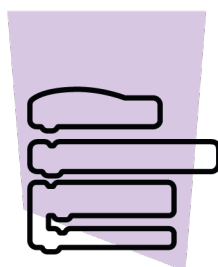
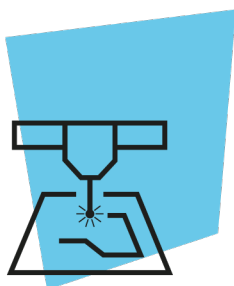
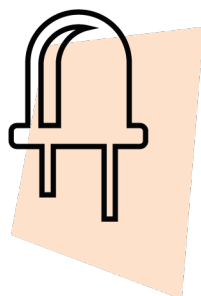
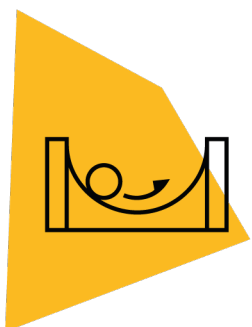
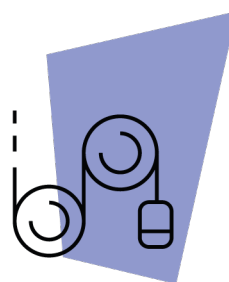
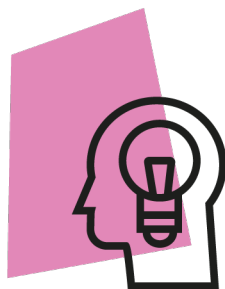


ROBÓTICA

Módulo 2



Robô Wireless*

AULA 39

*Disponível no kit distribuído em 2023

GOVERNADOR DO ESTADO DO PARANÁ

Carlos Massa Ratinho Júnior

SECRETÁRIO DE ESTADO DA EDUCAÇÃO

Renato Feder

DIRETOR DE TECNOLOGIA E INOVAÇÃO

Andre Gustavo Souza Garbosa

COORDENADOR DE TECNOLOGIAS EDUCACIONAIS

Marcelo Gasparin

Produção de Conteúdo

Cleiton Rosa

Michelle dos Santos

Validação de Conteúdo

Cleiton Rosa

Revisão Textual

Adilson Carlos Batista

Leitura Crítica e Normalização Bibliográfica

Ricardo Hasper

Projeto Gráfico e Diagramação

Edna do Rocio Becker

2024

	Aula 01	O que já vimos?
Aula 02	Arduíno: Bibliotecas e Funções	
	Aula 03	Código Morse
Aula 04	Semáforo Inteligente com IR	
	Aula 05	Semáforo Completo com Display
Aula 06	Matriz de LED 8X8	
	Aula 07	Desenhando na matriz de LEDs
Aula 08	Painel de Senhas	
	Aula 09	Escrevendo mensagens
Aula 10	Robô Autônomo	
	Aula 11	Sensor de Chuva
Aula 12	Sensor de Umidade do Solo	
	Aula 13	Irrigador Automático
Aula 14	Feedbacks + Inventário I	
	Aula 15	Teclado Matricial de Membrana
Aula 16	Servos Motores	
	Aula 17	Fechadura Eletrônica
Aula 18	Controlando Servos Motores	
	Aula 19	JoyStick Shield
Aula 20	Braço Robótico	
	Aula 21	Sensor de Movimento Presença
Aula 22	Sensor de Som	
	Aula 23	Sensor de Umidade e Temperatura
Aula 24	Termômetro Digital	
	Aula 25	Sensor de Gás e Fumaça
Aula 26	Acelerômetro e Giroscópio	
	Aula 27	Motor de Passo
Aula 28	Feedbacks + Inventário II	
	Aula 29	Receptor IR e Controle Remoto
Aula 30	Relé	
	Aula 31	Módulo RF 433mhz - I
Aula 32	Módulo RF 433mhz - II	
	Aula 33	Projeto CHAT via RF
Aula 34	Módulo Wireless	
	Aula 35	Comunicação do Módulo WI-FI em HTML
Aula 36	Módulo WI-FI - IoT com Sensores	
	Aula 37	Módulo WI-FI - IoT com Atuadores (LED)
Aula 38	Módulo WI-FI - IoT com Atuadores (Relé)	
	Aula 39	Monitor de Sensores em HTML I
Aula 40	Monitor de Sensores em HTML II	
	Aula 41	Mostra de Robótica
Aula 42	Feedbacks + Inventário III	

Aula 38
**Módulo WI-FI
com Atuador**

Aula 39 Robô Wireless

*Disponível no kit distribuído
em 2023

Aula 40
**Braço Robótico Via RF
(Rádio Frequência)**

Sumário

Introdução	2
Objetivos desta aula	2
Competências gerais previstas na BNCC	3
Habilidades do século XXI a serem desenvolvidas	4
Lista de materiais	4
Roteiro da aula	5
1. Contextualização	5
2. Montagem e programação	7
3. Feedback e finalização	27
Referências	28



Introdução

As tecnologias da Internet das coisas (IoT) vêm mudando como interagimos com a casa e objetos. Casas e cidades inteligentes são termos que já começam a tomar forma e fazer parte de nossa realidade. Um dos maiores benefícios desta aplicação é poder controlar e acompanhar várias “coisas”, de onde você estiver.

Então, imagine isso aplicado nas tarefas domésticas, como, por exemplo, programar um robô aspirador para limpar o chão da casa?

Nesta aula, você aprenderá a controlar um robô através de um celular.



Objetivos desta aula

- Utilizar o Módulo Wireless para controlar um robô através do celular;
- Programar um robô Web controlado;
- Estimular habilidades para desenvolvimento de novos projetos Wireless.

Competências gerais previstas na BNCC

[CG02] - Exercitar a curiosidade intelectual e recorrer à abordagem própria das ciências, incluindo a investigação, a reflexão, a análise crítica, a imaginação e a criatividade, para investigar causas, elaborar e testar hipóteses, formular e resolver problemas e criar soluções (inclusive tecnológicas) com base nos conhecimentos das diferentes áreas.

[CG04] - Utilizar diferentes linguagens – verbal (oral ou visual-motora, como Libras, e escrita), corporal, visual, sonora e digital –, bem como conhecimentos das linguagens artística, matemática e científica, para se expressar e partilhar informações, experiências, ideias e sentimentos em diferentes contextos e produzir sentidos que levem ao entendimento mútuo.

[CG05] - Compreender, utilizar e criar tecnologias digitais de informação e comunicação de forma crítica, significativa, reflexiva e ética nas diversas práticas sociais (incluindo as escolares) para se comunicar, acessar e disseminar informações, produzir conhecimentos, resolver problemas e exercer protagonismo e autoria na vida pessoal e coletiva.

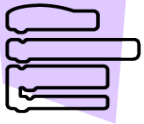
[CG09] - Exercitar a empatia, o diálogo, a resolução de conflitos e a cooperação, fazendo-se respeitar e promovendo o respeito ao outro e aos direitos humanos, com acolhimento e valorização da diversidade de indivíduos e de grupos sociais, seus saberes, identidades, culturas e potencialidades, sem preconceitos de qualquer natureza.

[CG10] - Agir pessoal e coletivamente com autonomia, responsabilidade, flexibilidade, resiliência e determinação, tomando decisões com base em princípios éticos, democráticos, inclusivos, sustentáveis e solidários.

- Pensamento crítico;
- Afinidade digital;
- Resiliência;
- Resolução de problemas;
- Colaboração;
- Comunicação.

