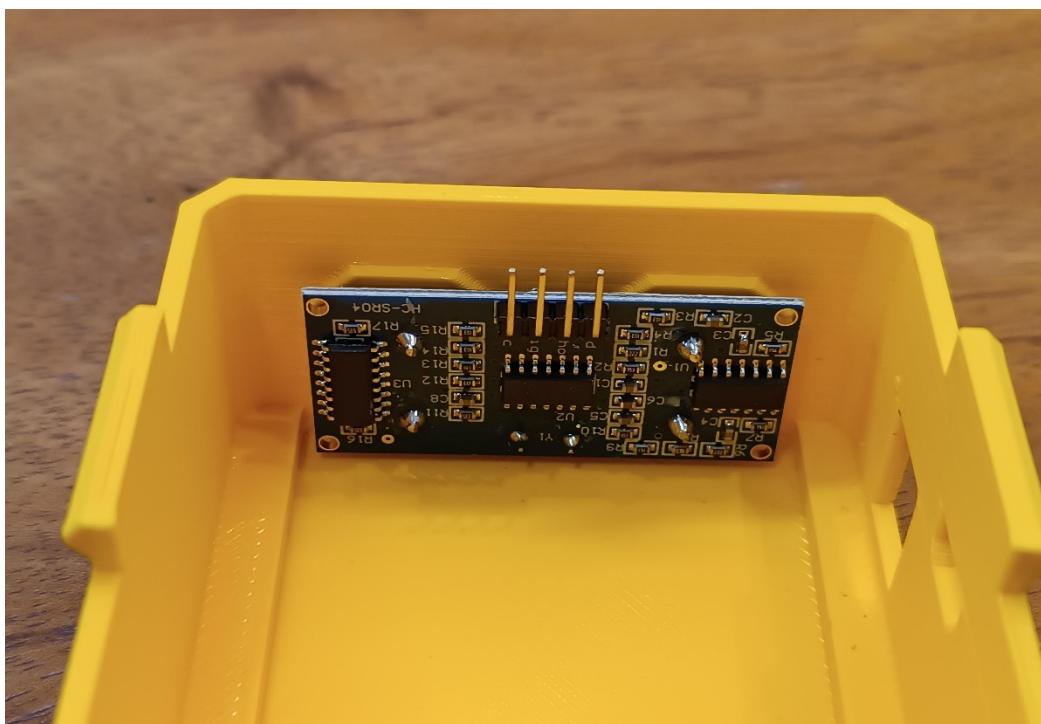
Com o pé posicionado, parafuse cada um deles usando os dois pequenos parafusos restantes, de modo a travá-los permanentemente no eixo do servo:



Feito isso, terminamos a montagem estrutural da parte inferior do Otto. Nesse ponto é esperado que esteja como na imagem abaixo:

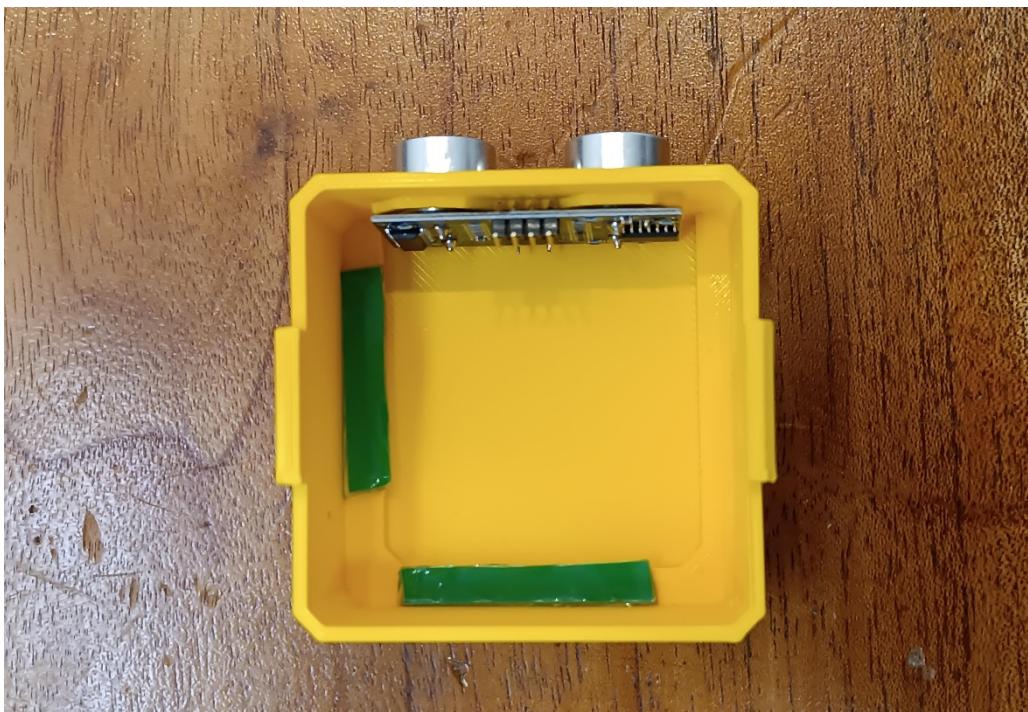


Vamos partir agora para a montagem da cabeça do nosso robô. O processo aqui é bem simples, precisamos apenas encaixar o sensor ultrassônico e a shield com o Arduino. O sensor deverá ser posicionado dessa forma:

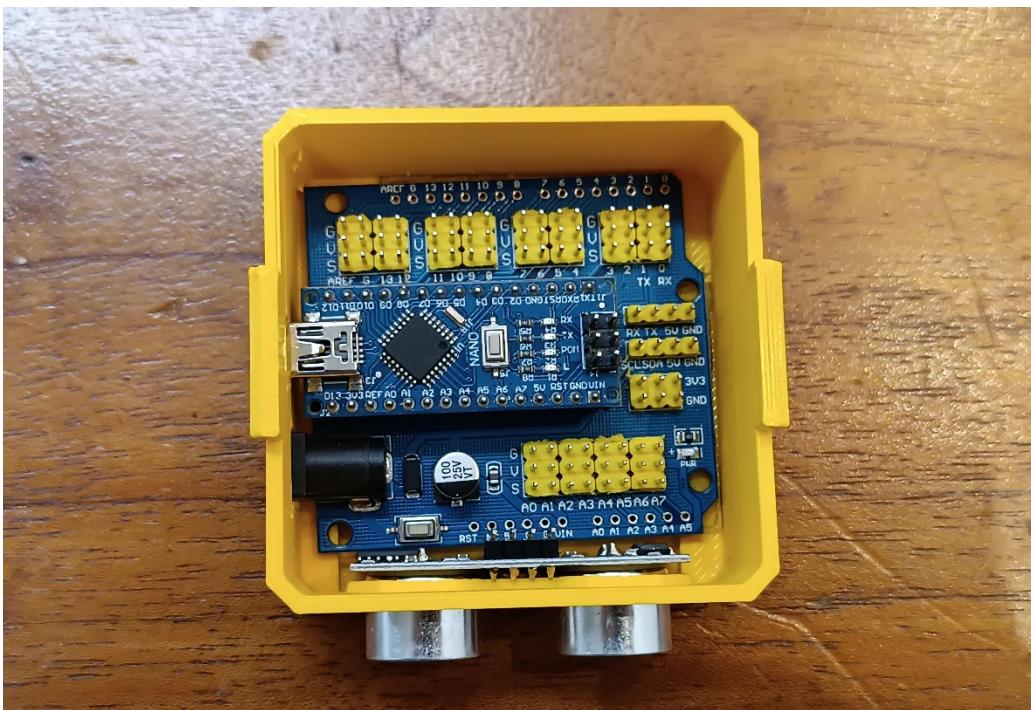


O Arduino e a shield, por sua vez, precisam ser colados na base da peça impressa. Para isso, você pode optar por utilizar um pouco de cola quente ou mesmo fita dupla face do tipo VHB (aqueelas mais emborrachadas), sendo necessário apenas acertar a altura da furação tanto para a porta USB quanto para o conector de energia do Arduino.

