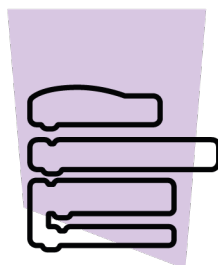
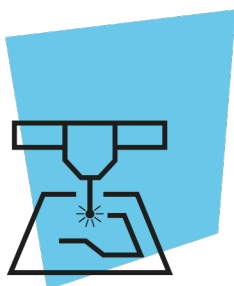
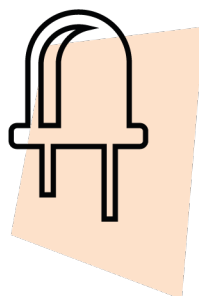
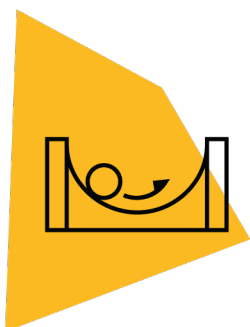
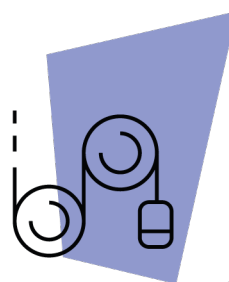
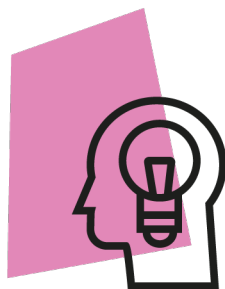


ROBÓTICA

Módulo 2



Robô Wireless

AULA 39

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

Revisão Textual

Adilson Carlos Batista

Validação de Conteúdo

Cleiton Rosa

Leitura Crítica e Normalização Bibliográfica

Ricardo Hasper

Projeto Gráfico e Diagramação

Edna do Rocio Becker

Ilustração

Jocelin Vianna (Educa Play)

2022

Aula 01
O que já vimos?

Aula 02
Arduino:
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

Aula 40
Braço Robótico Via RF
(Rádio Frequência)

Sumário

Introdução	2
Objetivos desta Aula	2
Competências 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	8
3. Feedback e Finalização	42
Videotutorial	43
Referências	44



Introdução

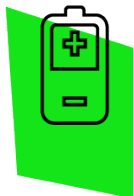
As tecnologias da Internet das coisas (IoT) vêm mudando a forma como interagimos com a casa e objetos. Casas e cidades inteligentes são termos que já começam a tomar forma e fazer parte de nossa realidade. Um dos maiores benefícios desta aplicação é poder controlar e acompanhar várias “coisas”, de onde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

39 Robô Wireless



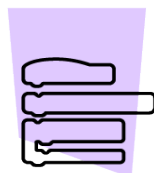
Habilidades do Século XXI a Serem Desenvolvidas

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



Lista de Materiais

- 01 Placa Arduino Uno R3;
- 01 Cabo USB;
- 01 Motor Shield L293D Driver Ponte H para Arduino;
- 01 Módulo Wireless ESP01;
- 01 Kit Chassi 2WD Robô para Arduino;
- 01 Resistor 1 K Ω ;
- 01 Resistor 2,2 K Ω ;
- 06 Jumpers Fêmea-Fêmea;
- 08 Jumpers Macho-Macho;
- 01 Bateria 9V ou compartimento com baterias;
- 01 Chave de fenda pequena;
- 01 Fita adesiva para fixar o módulo Wireless ESP01;
- Notebook;
- Software Arduino IDE.