Lista de materiais

- 01 Placa Arduino Uno R3;
- 01 Placa Protoboard;
- 01 Cabo USB;
- 01 Ponte H L298n (driver);
- 01 Módulo Wireless ESP01;
- 01 Adaptador ESP-01 para encaixe na protoboard;
- 01 Adaptador USB ESP-01;
- 01 Kit Chassi 2WD Robô para Arduino;
- 04 Jumpers macho-fêmea;
- 07 Jumpers macho-macho;
- 01 Bateria 9V ou compartimento com baterias;
- 01 Chave de fenda pequena;
- Notebook;
- Software Arduino IDE.



Roteiro da aula

1. Contextualização

Diante dos desafios atuais, que incluem mudanças econômicas, sanitárias e globais, a automação nunca foi tão crucial e urgente. A necessidade de cumprir protocolos e adotar novos comportamentos devido à recente crise de saúde global causada pelo novo coronavírus destaca a importância da automação.

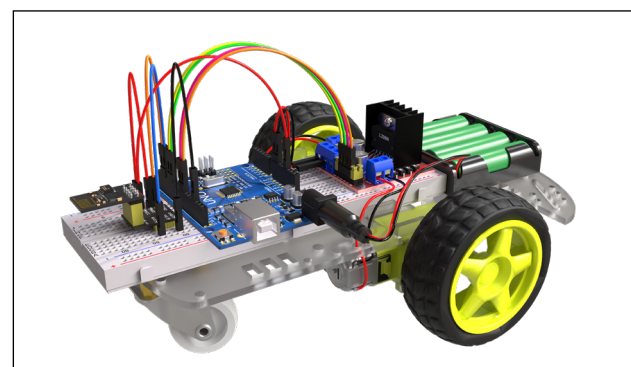
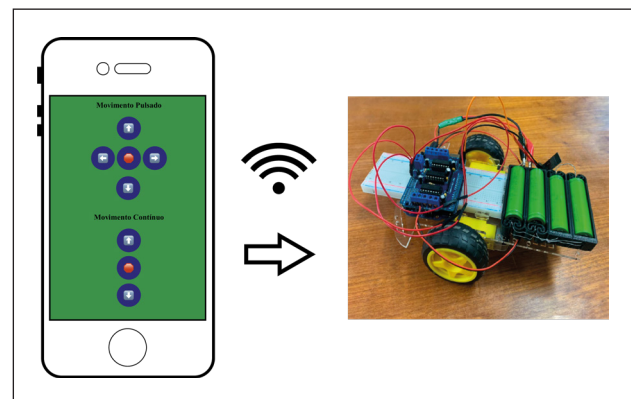
Para preservar a vida das pessoas, tanto em espaços públicos quanto internamente em empresas e comércios, foi necessário implementar sistemas de tecnologia com software e robôs para compensar a falta de mão de obra e manter o distanciamento social.

Nesse contexto, a pandemia da COVID-19 impulsionou a tecnologia, acelerando a transformação digital e o desenvolvimento de robôs mais acessíveis em quase todas as áreas da sociedade. Diante dessa relevância, que tal aprender a programar e controlar um robô por meio de um celular?

Nesta aula, você aprenderá a controlar remotamente um robô

autônomo (Kit Chassi 2WD) por meio de uma página HTML projetada para funcionar como um servidor web. Acessando a página da web em um dispositivo (tablet, notebook ou celular), você poderá guiar o robô usando as teclas de controle (conforme ilustrado na figura 1).

Figura 1 - Teclas de controle do robô



Fonte: SEED/DTI/CTE, 2022

- SAIBA MAIS

Xiaomi anuncia o CyberDog, um cão-robô inteligente e de código aberto.



<https://canaltech.com.br/robotica/xiaomi-cyberdog-cao-robo-192228/>

Robôs inteligentes para manter a produtividade em alta pós pandemia.



<https://exame.com/bussola/precisa-se-de-robos-inteligentes-para-manter-a-produtividade-em-alta/>

Pandemia disparou interesse por uso de robôs.



<https://www.uol.com.br/tilt/noticias/afp/2021/01/13/ces-2021-pandemia-disparou-interesse-por-uso-de-robos.htm>

2. Montagem e programação:

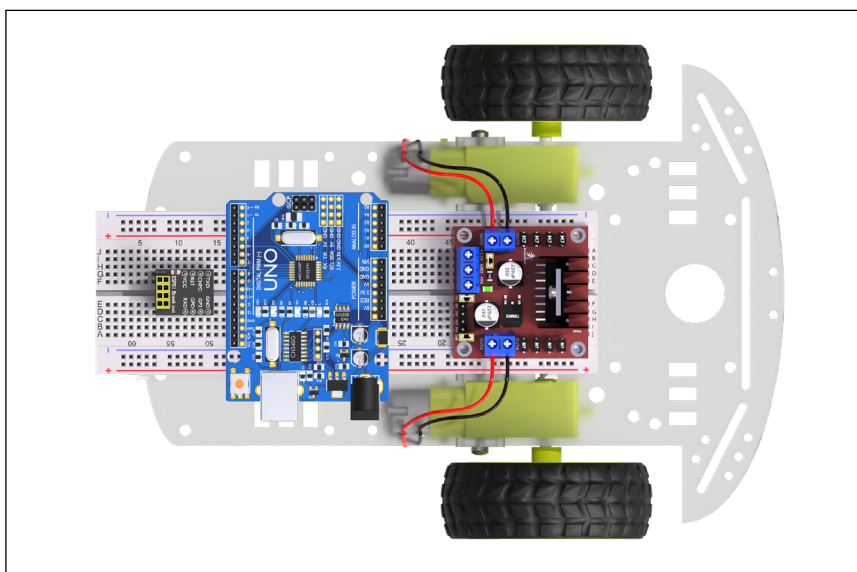
Antes de prosseguirmos com a montagem do nosso robô Wireless, é crucial garantir que o kit chassi 2WD esteja montado corretamente, o que inclui fixar os motores e as rodas na estrutura de acrílico. Caso seja necessário montar ou ajustar algo, você pode consultar o guia de montagem do chassi 2WD por meio deste link: [Manual de montagem do Kit Chassi 2WD](#).

É fundamental destacar que cada etapa descrita a seguir é acompanhada por uma ilustração correspondente, proporcionando um guia visual essencial para facilitar o processo de montagem.

Para começar a montagem, posicione a protoboard sobre o chassi 2WD. Em seguida, encaixe o adaptador do módulo wireless no centro da protoboard e posicione o Arduino e a ponte H conforme mostrado na Figura 2.

Usando uma pequena chave de fenda, fixe os dois pares de fios dos motores nos bornes da ponte H. A ordem dos fios não é relevante neste momento. Durante a fase de programação, faremos testes para garantir que os motores girem na direção desejada. Se necessário, ajustaremos a posição dos fios nesse momento.