Optamos por utilizar a fita dupla face, posicionando-a dessa forma:



Com a shield e o Arduino fixados no lugar:



Nesse ponto, finalizamos a montagem da parte estrutural. O próximo passo é calibrar o regulador de tensão, programar e finalizar a conexão dos componentes.

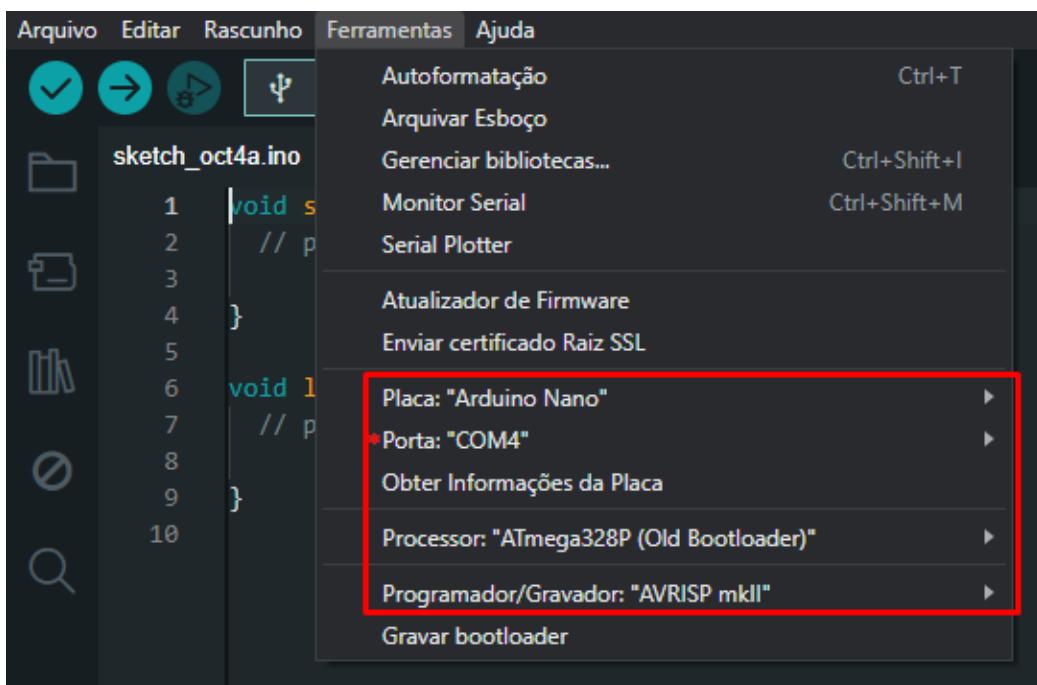
Seguindo nessa ordem, vamos começar abrindo a Arduino IDE. Clique no menu “Ferramentas” e configure da seguinte forma:

**Placa:** Arduino Nano;

**Porta:** A porta COM deve ser a mostrada no Gerenciador de Dispositivos, quando o Arduino está conectado ao computador. No nosso caso, foi a COM4.

**Processador:** ATmega328p (Old Bootloader) – e aqui vale um adendo: o Arduino Nano possui duas versões de processador baseado no ATmega328P. Se essa versão não permitir o carregamento do código, mude para “ATmega328P”.

**Programador:** AVRISP mkII



Com essas configurações feitas, vamos carregar o primeiro código. Ele será responsável por calibrar o regulador para a faixa de tensão que o Arduino e os servos trabalham.

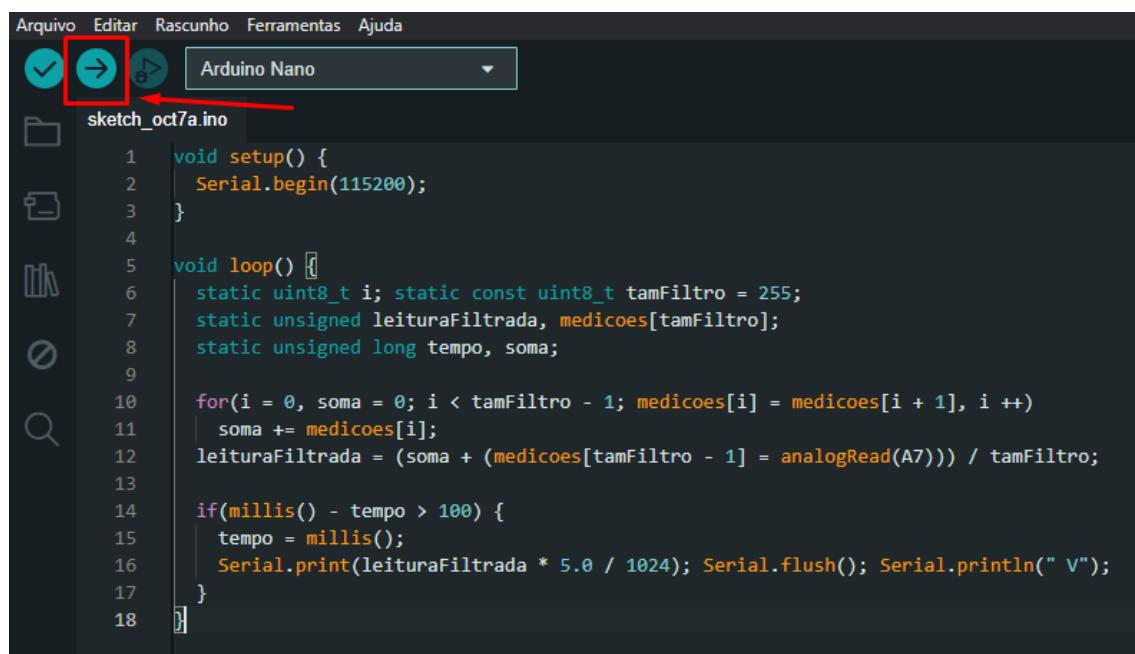
```
void setup() {
    Serial.begin(115200);
}

void loop() {
    static uint8_t i; static const uint8_t tamFiltro = 255;
    static unsigned leituraFiltrada, medicoes[tamFiltro];
    static unsigned long tempo, soma;

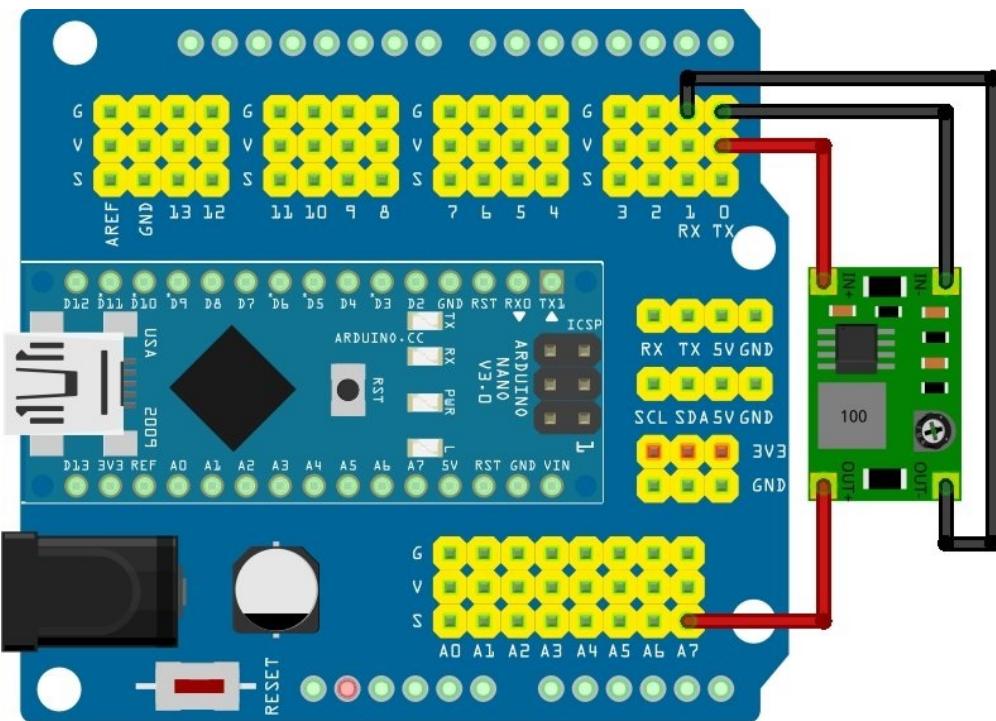
    for(i = 0, soma = 0; i < tamFiltro - 1; medicoes[i] = medicoes[i + 1], i++)
        soma += medicoes[i];
    leituraFiltrada = (soma + (medicoes[tamFiltro - 1] = analogRead(A7))) / tamFiltro;

    if(millis() - tempo > 100) {
        tempo = millis();
        Serial.print(leituraFiltrada * 5.0 / 1024); Serial.flush();
        Serial.println(" V");
    }
}
```