Roteiro da Aula

1. Contextualização (15min):

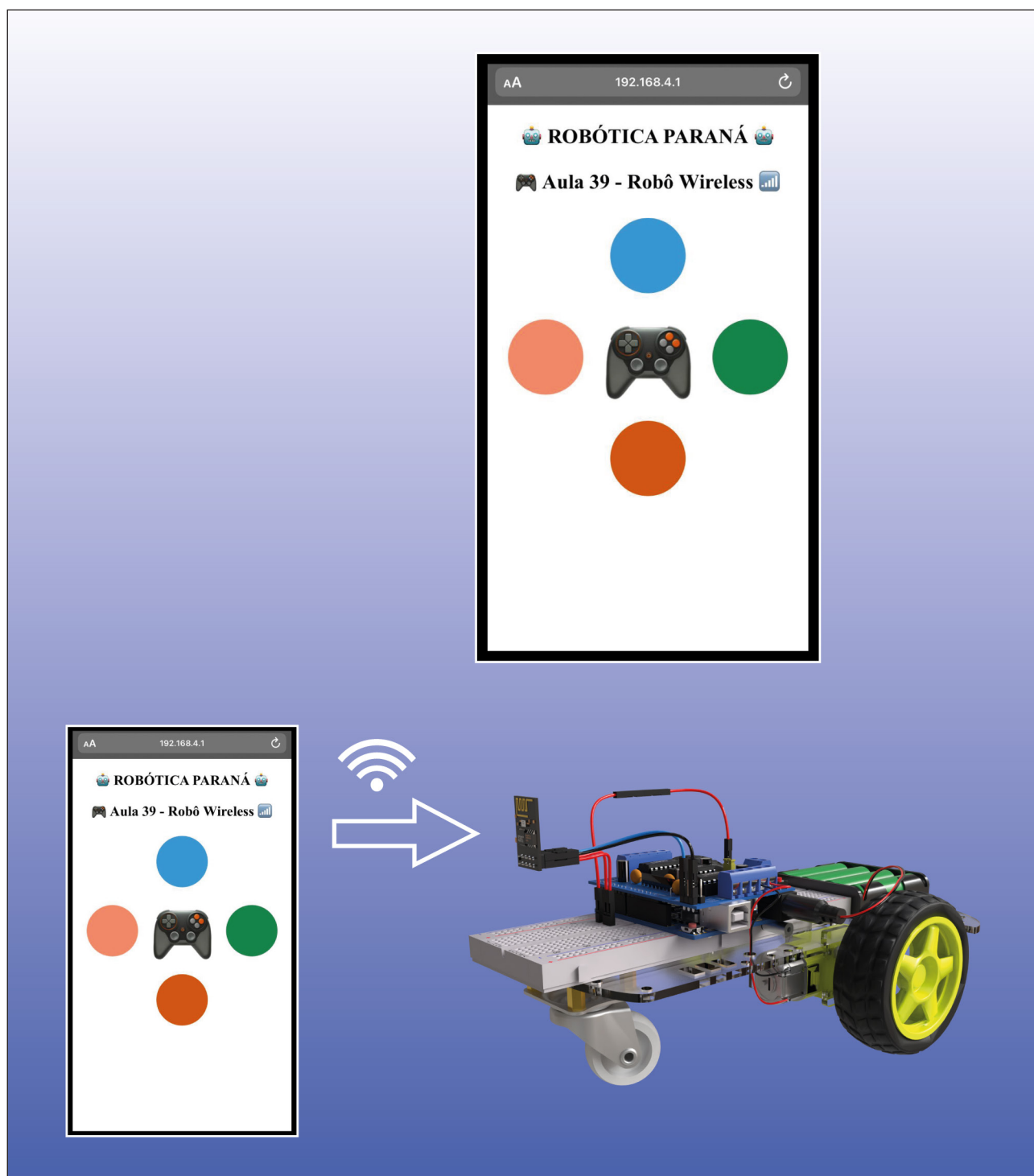
Em meio aos desafios da atualidade, incluindo mudanças no âmbito econômico, sanitário e global, a automação nunca teve uma importância e urgência tão grande diante da necessidade de cumprir protocolos e de um novo comportamento imposto pela recente crise mundial de saúde causado pelo novo coronavírus.

Além do mais, para preservar a vida das pessoas, tanto em locais públicos como em partes internas as empresas e comércios, foi preciso implementar sistemas de tecnologias com software e robôs que supram a mão de obra com o objetivo de manter o distanciamento social.

Nesse sentido, a pandemia com o COVID 19 impulsionou a tecnologia, acelerando a transformação digital e o desenvolvimento de robôs mais acessíveis em praticamente todas as áreas da sociedade. Diante de tal importância, que tal aprender a programar e controlar um robô através de um celular?

Nesta aula, você aprenderá a controlar remotamente um robô autônomo (Kit Chassi 2WD) por meio de uma página HTML gerada para trabalhar como servidor Web. Acessando a página Web por meio de um dispositivo (Tablet, Notebook ou Celular) você poderá conduzir o robô através das teclas do controle (figura 1).

Figura 1 - Teclas de controle do robô.