Figura 2 - Reunindo os componentes ao Chassi 2WD



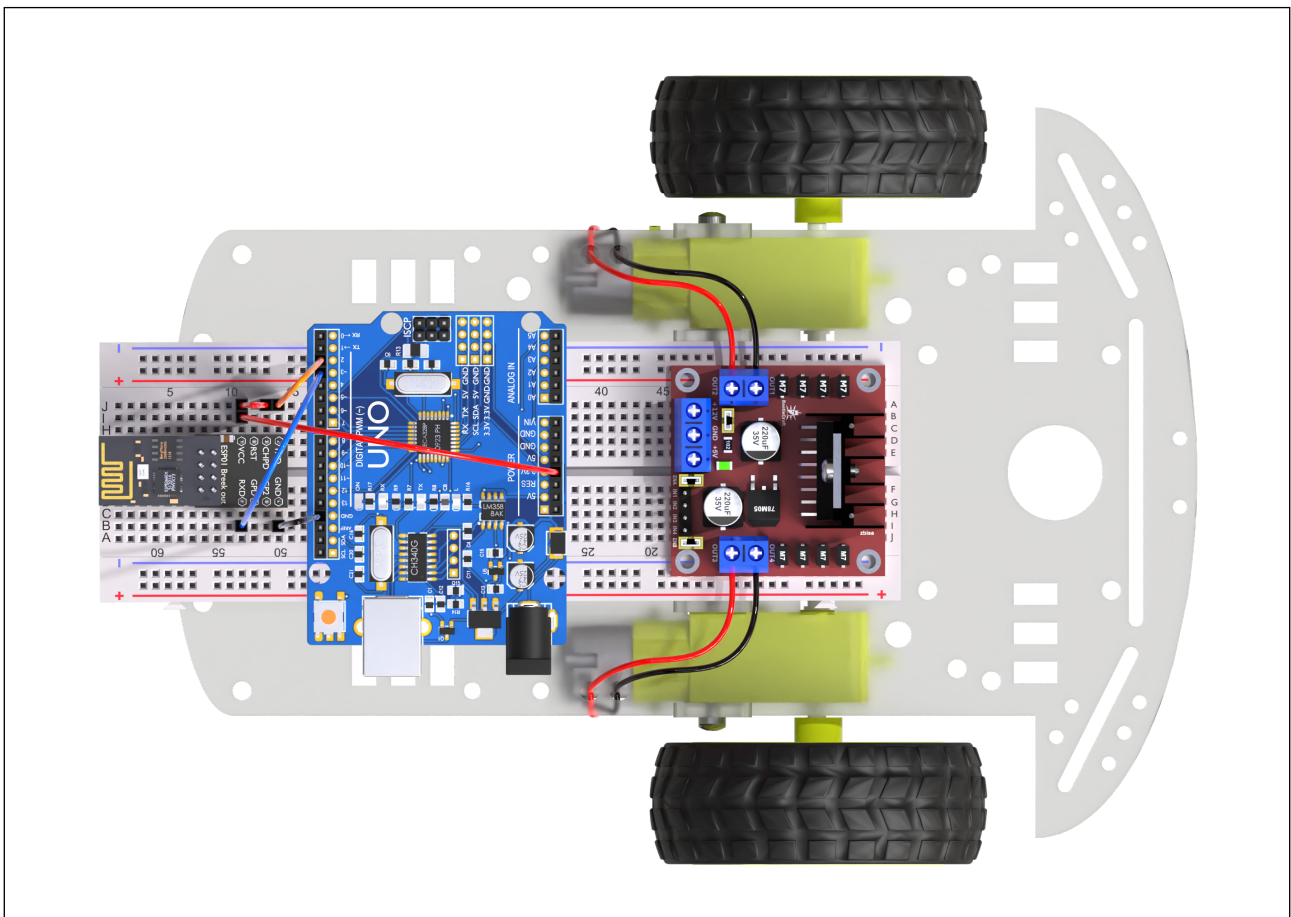
Fonte: Blender, 2024



Após, encaixe o módulo wireless Esp-01 no adaptador e utilize 4 jumpers macho-macho para conectar o módulo wireless ao Arduino da seguinte maneira:

- Conecte um jumper entre os pinos **VCC** e **CHPD** do adaptador.
- Conecte outro jumper entre o pino **VCC** do adaptador e a saída **3,3V** do Arduino.
- Faça a ligação dos pinos **TXD** e **RXD** do adaptador wireless com os pinos digitais **2** e **3**, **respectivamente**, do Arduino.
- Por fim, conecte um jumper entre o pino **GND** do adaptador e a saída **GND** do Arduino, como exemplificado na Figura 3.

Figura 3 - Conexão do módulo wireless ao Arduino



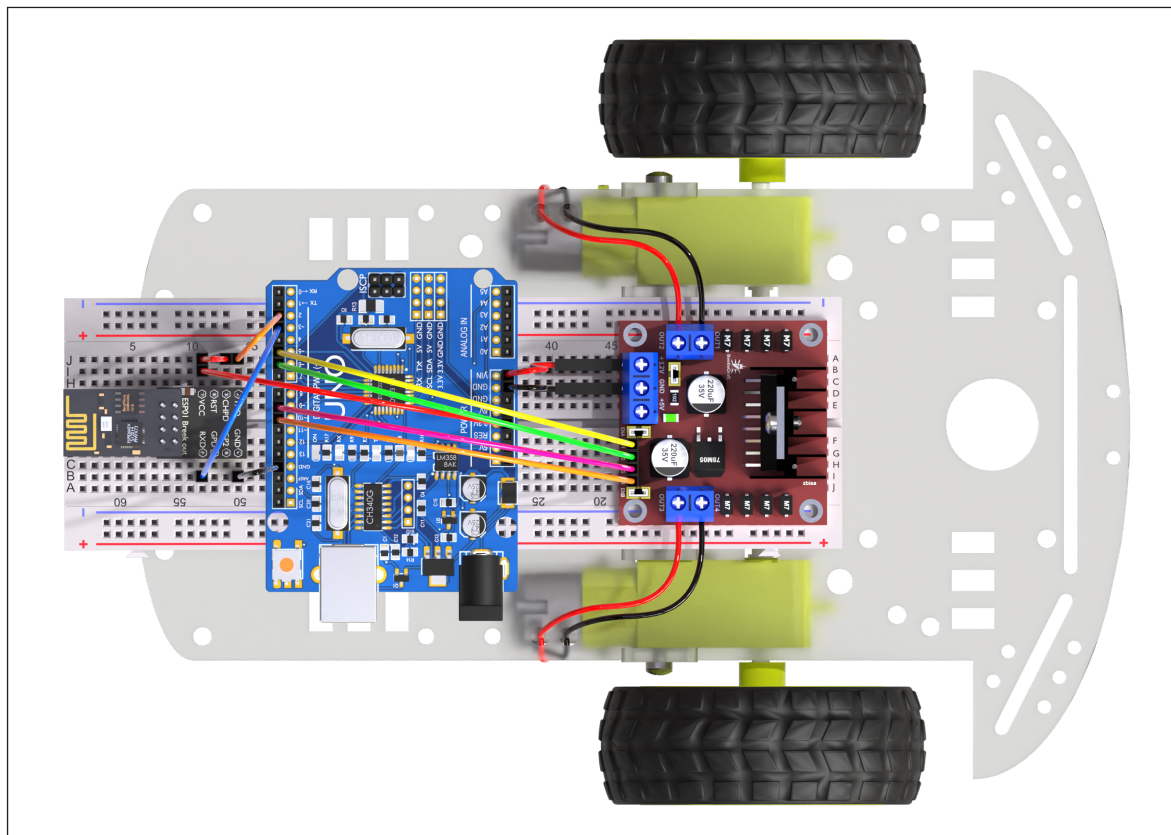
Fonte: Blender, 2024

Para conectar a ponte H ao Arduino, você vai precisar de 4 jumpers macho-fêmea e 2 jumpers macho-macho. Siga estas instruções:

- Utilize 2 jumpers macho-macho para conectar os bornes **12V** e **GND** da ponte H às portas **Vin** e **GND** do Arduino, respectivamente.
- Em seguida, utilize os 4 jumpers macho-fêmea para interligar os pinos **IN1**, **IN2**, **IN3** e **IN4** da ponte H às portas digitais **5**, **6**, **9** e **10** do Arduino, respectivamente.
- É importante observar que os pinos **IN1** a **IN4** da ponte H devem ser conectados às portas digitais que suportam PWM (sinalizadas com o símbolo ~), permitindo assim o controle da potência dos dois motores.