Para carregar, basta clicar no botão “Upload”, mostrado abaixo (com as configurações já feitas, conforme mostrado anteriormente):



Após carregar o código, o regulador de tensão e alguns jumpers para montar o seguinte circuito (com o Arduino desligado do computador):

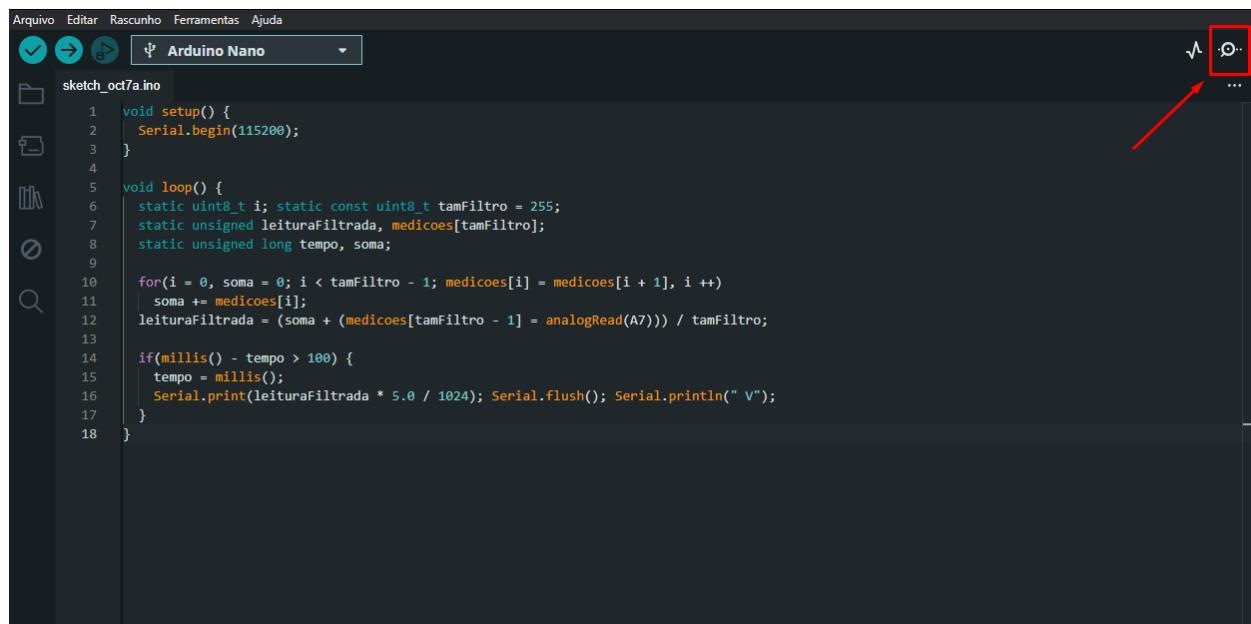


\*\*\*\*\*

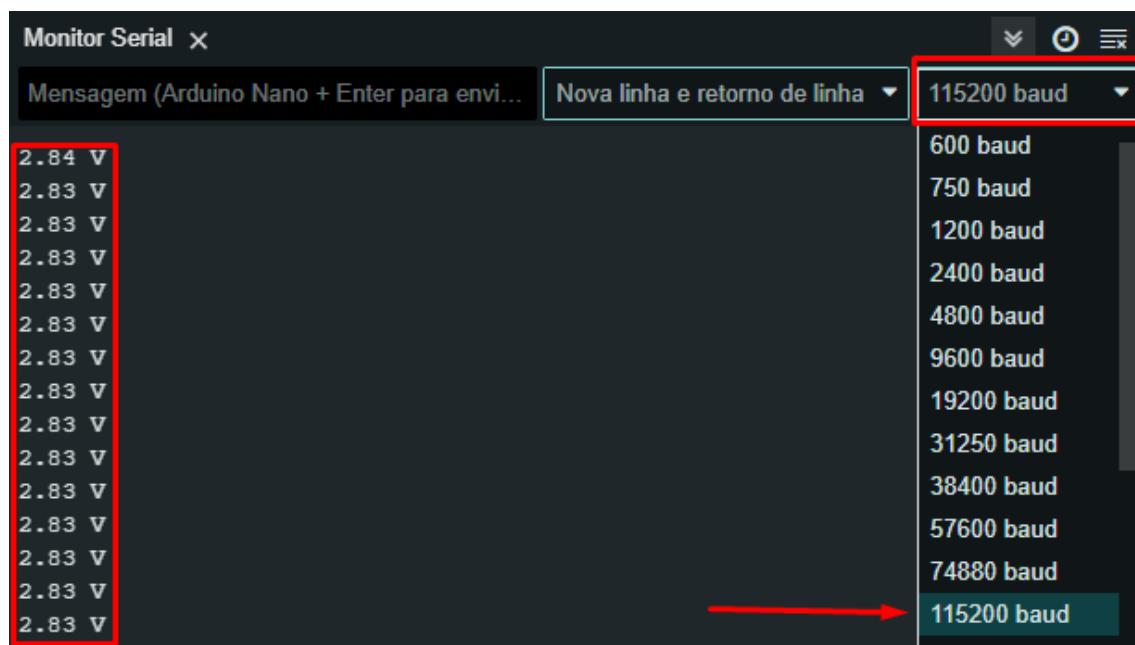
**IMPORTANTE:** Atente-se ao nome dos terminais do regulador, tanto no diagrama quanto no componente físico. Se invertê-los, poderá danificar permanentemente o próprio regulador ou até mesmo o Arduino, no pior dos casos.