Fonte: SEED / DTI / CTE, 2024



Para Saber Mais...

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/>

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



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

Pandemia disparou interesse por robôs.



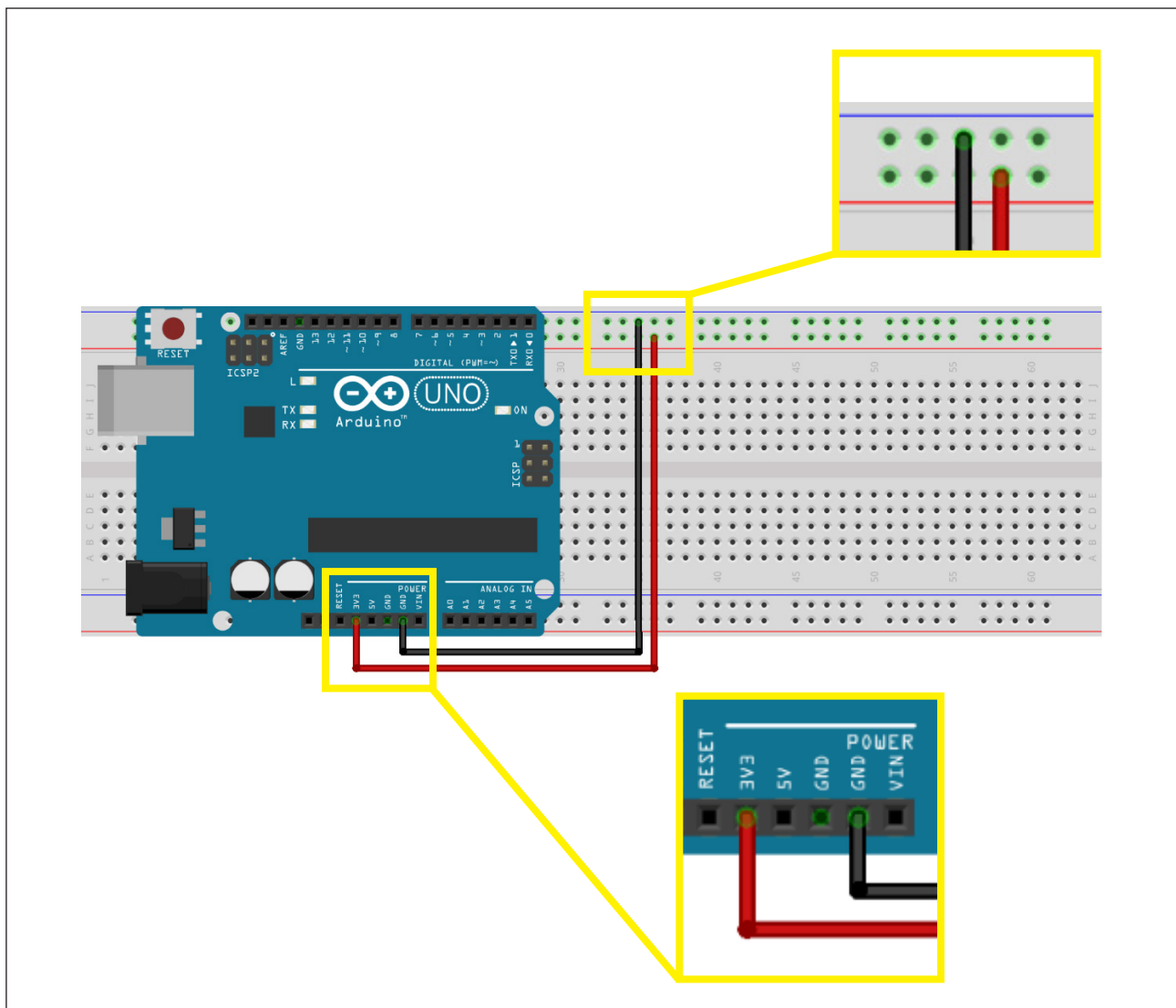
<https://www.uol.com.br/tilt/noticias/afp/2021/01/13/ces-2021-pandemia-disparou-interesse-por-uso-de-robos.htm>



2. Montagem e Programação (60min):

Vamos iniciar a montagem preparando 6 Jumpers Macho-Fêmea, unindo 6 Jumpers Macho-Macho a outros 6 Jumpers Fêmea-Fêmea e reserve-os. Agora, utilizando 2 Jumpers Macho-Macho, conecte o pino **3,3V** do Arduino à linha lateral VERMELHA da Protoboard e o pino **GND** do Arduino à linha lateral AZUL da Protoboard, como mostrado na figura 2.