Verifique esses procedimentos na figura 4 a seguir para uma referência visual.

Figura 4 - Conexão da ponte H ao Arduino

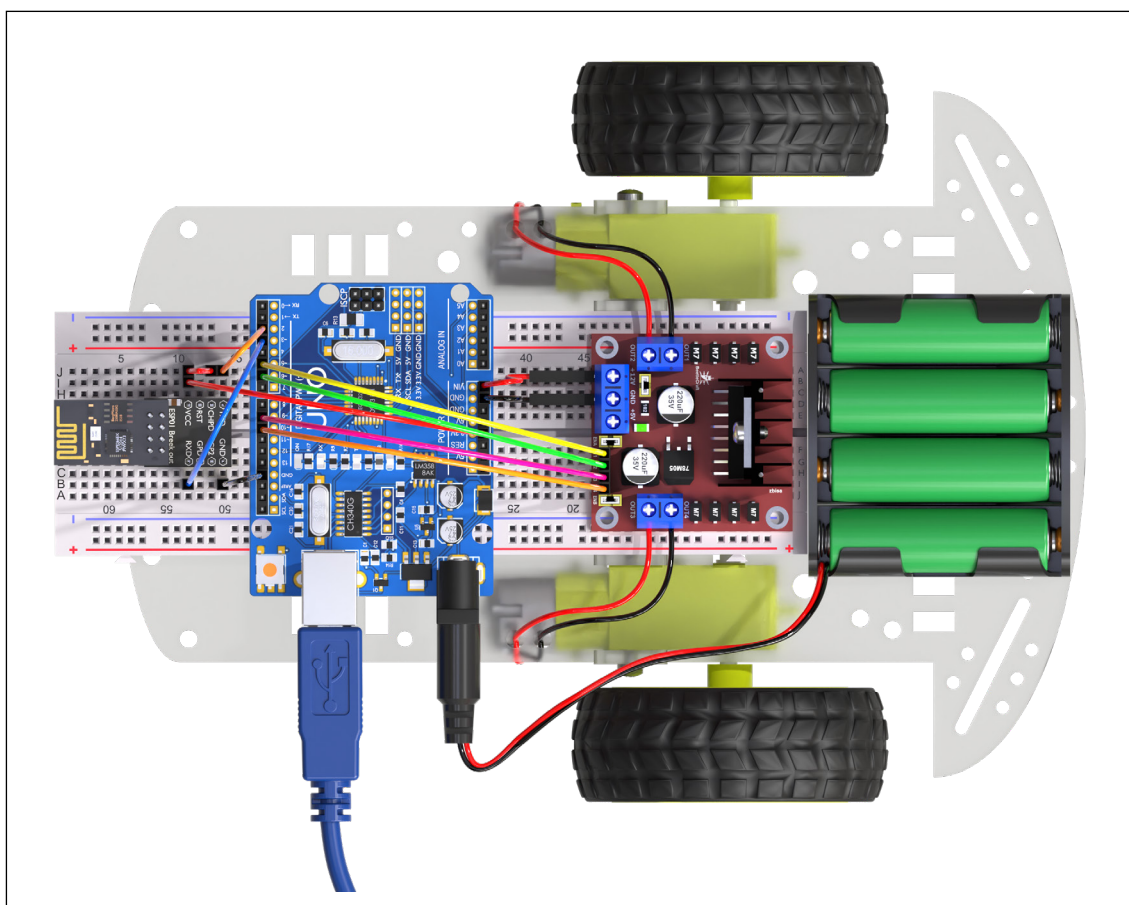


Fonte: Blender, 2024

Agora finalize a montagem conectando o cabo USB e a fonte de energia (pilhas ou bateria) ao conector jack P4 da placa Arduino. É crucial estar atento à fonte de energia utilizada, uma vez que os motores demandam um consumo significativo durante o funcionamento. Portanto, é essencial empregar uma fonte capaz de fornecer uma tensão de 6 a 12 Volts.

Exemplificamos o uso de um estojo com quatro pilhas 18650 (figura 5). No entanto, você pode escolher qualquer outra fonte de energia que satisfaça os requisitos mínimos de operação do seu projeto.

Figura 5 - Conexão da fonte de alimentação e cabo USB ao robô



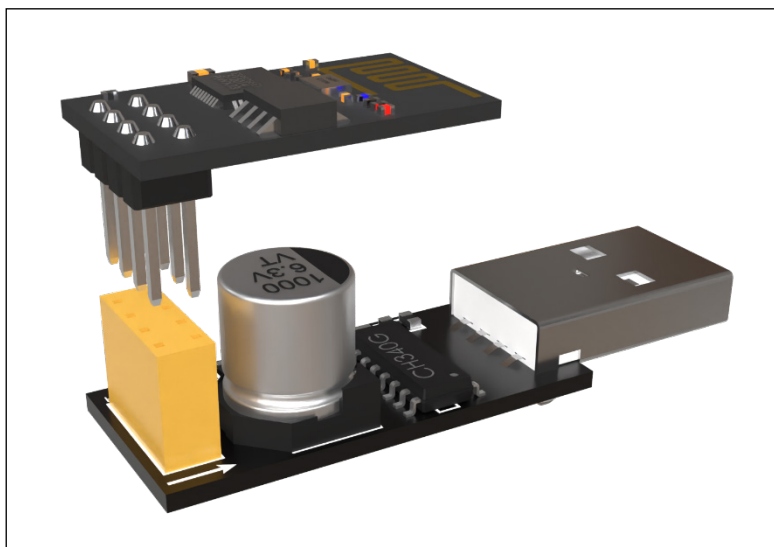
Fonte: Blender, 2024

Após a montagem dos componentes eletrônicos, o próximo passo é programar tanto o módulo Esp-01 quanto o Arduino para controlar os movimentos do robô wireless.

Agora, vamos programar o Módulo Wireless!

Antes de programarmos o Módulo Wireless, é necessário remover o módulo ESP8266 do adaptador ESP-01 e conectá-lo ao adaptador USB para ESP-01. (figura 6).

Figura 6 – Encaixe do módulo ao adaptador USB



Com o adaptador conectado ao notebook, no software Arduino IDE, selecione a **Placa: Generic ESP8266 Module** e a **Porta de comunicação COM** que o notebook atribuiu ao Arduino nos menus **Ferramentas > Placa e Ferramentas > Porta**.

ATENÇÃO

Caso seu Arduino IDE não tenha as informações da **Placa: Generic ESP8266 Module** instaladas, siga as orientações da **Aula 35 - Módulo Wireless**, do Módulo 2, para realizar a instalação.

No software Arduino IDE, escreva ou copie e cole o código-fonte de programação (quadro 1).



IMPORTANTE

Antes de enviar o programa ao adaptador USB, aperte o push button na parte de trás do componente para permitir o controle do módulo ESP8266 e colocá-lo no modo configuração e, assim, conseguir carregar o código desta aula.