\*\*\*\*\*

Com o circuito corretamente montado, conecte o Arduino ao computador e abra o monitor serial da Arduino IDE clicando nesse botão:

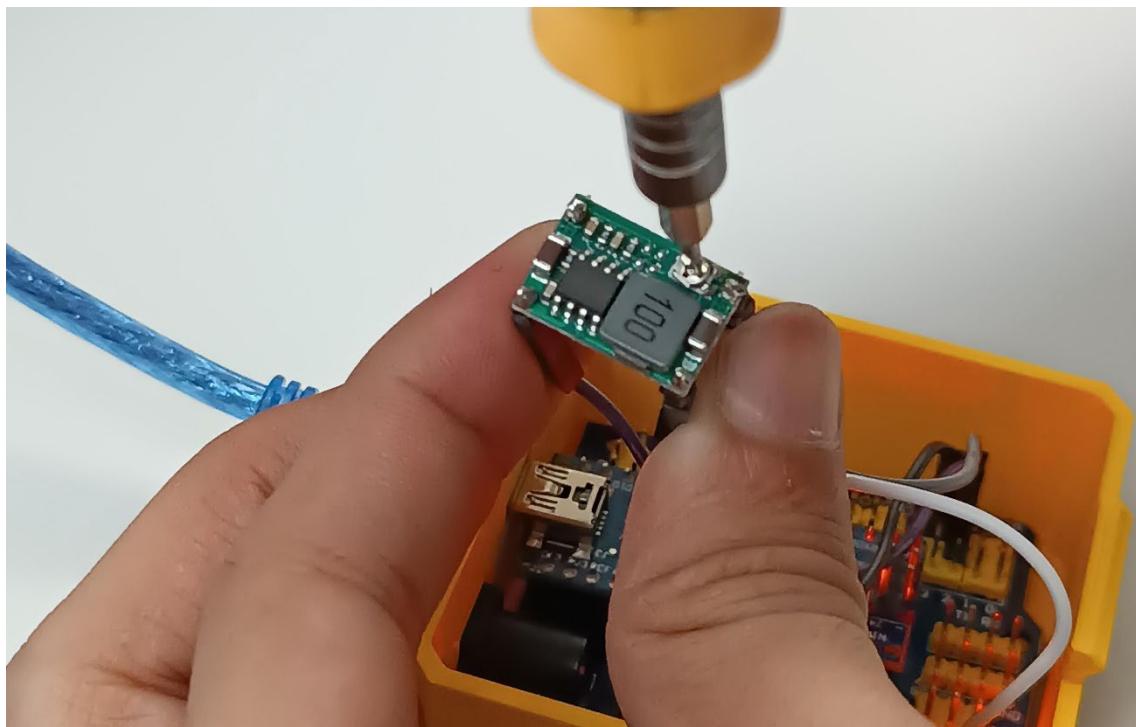


Ao fazer isso, a aba abaixo será exibida na parte inferior. Para exibir os valores lidos, clique na caixinha destacada em vermelho à direita e mude o valor para “115200 bauds”. A saída de dados exibirá o valor de tensão lido pelo regulador:



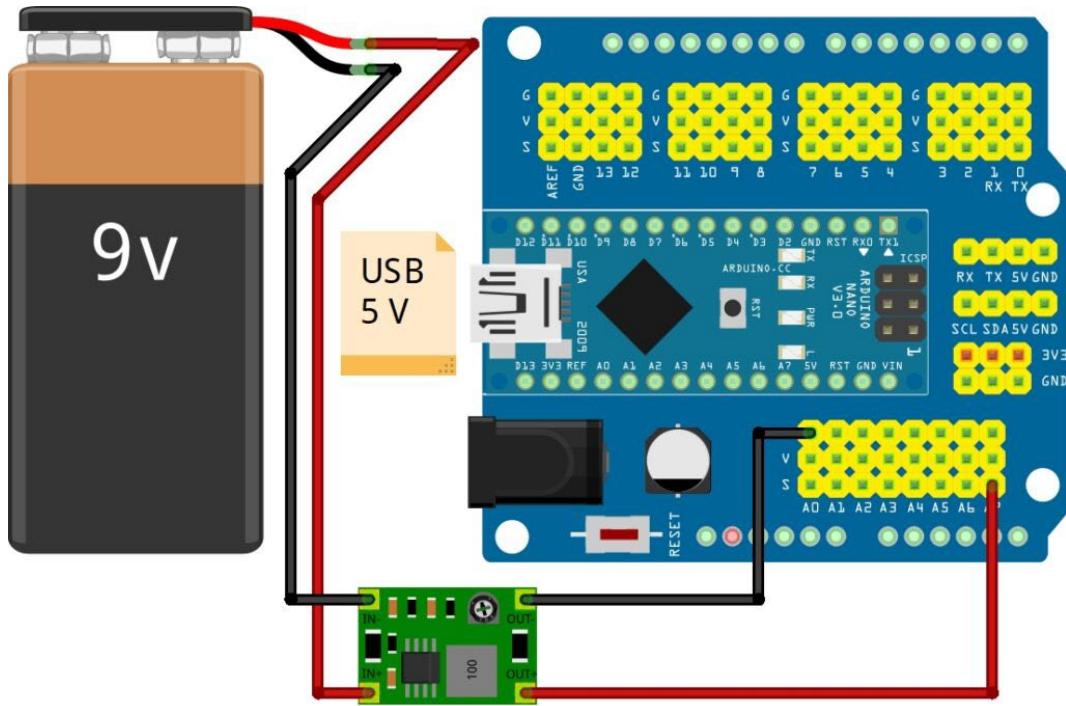
Provavelmente, os valores do seu regulador estarão diferentes nessa etapa, portanto, com o auxílio de uma pequena chave Phillips ou Fenda, gire cuidadosamente o parafuso (trimpot) do regulador, até que o valor exibido esteja entre 2.50 V e 3.0 V.

Fazendo isso, garantimos que o Arduino não irá queimar quando conectarmos a bateria para o ajuste final. 2.84 V, como mostrado na imagem acima, é um excelente valor.



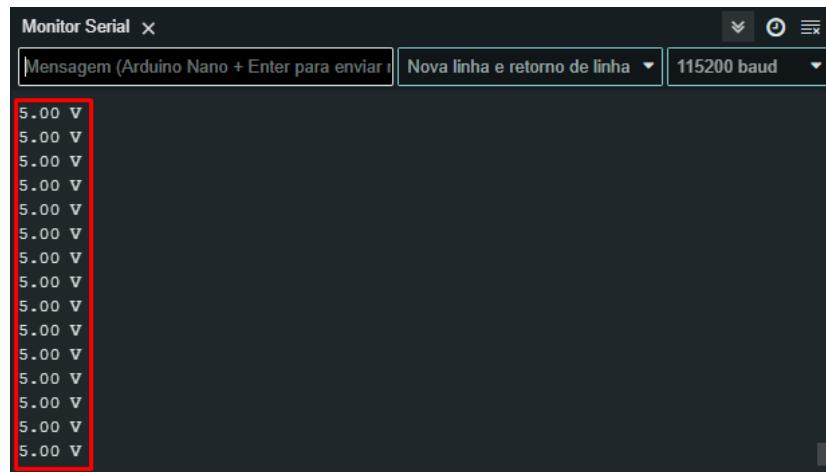
Com o trimpot ajustado para essa faixa, vamos conectar o regulador ao chicote composto pelo conector da bateria e pela chave gangorra. O objetivo é deixarmos a tensão da bateria compatível com os servos e com o Arduino.