Figura 2 – Alimentação da Protoboard

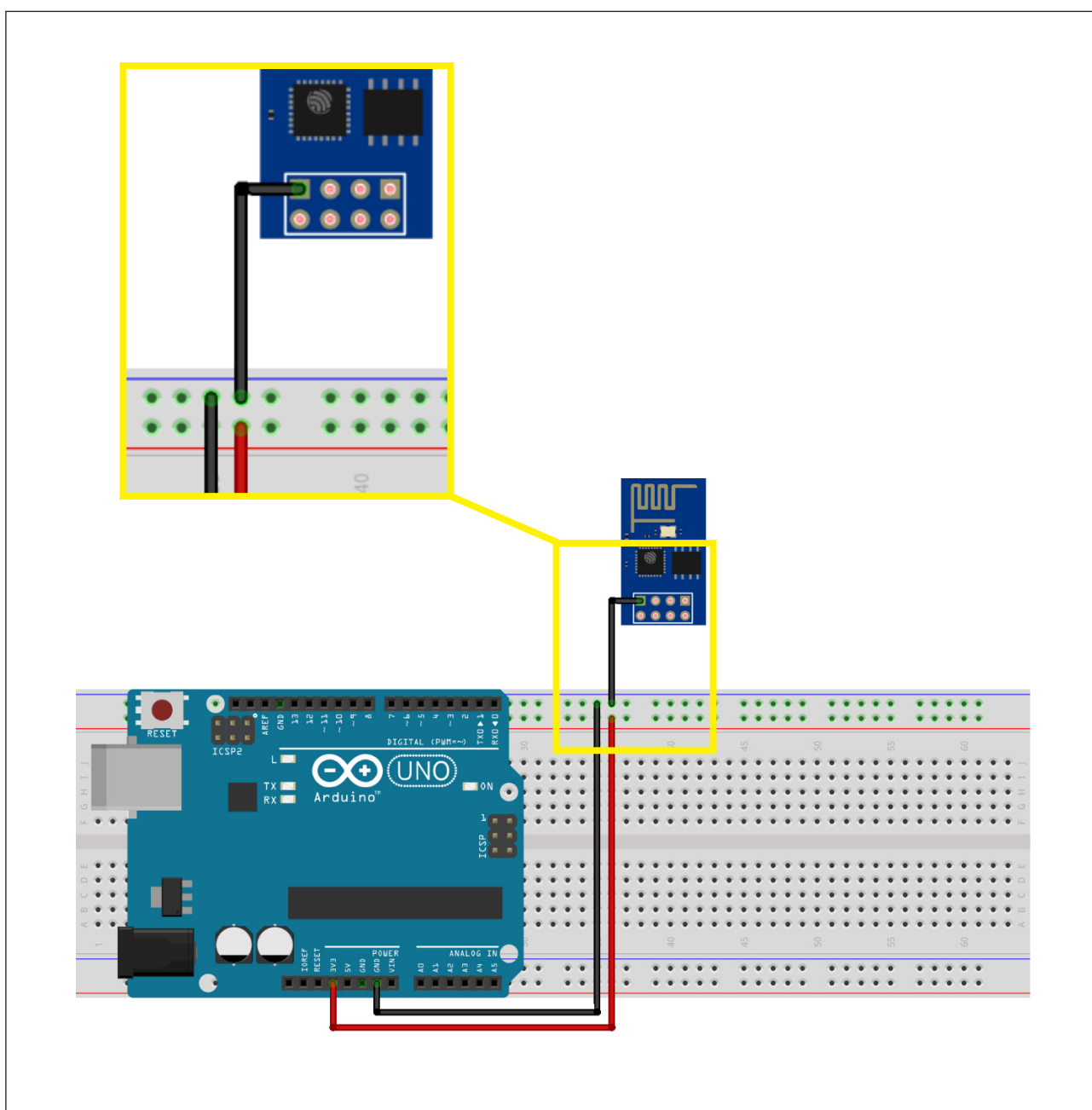


Fonte: Fritzing, 2022

39 Robô Wireless

Utilizando 1 Jumper Macho-Fêmea, interligue o pino **GND** do Módulo Wireless à linha lateral AZUL da Protoboard, como mostrado na figura 3.