ATENÇÃO

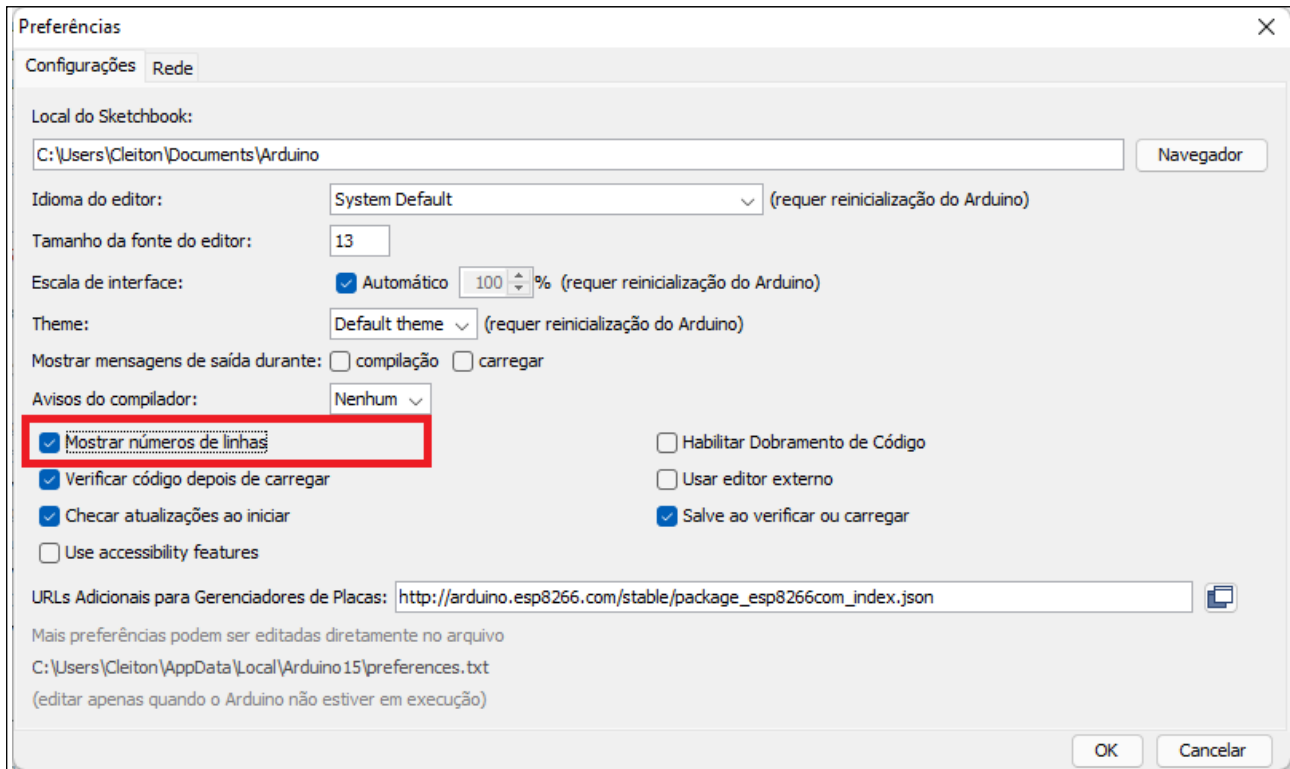
No campo "ALTERAR", na linha 16 do código, você deverá alterar o nome da rede wi-fi para que, após a transferência ao módulo, você consiga localizá-la e conectá-la. Opcionalmente, no campo "12345678", da linha 18, você também poderá alterar a senha para se conectar à sua rede, lembrando que a senha precisa ter no mínimo 8 caracteres.

DICA

Para facilitar a identificação das linhas do código, vá ao menu **Arquivo > Preferências** e marque a caixa **"Mostrar números de linhas"** (Figura 7).



Figura 7 – Ativando o recurso “Mostrar números de linhas” no software Arduino IDE



Quadro 1 - Programação por codificação da página html

```

/*****
/*          Aula 39 - Robô Wireless          */
/*      Programação do Módulo Wireless ESP-01      */
/* Este programa cria em seu módulo wireless um servidor */
/* web. Após conectá-lo e acessá-lo através do endereço */
/* http://192.168.4.1, você terá acesso ao controle de */
/* direção do robô. Os comandos serão: Mover para frente, */
/* mover para trás, girar no sentido horário e girar no */
/* sentido anti-horário. */
*****/

#include <ESP8266WiFi.h> /* Inclui a biblioteca para controle de Wi-Fi*/
#include <ESP8266WebServer.h> /* Inclui a biblioteca para criar um servidor web */

const char *nome_da_rede = "ALTERAR"; /* Define o nome da rede Wi-Fi (SSID) */
const char *senha = "12345678"; /* Define a senha da rede Wi-Fi */

```

```
ESP8266WebServer servidor(80); /* Cria um objeto servidor na porta 80 (HTTP) */

/* Cria uma variável para armazenar a página html do controle */
String pagina = R"===(
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <!-- Defina a metatag viewport para impedir o zoom -->
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-s-
cale=1, user-scalable=no">
    <style>
        body {
            text-align: center;
        }
        .prevent-select {
            -webkit-user-select: none; /* Safari */
            -ms-user-select: none; /* IE 10 and IE 11 */
            user-select: none; /* Standard syntax */
        }

        .joystick-table {
            width: 100%;
            height: 100%;
            border-collapse: collapse;
            table-layout: fixed;
            margin: 0 auto;
        }

        .joystick-cell {
            width: 33.33%;
            text-align: center;
            vertical-align: middle;
        }

        .joystick-button {
            width: 100px; /* Largura dos botões */
            height: 100px; /* Altura dos botões */
            border: 6px solid;
            border-radius: 50%;
            font-size: 18px;
            cursor: pointer;
        }
    </style>
</head>
<body>
    <table class="joystick-table">
        <tr>
            <td class="joystick-cell">
                <div class="joystick-button">
                    <img alt="Joystick button icon" data-bbox="114 202 883 876" />
                </div>
            </td>
            <td class="joystick-cell">
                <div class="joystick-button">
                    <img alt="Joystick button icon" data-bbox="114 202 883 876" />
                </div>
            </td>
            <td class="joystick-cell">
                <div class="joystick-button">
                    <img alt="Joystick button icon" data-bbox="114 202 883 876" />
                </div>
            </td>
        </tr>
    </table>
</body>
</html>
)===;
```

```
        transition: background-color 0.3s, transform 0.3s, box-shadow 0.3s;
        display: inline-block;
    }

    #botao1 {
        background-color: #3498db; /* Cor 1 */
        color: #ffffff;
    }

    #botao1:active {
        background-color: #2980b9;
        transform: translateY(2px);
        box-shadow: none;
    }

    #botao2 {
        background-color: #ff8c66;
        color: #ffffff;
    }

    #botao2:active {
        background-color: #ff5733;
        transform: translateY(2px);
        box-shadow: none;
    }

    #botao3 {
        background-color: #1d8348;
        color: #ffffff;
    }

    #botao3:active {
        background-color: #27ae60;
        transform: translateY(2px);
        box-shadow: none;
    }

    #botao4 {
        background-color: #d35400;
        color: #ffffff;
    }
```

```

        #botao4:active {
            background-color: #f39c12;
            transform: translateY(2px);
            box-shadow: none;
        }

        #emoji-robô {
            font-size: 100px; /* Tamanho do emoji do robô */
        }

    </style>
</head>
<body unselectable="on" class="prevent-select">
    <h2 class="prevent-select">#129302; ROBÓTICA PARANÁ &#129302;</h2>
    <h3 class="prevent-select">#x1F3AE; Aula 39 - Robô Wireless &#128246;</h3>

    <table class="joystick-table">
        <tr>
            <td class="joystick-cell"></td>
            <td class="joystick-cell"><button class="prevent-select joystick-but-
ton" id="botao1"></button></td>
            <td class="joystick-cell"></td>
        </tr>
        <tr>
            <td class="joystick-cell"><button class="prevent-select joystick-but-
ton" id="botao2"></button></td>
            <td class="joystick-cell"><div id="emoji-robô" class="prevent-select">
🤖</div></td>
            <td class="joystick-cell"><button class="prevent-select joystick-but-
ton" id="botao3"></button></td>
        </tr>
        <tr>
            <td class="joystick-cell"></td>
            <td class="joystick-cell"><button class="prevent-select joystick-but-
ton" id="botao4"></button></td>
            <td class="joystick-cell"></td>
        </tr>
    </table>

    <script>
        const botao1 = document.getElementById('botao1');
        const botao2 = document.getElementById('botao2');
        const botao3 = document.getElementById('botao3');
        const botao4 = document.getElementById('botao4');