Para isso, desligue o Arduino do computador e monte o circuito abaixo:



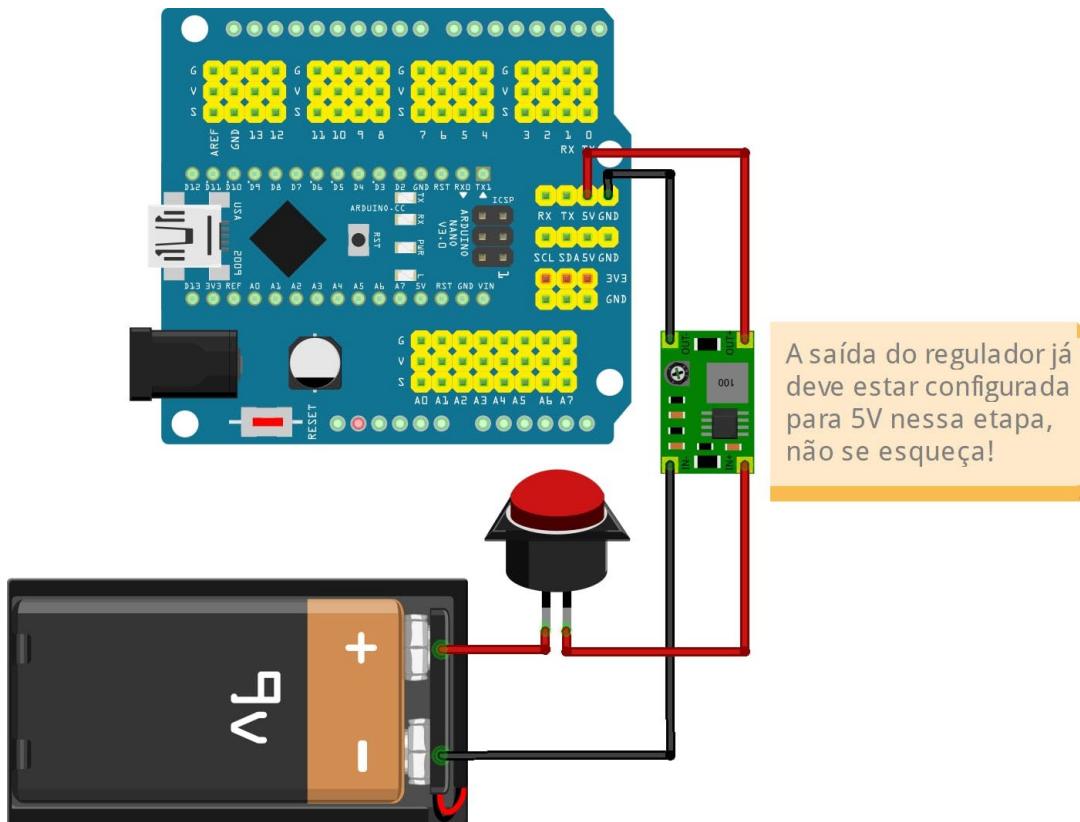
Montado o circuito, conecte novamente o Arduino no computador, abra a IDE e o monitor serial (caso tenha fechado) e repita o ajuste, porém, dessa vez, note que a leitura se inicia no valor configurado anteriormente. Você precisa ajustar, dessa vez, cuidadosamente, para o valor exato de 5V.

Valores acima podem danificar seu Arduino e valores abaixo podem ocasionar mal funcionamento, portanto, seja cuidadoso e tenha muita atenção nessa etapa!



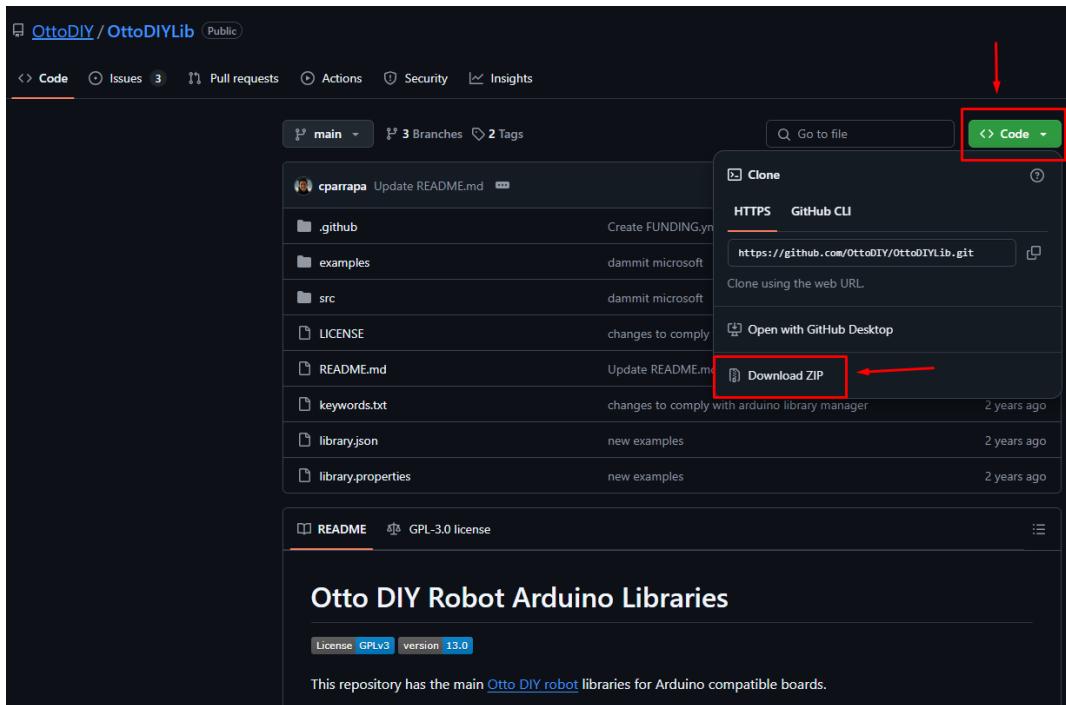
Concluído essa etapa, desligue o Arduino, desconecte o chicote e vamos calibrar os servos antes de carregar o código definitivo do Otto. É importante fazermos isso pois, caso estejam descalibrados, seu robô não se movimentará corretamente.

Comece conectando o chicote da bateria da seguinte forma, mantendo-o desligado por enquanto:

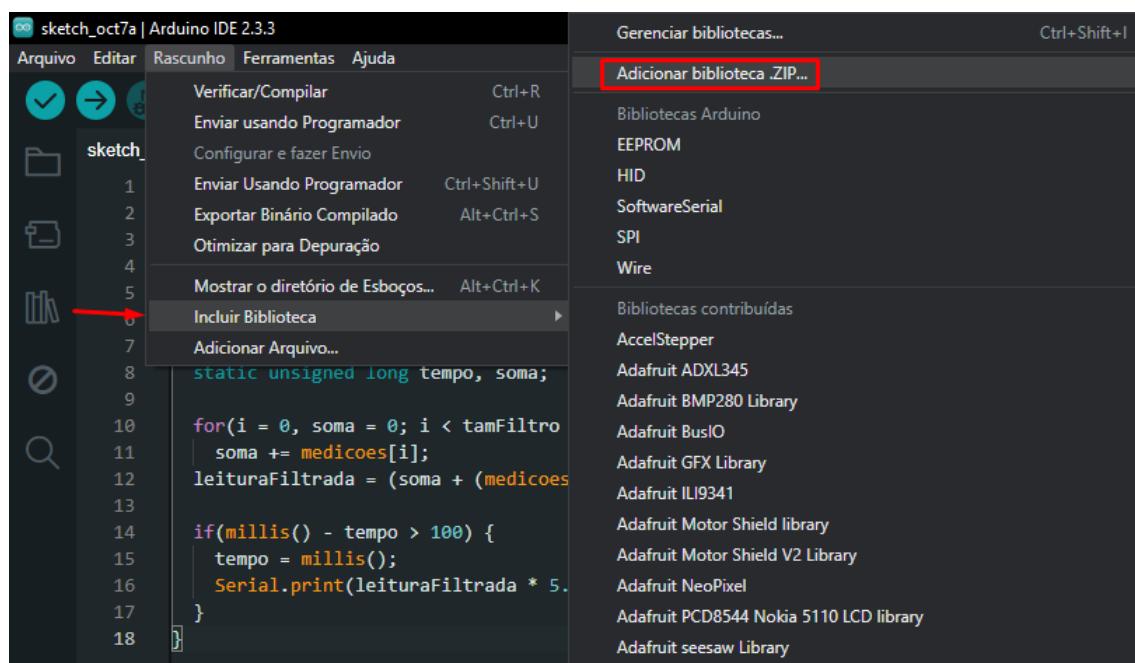


A P O S T I L A K I T  
**ROBÔ OTTO**

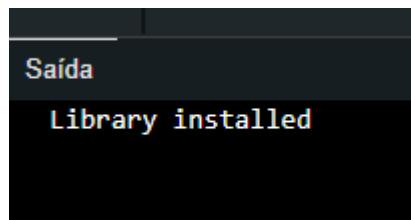
Vamos começar a parte de software instalando a biblioteca de comandos do Otto. Para isso, acesse o [repositório oficial do projeto clicando aqui](#). Em seguida, faça o download da biblioteca, clicando no botão “< > Code” e, em seguida, “Download ZIP”:



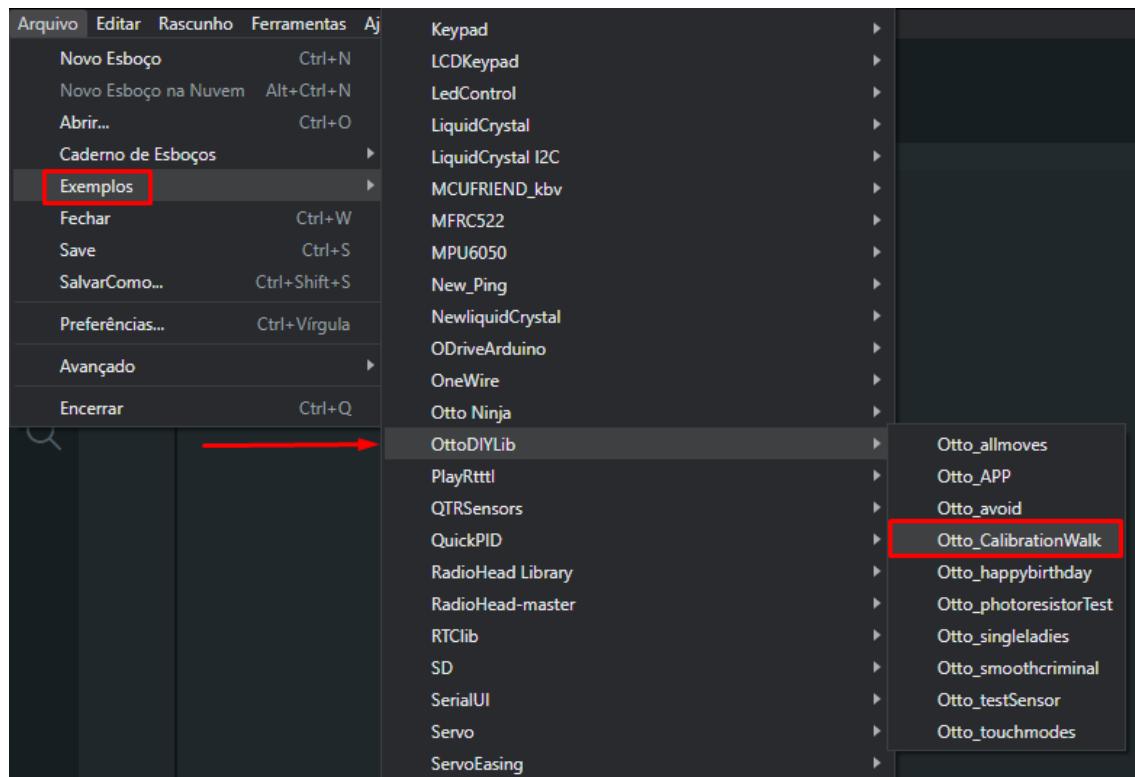
Feito isso, vamos carregá-la na Arduino IDE. Com ela ainda aberta, vá em “Rascunho”, “Adicionar biblioteca .zip” e selecione o arquivo que você acabou de baixar (OttoDIYLib-main.zip), clicando em “Abrir” na sequência.



Você receberá a seguinte mensagem após a instalação ser concluída com sucesso:



Com a biblioteca instalada, vamos calibrar os servos. Para isso, na IDE, vá em Arquivo > Exemplos e procure pela biblioteca OttoDIYLib. Nela, procure pelo exemplo "Otto\_CalibrationWalk".



Abra esse exemplo e carregue o código em seu Arduino.

Depois de carregado, abra o monitor serial e configure a velocidade para 9600 bauds.  
A seguinte mensagem será exibida:

```
Monitor Serial x
Mensagem (Arduino Nano + Enter para enviar mensagem para 'COM4' e
OTTO CALIBRATION PROGRAM
PRESS a or z for adjusting Left Leg
PRESS s or x for adjusting Left Foot
PRESS k or m for adjusting Right Leg
PRESS j or n for adjusting Right Foot

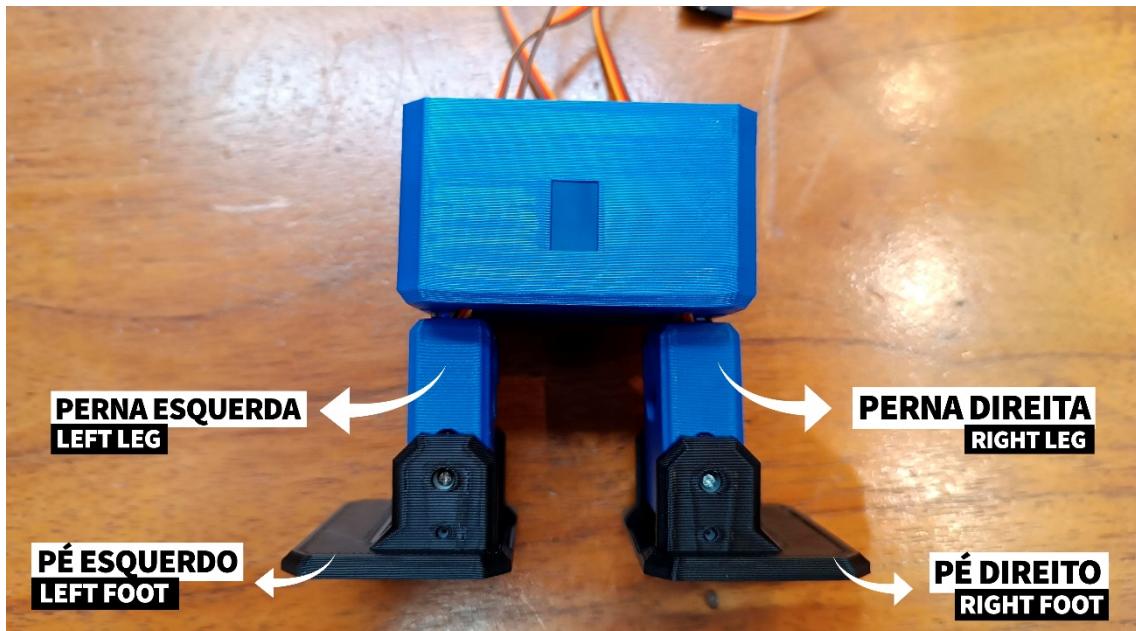
PRESS f to test Otto walking
PRESS h to return servos to home position
```

Desconecte o Arduino do computador e faça as seguintes conexões:

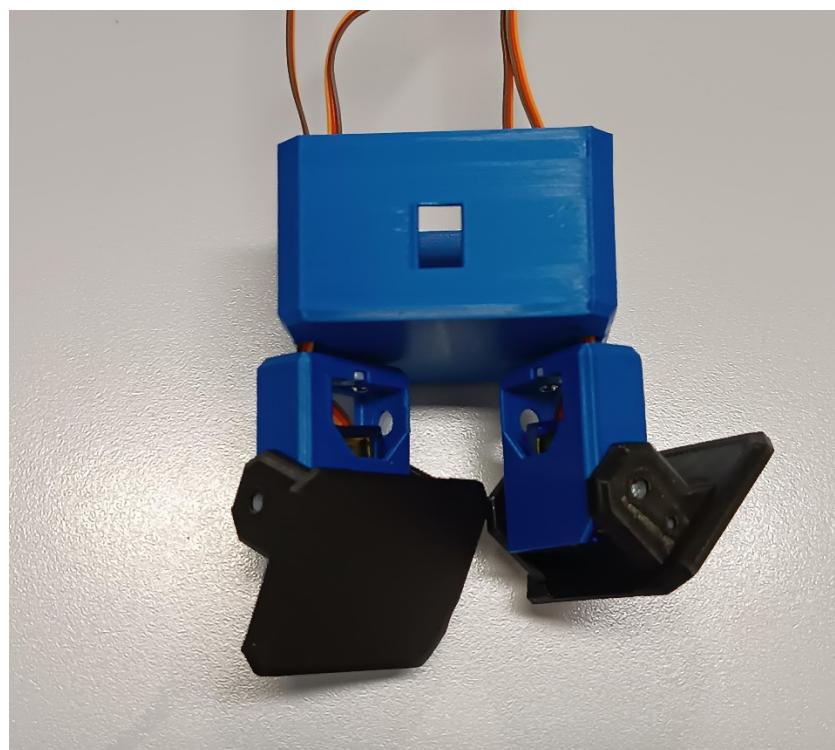


Após conectar os servos, energize o Arduino pela bateria e conecte-o novamente no computador, abrindo novamente o monitor serial.

Com o monitor serial aberto, vamos começar o processo de calibragem, que nada mais é que centralizar os servos individualmente. Para isso, vamos tomar o corpo do Otto com a parte de trás voltada para nós, conforme a imagem abaixo mostra:



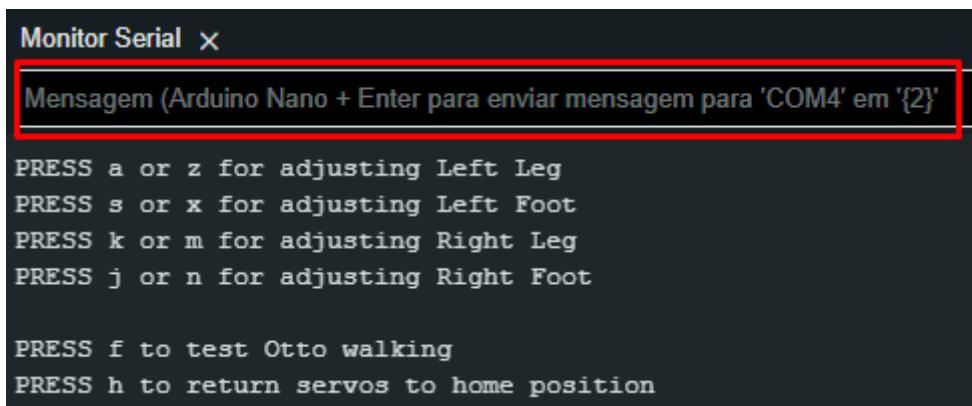
Em nosso caso, ao ligar, os pés se comportaram dessa forma, por isso precisamos alinhá-los.



Note que, no monitor serial, temos 6 opções. Destas, vamos usar apenas 4:

- *PRESS a or z for adjusting Left Leg* (Pressione a ou z para ajustar a perna esquerda);
- *PRESS s or x for adjusting Left Foot* (Pressione s ou x para ajustar o pé esquerdo);
- *PRESS k or m for adjusting Right Leg* (Pressione k ou m para ajustar a perna direita);
- *PRESS j or n for adjusting Right Foot* (Pressione j ou n para ajustar o pé direito);

O processo de ajuste é bem simples. Tomando como exemplo o pé esquerdo, você só precisa, no campo de mensagem do monitor serial, digitar a letra “s” ou a letra “x” e apertar a tecla Enter para que o servo se movimente para a esquerda ou para a direita (no caso dos pés, para cima ou para baixo).



Cada combinação de letra + Enter que você der, o servo referente se moverá um pouquinho. Faça esse ajuste até que as pernas estejam perfeitamente viradas para frente e, os pés, perfeitamente alinhados com a mesa ou bancada em que você estiver trabalhando.

Com tudo alinhado, pode desligar o Arduino tanto da USB como da bateria e desconectar os servos, pois precisaremos carregar o código definitivo do projeto.

É importante fazer essa desconexão pois, como não estamos mais energizando o Arduino pela bateria, a USB não fornece energia suficiente para os servos e isso faz com que o Arduino não consiga ser reconhecido pelo computador no momento de carregar o novo código.

Ah, e não se preocupe com a posição dos servos ao ligar o robô, pois o ajuste fica armazenado na memória do Arduino.