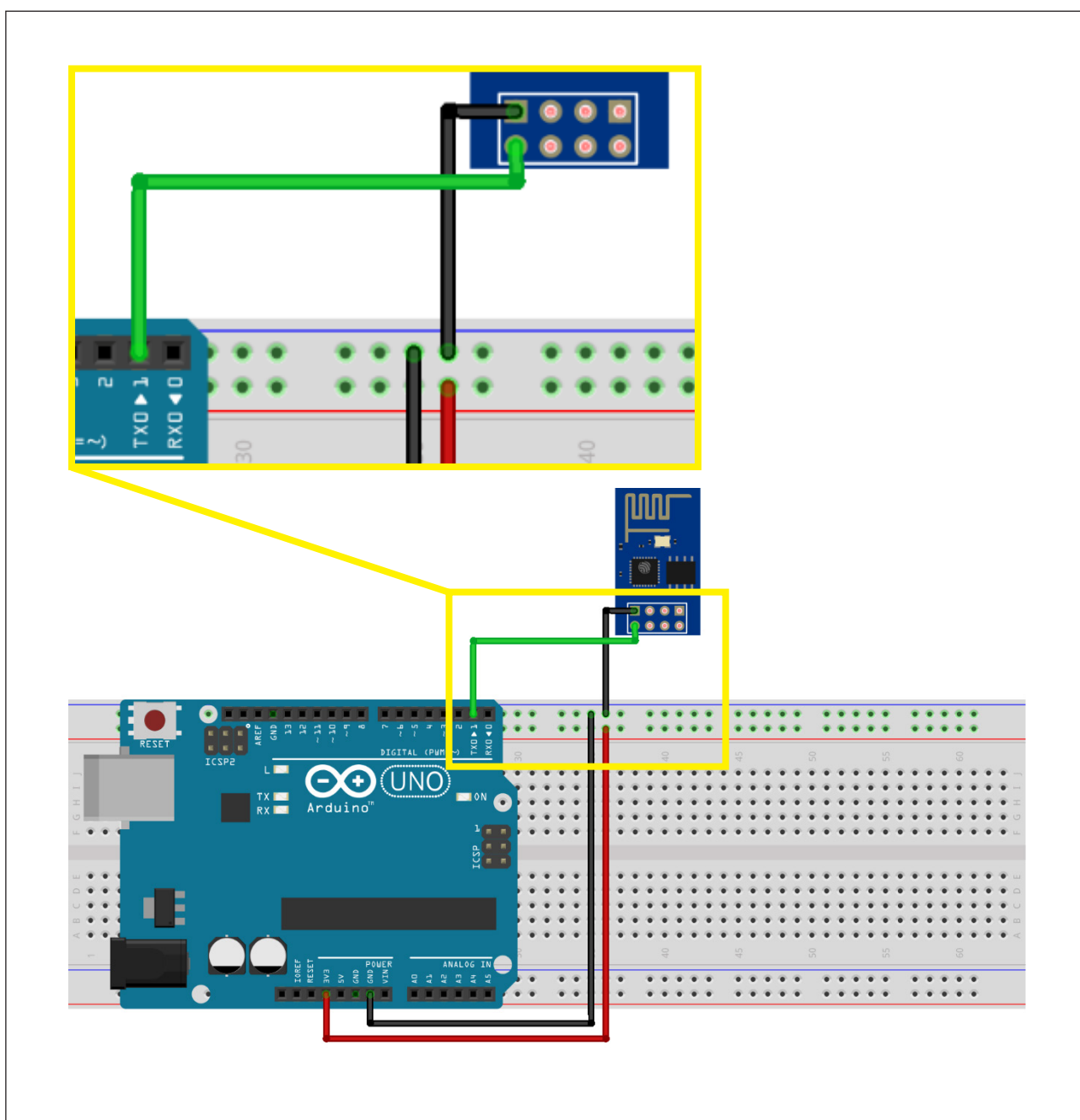
Figura 3 - Conexão do pino **GND** do Módulo Wireless à Protoboard



39 Robô Wireless

Através de outro Jumper Macho-Fêmea, interligue o pino **TX** do Módulo Wireless à porta **digital 1** do Arduino, como mostrado na figura 4.