```

```
let intervalID = null;

botao1.addEventListener('touchstart', () => {
  botao1.style.backgroundColor = '#2980b9';
  botao1.style.transform = 'translateY(2px)';
  botao1.style.boxShadow = 'none';
  intervalID = setInterval(() => {
    fetch('/pressionado', { method: 'POST', body: '1' });
  }, 100);
});

botao1.addEventListener('touchend', () => {
  botao1.style.backgroundColor = '#3498db';
  botao1.style.transform = 'translateY(0)';
  botao1.style.boxShadow = 'none';
  clearInterval(intervalID);
  fetch('/pressionado', { method: 'POST', body: '0' });
});

botao2.addEventListener('touchstart', () => {
  botao2.style.backgroundColor = '#ff8c66';
  botao2.style.transform = 'translateY(2px)';
  botao2.style.boxShadow = 'none';
  intervalID = setInterval(() => {
    fetch('/pressionado', { method: 'POST', body: '2' });
  }, 100);
});

botao2.addEventListener('touchend', () => {
  botao2.style.backgroundColor = '#ff5733';
  botao2.style.transform = 'translateY(0)';
  botao2.style.boxShadow = 'none';
  clearInterval(intervalID);
  fetch('/pressionado', { method: 'POST', body: '0' });
});

botao3.addEventListener('touchstart', () => {
  botao3.style.backgroundColor = '#1d8348';
  botao3.style.transform = 'translateY(2px)';
  botao3.style.boxShadow = 'none';
  intervalID = setInterval(() => {
    fetch('/pressionado', { method: 'POST', body: '3' });
  }, 100);
});
```

```

    botao3.addEventListener('touchend', () => {
        botao3.style.backgroundColor = '#27ae60';
        botao3.style.transform = 'translateY(0)';
        botao3.style.boxShadow = 'none';
        clearInterval(intervalID);
        fetch('/pressionado', { method: 'POST', body: '0' });
    });

    botao4.addEventListener('touchstart', () => {
        botao4.style.backgroundColor = '#d35400';
        botao4.style.transform = 'translateY(2px)';
        botao4.style.boxShadow = 'none';
        intervalID = setInterval(() => {
            fetch('/pressionado', { method: 'POST', body: '4' });
        }, 100);
    });

    botao4.addEventListener('touchend', () => {
        botao4.style.backgroundColor = '#f39c12';
        botao4.style.transform = 'translateY(0)';
        botao4.style.boxShadow = 'none';
        clearInterval(intervalID);
        fetch('/pressionado', { method: 'POST', body: '0' });
    });
</script>
</body>
</html>

)===";

void setup() {
    /* Função de configuração, executada uma vez */
    Serial.begin(9600); /* Inicia a comunicação serial a 9600 bps */
    WiFi.softAP(nome_da_rede, senha); /* Inicia o ponto de acesso Wi-Fi com o SSID
e senha */

    servidor.on("/", []() { /* Define a ação ao acessar a raiz ("/") do servidor */
        servidor.send(200, "text/html", pagina); /* Envia uma resposta 200 OK com
conteúdo HTML */
    });
}

```

```
servidor.on("/pressionado", HTTP_POST, []() { /* Define a ação para requisições
POST na rota "/pressionado" */
    String estado = servidor.arg("plain"); /* Obtém o corpo da requisição e ar-
mazena na variável 'estado' */
    Serial.println(estado); /* Imprime o estado no monitor serial */
    servidor.send(200, "text/plain", "OK"); /* Envia uma resposta 200 OK com o
texto "OK" */
});

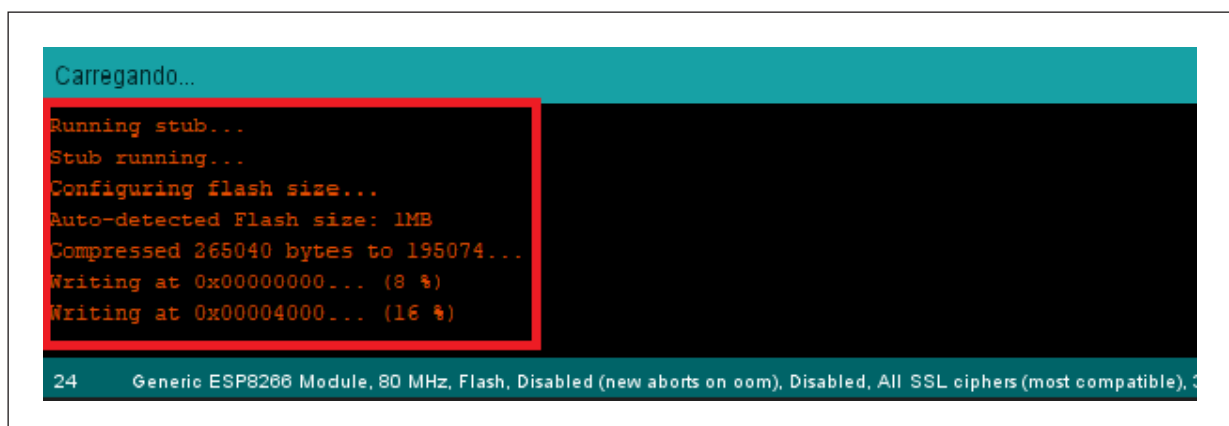
servidor.begin(); /* Inicia o servidor web, permitindo que ele comece a pro-
cessar requisições */
}

void loop() {
servidor.handleClient(); /* Processa as requisições recebidas pelo servidor web
*/
}
```

Para carregar o programa ao Módulo Wireless, clique em **Verificar** e a seguir, **Carregar** para enviar o programa ao Módulo Wireless.

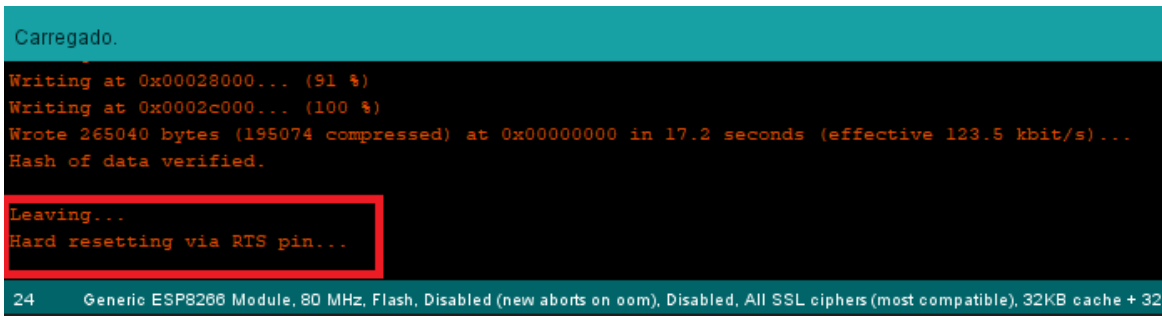
Durante o carregamento, é indicado, na parte inferior do Software Arduino IDE, o progresso do carregamento, como mostrado na figura 8.