Dito isso, copie e cole o código a seguir (lembre-se de apagar qualquer outro código presente na IDE, beleza?):

```
#include <Otto.h>
#define PernaEsq 2
#define PernaDir 3
#define PeEsq 4
#define PeDir 5
#define Buzzer 13
#define Trigger 8
#define Echo 9

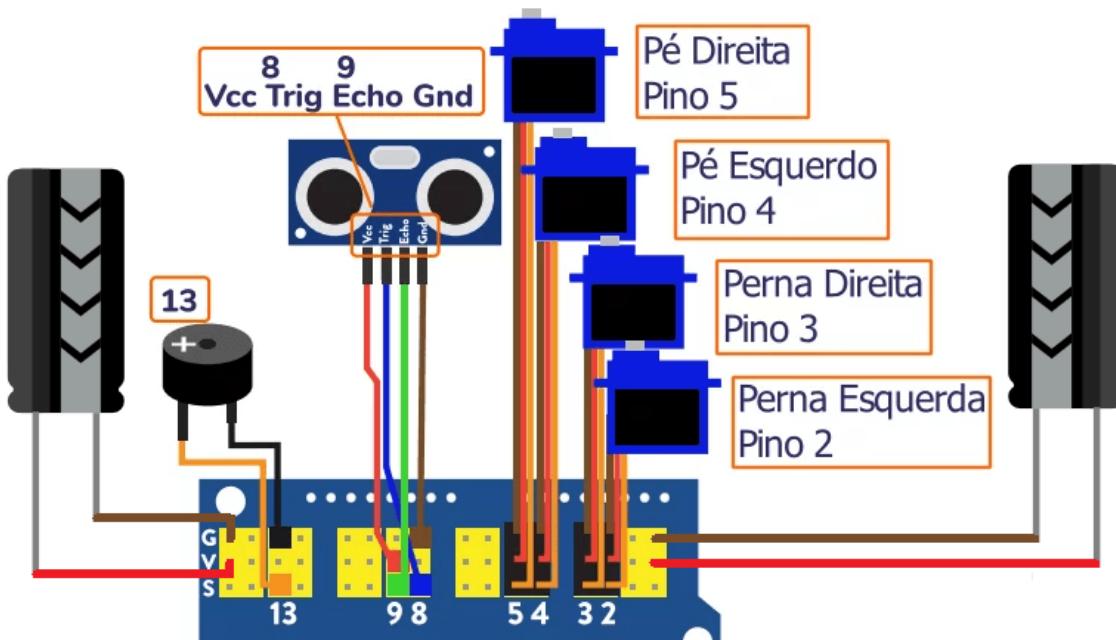
Otto Otto;

long ultrasound(){
    long duration, distance;
    digitalWrite(Trigger, LOW);
    delayMicroseconds(2);
    digitalWrite(Trigger, HIGH);
    delayMicroseconds(10);
    digitalWrite(Trigger, LOW);
    duration = pulseIn(Echo, HIGH);
    distance = duration/58;
    return distance;
}

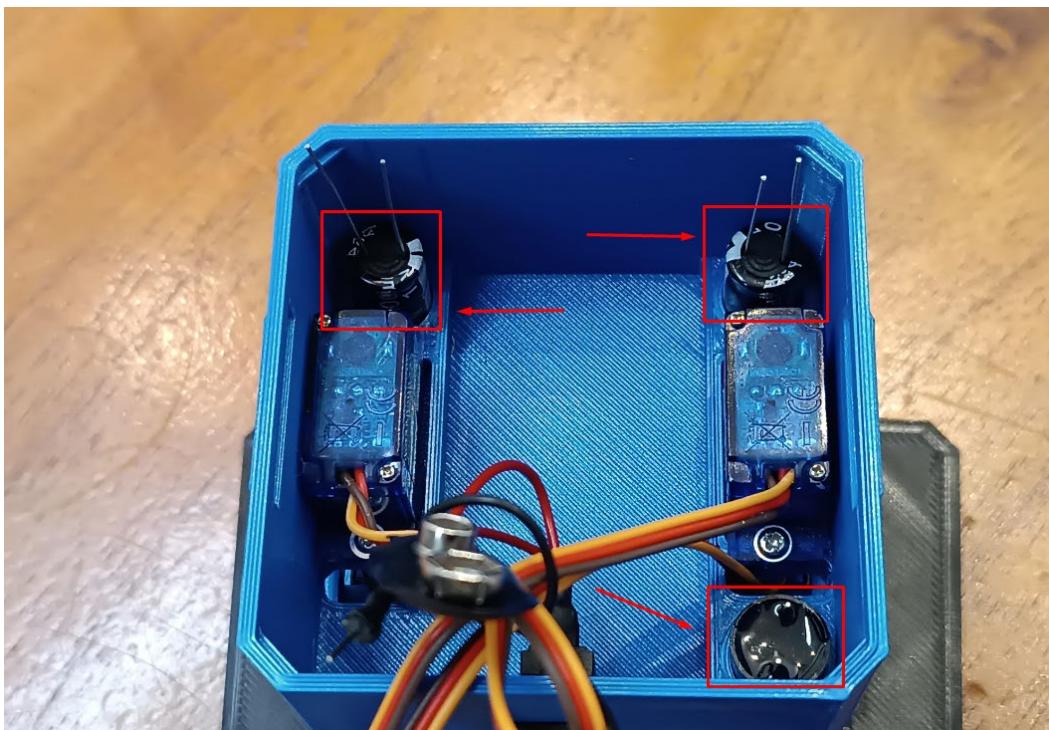
void setup() {
    Otto.init(PernaEsq, PernaDir, PeEsq, PeDir, true, Buzzer);
    pinMode(Trigger, OUTPUT);
    pinMode(Echo, INPUT);
    Otto.swing(2, 1000, 20); // Balançando de um lado para o outro
    Otto.shakeLeg (1,2000,-1); // Chacoalhar a perna direita
}

void loop(){
    if(ultrasound() <= 15){
        Otto.sing(S_surprise); // Emite um som de surpresa
        Otto.walk(2,1000,-1); // Volta dois passo
        Otto.turn(4,1000,1); // Faz uma curva pela esqueda em 3 passos
    }
    Otto.walk(1,1000,1); // Anda um passo para frente
}
```

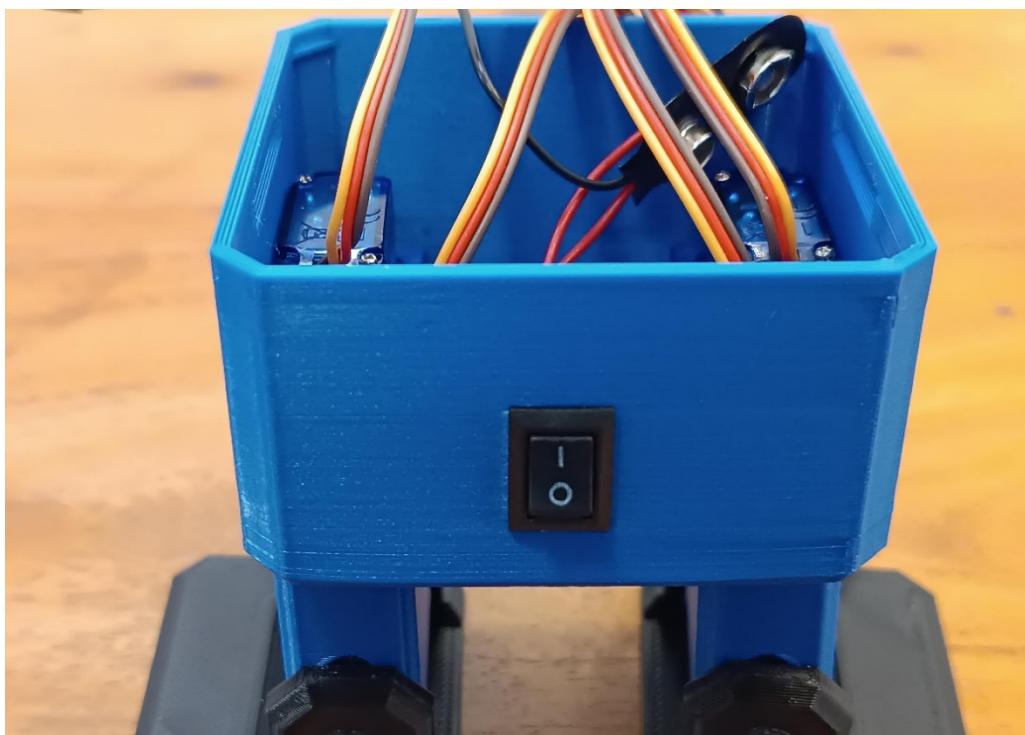
Após carregar o código, desconecte o Arduino do computador e dê início à montagem final do circuito. Siga o diagrama abaixo:



Uma dica ao montar os capacitores e o buzzer é posicioná-los nos locais indicados e prendê-los com um pedacinho de dupla face ou um pequeno pingo de cola quente, para organizar a montagem. Você também pode cortar o excesso de terminais dos capacitores, para não correr riscos de encostarem em algum outro componente:



Com todos os componentes do diagrama conectados, vamos posicionar o chicote da bateria. Primeiramente, encaixe a chave gangorra na parte de trás do corpo do robô, passando todos os fios por dentro do furo e encaixando a chave, de fora para dentro:



Ah, e não se esqueça de reconectar o chicote da bateria conforme o [diagrama da página 29](#), pois o robô não funciona sem ele!

Se você concluiu todos os passos até aqui com sucesso, meus parabéns! Seu Otto está montado e pronto para funcionar. Basta acomodar todos os fios e a bateria internamente, encaixar a cabeça no corpo, colocar os óculos e ligá-lo. Sua montagem final ficará dessa forma:



Basta ligar a chave gangorra e vê-lo se movimentar!

Você pode conferir o funcionamento do robô com esse código [clicando nesse link](#).

## **5. CONSIDERAÇÕES FINAIS**

Com esta apostila, é possível alcançar o funcionamento básico do robô, mesmo que sem conhecimentos anteriores acerca da plataforma Arduino. Entretanto, este projeto pode ser expandido, melhorado e adaptado de diversas maneiras.

Para compreender os componentes e recursos deste projeto, de forma a ser capaz de decidir quais e como alterar para obter as funcionalidades que deseja, recomendamos que explore nosso blog, acessado pelo link <http://blog.eletrogate.com/> e nossas demais apostilas, encontradas em <https://www.eletrogate.com/pagina/apostilas.html>.

APOSTILA

# KIT DIY ROBÔ OTTO

Esta apostila acompanha o KIT ROBÔ OTTO DIY da Eletrogate, e contém conteúdos relacionados a todos os componentes do Kit.

**WWW.ELETROGATE.COM**



ELETROGATE