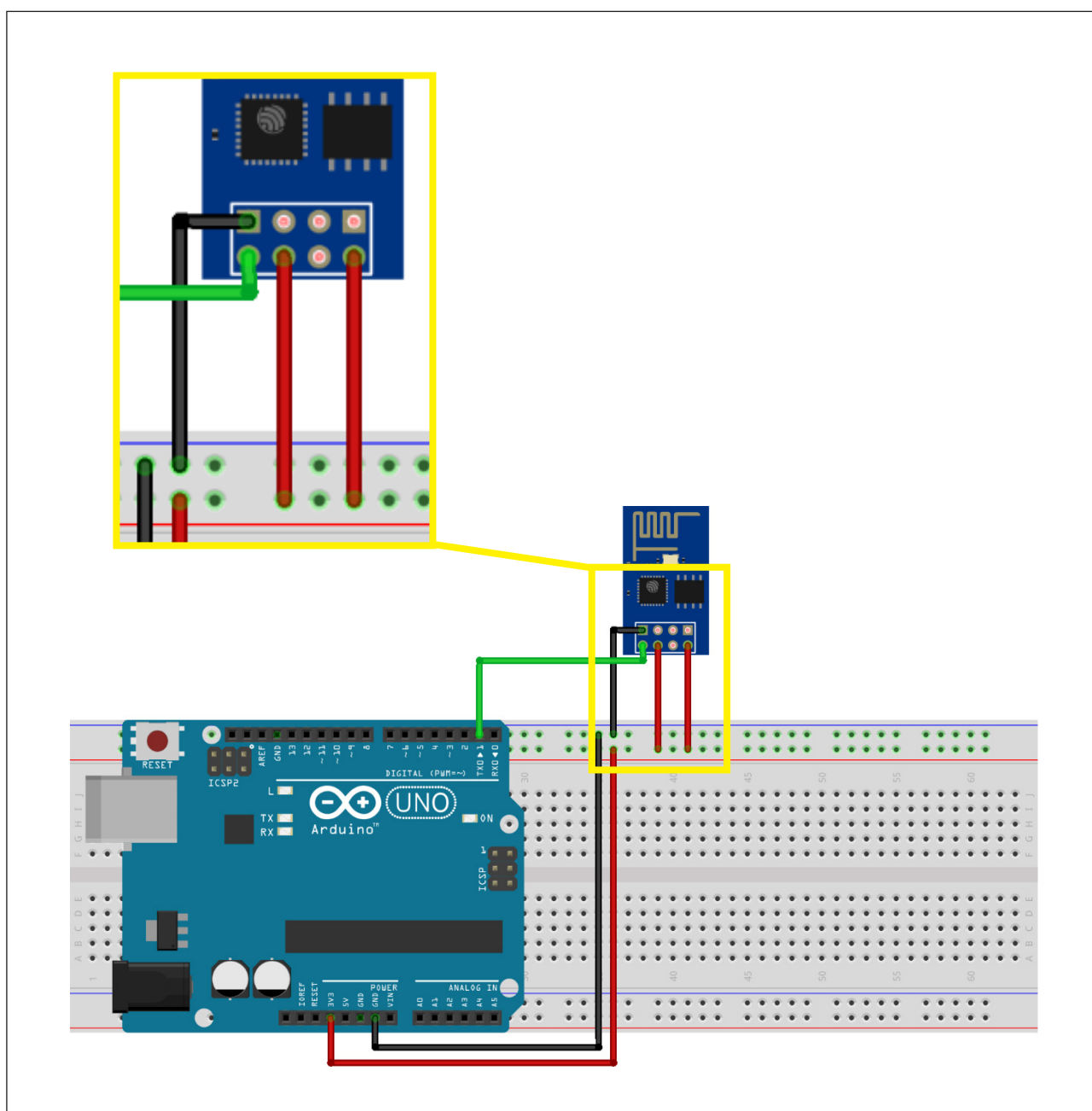
Figura 4 - Conexão do pino **TX** do Módulo Wireless ao Arduino