Figura 8 - Gravação do código ao Módulo Wireless



Ao terminar o processo de gravação, será apresentada a mensagem **Hard resetting via RTS pin...**, (figura 9).

Figura 9 - Término da gravação do código ao Modulo Wireless



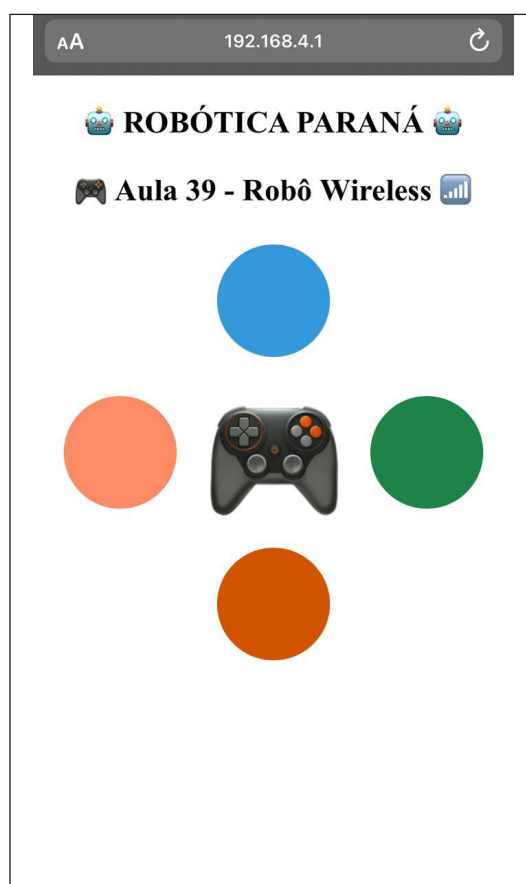
```
Carregado.
Writing at 0x00028000... (91 %)
Writing at 0x0002c000... (100 %)
Wrote 265040 bytes (195074 compressed) at 0x00000000 in 17.2 seconds (effective 123.5 kbit/s)...
Hash of data verified.
Leaving...
Hard resetting via RTS pin...
24 Generic ESP8266 Module, 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32
```

Após reiniciar o Módulo Wireless conectando o adaptador USB ao computador sem pressionar o jumper, aguarde aproximadamente 30 segundos. Em seguida, você poderá observar que foi criada uma rede Wi-Fi com o nome que você escolheu anteriormente ao carregar o código.

Conecte-se a essa rede utilizando um dispositivo (como um smartphone, notebook ou tablet) e insira a senha "12345678", caso não tenha alterado no código anteriormente.

Após a conexão bem-sucedida, abra o navegador do seu dispositivo e acesse o endereço de IP padrão do Módulo, que é <http://192.168.4.1>. Isso abrirá a página criada no código-fonte da programação (consulte a Figura 10 para referência).

Figura 10 - Acesso da página HTML criada no Módulo Wireless



Dicas

- Para criar suas páginas HTML acesse o site w3school através do link: <https://www.w3schools.com/>
- Para converter a página HTML na linguagem que o Software Arduino IDE compreenda (C/C++), utilize a ferramenta online através do link http://tomeko.net/online_tools/cpp_text_escape.php?lang=en

Após concluir esta primeira etapa, podemos prosseguir reconectando o módulo wireless ao adaptador de protoboard no seu robô.

Agora, vamos programar o Arduino!

Antes de começarmos a programar o Arduino, precisamos ajustar a configuração da placa no Software Arduino IDE. Com o Arduino conectado ao Notebook, siga os seguintes passos:

1. No menu “Ferramentas”, selecione “**Placa**” e escolha “**Generic ESP8266 Module**”.
2. Em seguida, vá para “**Arduino AVR Boards**” e selecione “**Arduino Uno**”.
3. Escolha a porta de comunicação **COM** que o Notebook atribuiu ao Arduino. Isso pode ser feito em “**Ferramentas**” > “**Porta**” > “**COMx**”.

Depois de configurar corretamente a placa e a porta de comunicação, carregue o código mostrado no quadro 2 em seu Arduino e prossiga com os próximos passos.

Quadro 2 - Código-fonte do Arduino (Robô)

```
/* **** */
/* Aula 39 - Robô Wireless */
/* Programação da placa Arduino. */
/* Este programa faz seu robô se mover com comandos */
/* recebidos de um dispositivo conectado à sua rede */
/* wireless. Os comandos que o robô irá receber serão: */
/* Mover para frente, mover para trás, girar no sentido */
/* horário e girar no sentido anti-horário. */
/* **** */

/* Inclui a biblioteca para a comunicação serial. */
#include <SoftwareSerial.h>

/* Cria o objeto espSerial especificando as portas de */
/* comunicação. */
SoftwareSerial espSerial(2, 3);

/* Definições dos pinos IN1 a IN4 da ponte H. */
/* Para controlar a velocidade dos motores, esses pinos */
/* deverão ser conectados em portas com o recurso PWM. */

#define IN1 10 // Porta PWM ~10
#define IN2 9 // Porta PWM ~9
#define IN3 6 // Porta PWM ~6
#define IN4 5 // Porta PWM ~5

/* Variável que define as potências dos motores (0 a 255). */
int potA = 255;
int potB = 255;

/* Variável que define as potências de manobra dos motores (0 a 255). */
int pot_manobra = 150;

/* Variável que receberá os comandos do controle. */
char codigo;

/* Criando as funções de controle dos motores. */
```

```
/* Move para frente. */
void Frente() {
    /* Motor A */
    analogWrite(IN1, potA);
    analogWrite(IN2, 0);
    /* Motor B */
    analogWrite(IN3, potB);
    analogWrite(IN4, 0);
}

/* Move para trás. */
void Re() {
    /* Motor A */
    analogWrite(IN1, 0);
    analogWrite(IN2, potA);
    /* Motor B */
    analogWrite(IN3, 0);
    analogWrite(IN4, potB);
}

/* Giro no sentido horário. */
void GiroHorario() {
    /* Motor A */
    analogWrite(IN1, pot_manobra);
    analogWrite(IN2, 0);
    /* Motor B */
    analogWrite(IN3, 0);
    analogWrite(IN4, pot_manobra);
}

/* Giro no sentido anti-horário. */
void GiroAntiHorario() {
    /* Motor A */
    analogWrite(IN1, 0);
    analogWrite(IN2, pot_manobra);
    /* Motor B */
    analogWrite(IN3, pot_manobra);
    analogWrite(IN4, 0);
}
```

```
/* Para o chassi. */
void Pare() {
  /* Motor A */
  analogWrite(IN1, 0);
  analogWrite(IN2, 0);
  /* Motor B */
  analogWrite(IN3, 0);
  analogWrite(IN4, 0);
}

void setup() {
  Serial.begin(9600);
  espSerial.begin(9600);
  /* Inicia com os motores parados. */
  Pare();
  delay(2000);
}

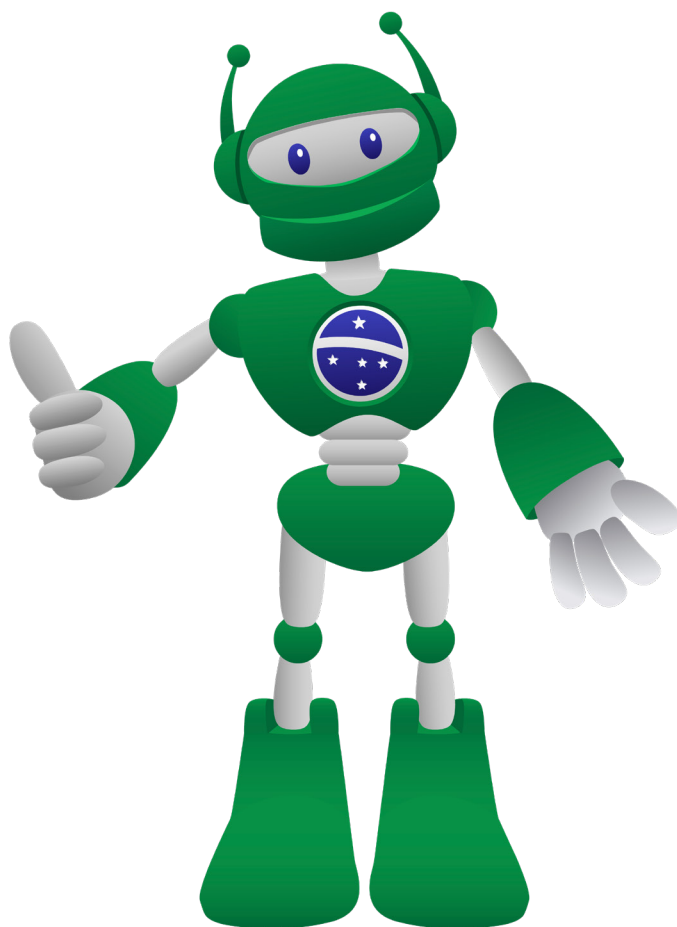
void loop() {
  codigo = '0';
  if (espSerial.available()) {
    codigo = espSerial.read();
  }

  Serial.println(codigo);
  switch (codigo) {
    case '1':
      Frente();
      break;
    case '2':
      GiroAntiHorario();
      break;
    case '3':
      GiroHorario();
      break;
    case '4':
      Re();
      break;
    case '0':
      Pare();
      break;
  }
}
```