39 Robô Wireless

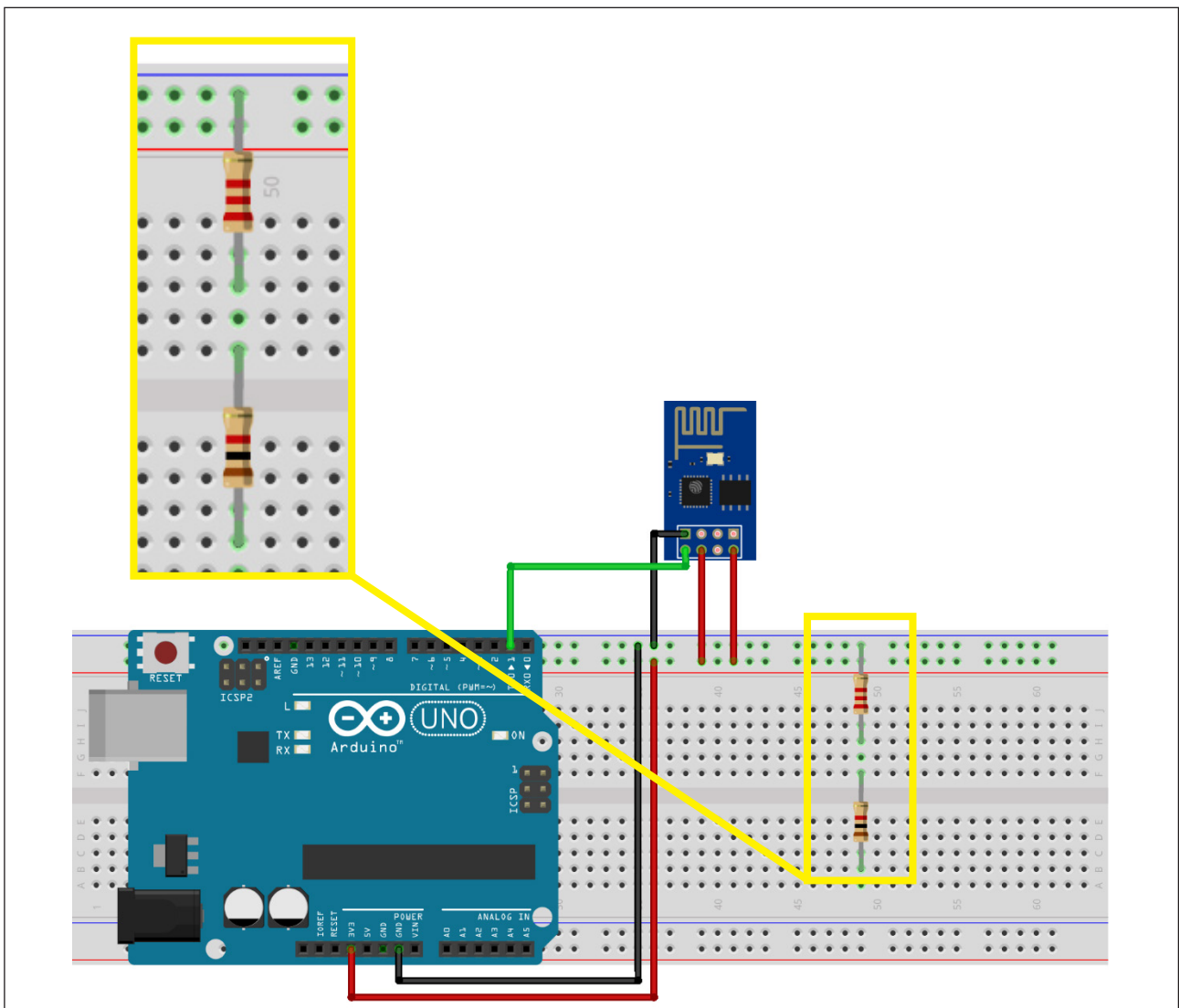
Através de mais 2 Jumpers Macho-Fêmea, interligue os pinos **CH_PD** e **3,3V** do Módulo Wireless à linha lateral VERMELHA da Proto-board, como mostrado na figura 5.