Após transferir o programa para o Arduino, desconecte o cabo USB e alimente o seu robô usando a fonte chaveada 9V para realizar os testes iniciais. Após o teste, você poderá alimentá-lo através de bateria ou a case com pilhas através do conector Jack P4 do Arduino (Recomenda-se duas pilhas em série do modelo 18650 por armazenar uma quantidade de carga maior que pilhas convencionais).

Em seguida, utilize um smartphone ou o próprio notebook para se conectar à rede WiFi criada pelo Módulo Wireless ESP01, utilizando o nome e senha definidos no código-fonte do programa.

Após a conexão bem-sucedida, abra um navegador e acesse o controle do seu robô através do endereço **192.168.4.1**, conforme mostrado na figura 10. Agora é só aproveitar a diversão!





Desafios

Que tal utilizar o módulo Wireless para controlar outros protótipos, como o braço robótico? Encare mais este desafio e faça as alterações necessárias nos programas deste projeto!



E se...?

O projeto não funcionar, se atente a alguns dos possíveis erros:

- Verifique se os Jumpers estão nos pinos certos, se estão na mesma coluna dos terminais dos componentes, fazendo assim as conexões;
- Verifique se os Jumpers estão ligados aos pinos corretos no Arduino;
- Reinicie o Módulo Wireless retirando, por alguns segundos, o pino 3,3V do Arduino e reconecte-o;
- Verifique se a programação está adequada a cada porta digital e se as funções foram usadas corretamente;
- Certifique-se de que a fonte de alimentação externa utilizada tenha carga suficiente para manter o robô em funcionamento, se necessário faça a substituição.

3. Feedback e finalização

- a.** Confira, compartilhando seu projeto com os demais colegas, se o objetivo foi alcançado.
- b.** Analise seu projeto desenvolvido, de modo a atender os requisitos para funcionamento do Robô Wireless.
- c.** Reflita se as seguintes situações ocorreram:
 - i. Colaboração e cooperação: Você e os membros de sua equipe interagiram entre si, compartilhando ideias que promoveram a aprendizagem e o desenvolvimento deste projeto?
 - ii. Pensamento Crítico e Resolução de Problemas: Você conseguiu identificar os problemas, analisar informações e tomar decisões de modo a contribuir para o projeto desenvolvido?
- d.** Reúna todos os componentes utilizados nesta aula e os organize novamente junto aos demais, no Kit de Robótica.



Referências

AGUIAR, Márcio. **Precisa-se de robôs inteligentes para manter a produtividade em alta.** Exame. Disponível em: <https://www.uol.com.br/tilt/noticias/afp/2021/01/13/ces-2021-pandemia-disparou-interesse-por-uso-de-robos.htm>. Acesso em: 22 abr. 2022.

ARDUINO. Site oficial. **Ambiente de Programação do Arduino.** Disponível em: <https://create.arduino.cc/editor>. Acesso em: 15 abr. 2022.

ARDUINO. Site oficial. **Documentação de Referência da Linguagem Arduino.** Funções. Disponível em: <https://www.arduino.cc/reference/pt/>. Acesso em: 15 abr. 2022.

BRASIL. **Ministério da Educação. Base Nacional Comum Curricular.** Brasília, 2018. Disponível em: http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518_versaofinal_site.pdf. Acesso em: 18 abr. 2022.

LISBOA, Alveni. **Xiaomi anuncia o CyberDog, um cão-robô inteligente e de código aberto.** Canaltech. Disponível em: <https://canaltech.com.br/robotica/xiaomi-cyberdog-cao-robo-192228/>. Acesso em: 15 abr. 2022.

MCROBERTS, Michael. **Arduino Básico.** 2ª ed. São Paulo: Novatec Editora Ltda. 2015. 506 p. ISBN: 978-85-7522-404-5.

PARANÁ. Secretaria de Estado da Educação e do Esporte. **Robótica Paraná - Aulas.** Robótica Educacional - Módulo 1. Escola Digital Aluno. Disponível em: <http://www.escoladigital.aluno.pr.gov.br/robotica/aulas>. Acesso em: 12 abr. 2022.

TILT. Uol. **CES 2021: Pandemia disparou interesse por uso de robôs.** Disponível em: <https://canaltech.com.br/robotica/xiaomi-cyberdog-cao-robo-192228/>. Acesso em: 15 abr. 2022.

W3SCHOOLS. **Learn to Code.** Disponível em: <https://www.w3schools.com/>. Acesso em: 15 abr. 2022.

DIRETORIA DE TECNOLOGIAS E INOVAÇÃO (DTI)
COORDENAÇÃO DE TECNOLOGIAS EDUCACIONAIS (CTE)

EQUIPE ROBÓTICA PARANÁ

Andrea da Silva Castagini Padilha
Cleiton Rosa
Darice Alessandra Deckmann Zanardini
Edgar Cavalli Júnior
Edna do Rocio Becker
José Feuser Meurer
Marcelo Gasparin
Michele Serpe Fernandes
Michelle dos Santos
Orlando de Macedo Júnior
Roberto Carlos Rodrigues

Os materiais, aulas e projetos da “Robótica Paraná” foram produzidos pela Coordenação de Tecnologias Educacionais (CTE), da Diretoria de Tecnologia e Inovação (DTI), da Secretaria de Estado da Educação do Paraná (SEED), com o objetivo de subsidiar as práticas docentes com os estudantes por meio da Robótica. Este material foi produzido para uso didático-pedagógico exclusivo em sala de aula.



Este trabalho está licenciado com uma Licença
Creative Commons – CC BY-NC-SA
[Atribuição - NãoComercial - Compartilha Igual 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