Figura 5 - Conexão dos pinos **CH_PD** e **3.3V** do Módulo Wireless à Protoboard



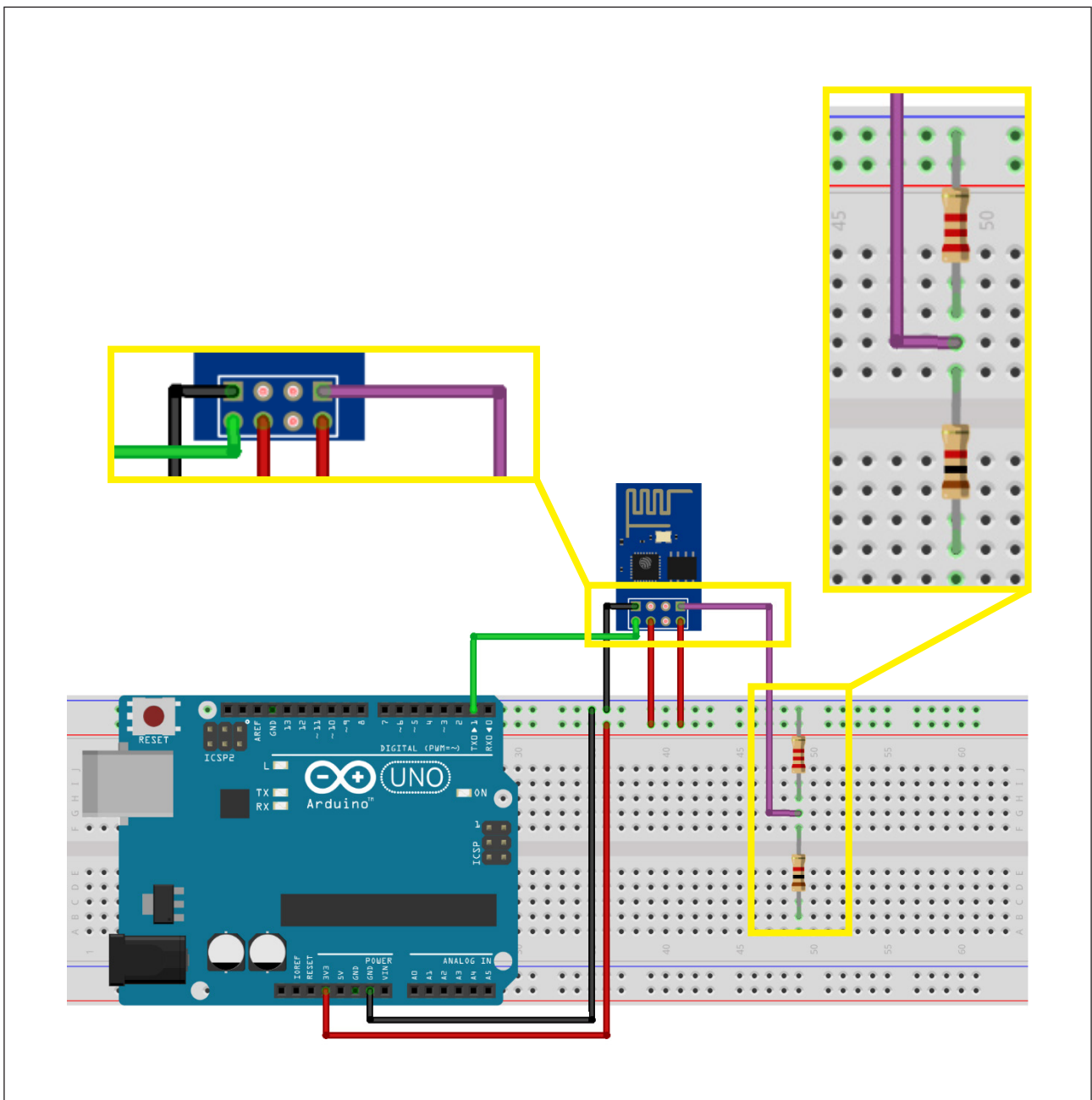
Utilizando os dois Resistores, monte o divisor de tensão para o pino RX do Módulo Wireless da seguinte forma: insira um dos terminais do Resistor de **2,2K Ω** na linha lateral AZUL da Protoboard e o outro terminal numa coluna de furos da região central superior. Insira um dos terminais do Resistor de **1K Ω** na mesma coluna de furos da região central superior do primeiro Resistor e o outro terminal, numa coluna de furos da região central inferior da Protoboard, como mostrado na figura 6.

Figura 6 – Montagem do divisor de tensão



Com outro Jumper Macho-Fêmea, conecte o pino **RX** do Módulo Wireless ao divisor de tensão, na mesma coluna de furos em que os dois Resistores estão interligados, como mostrado na figura 7.

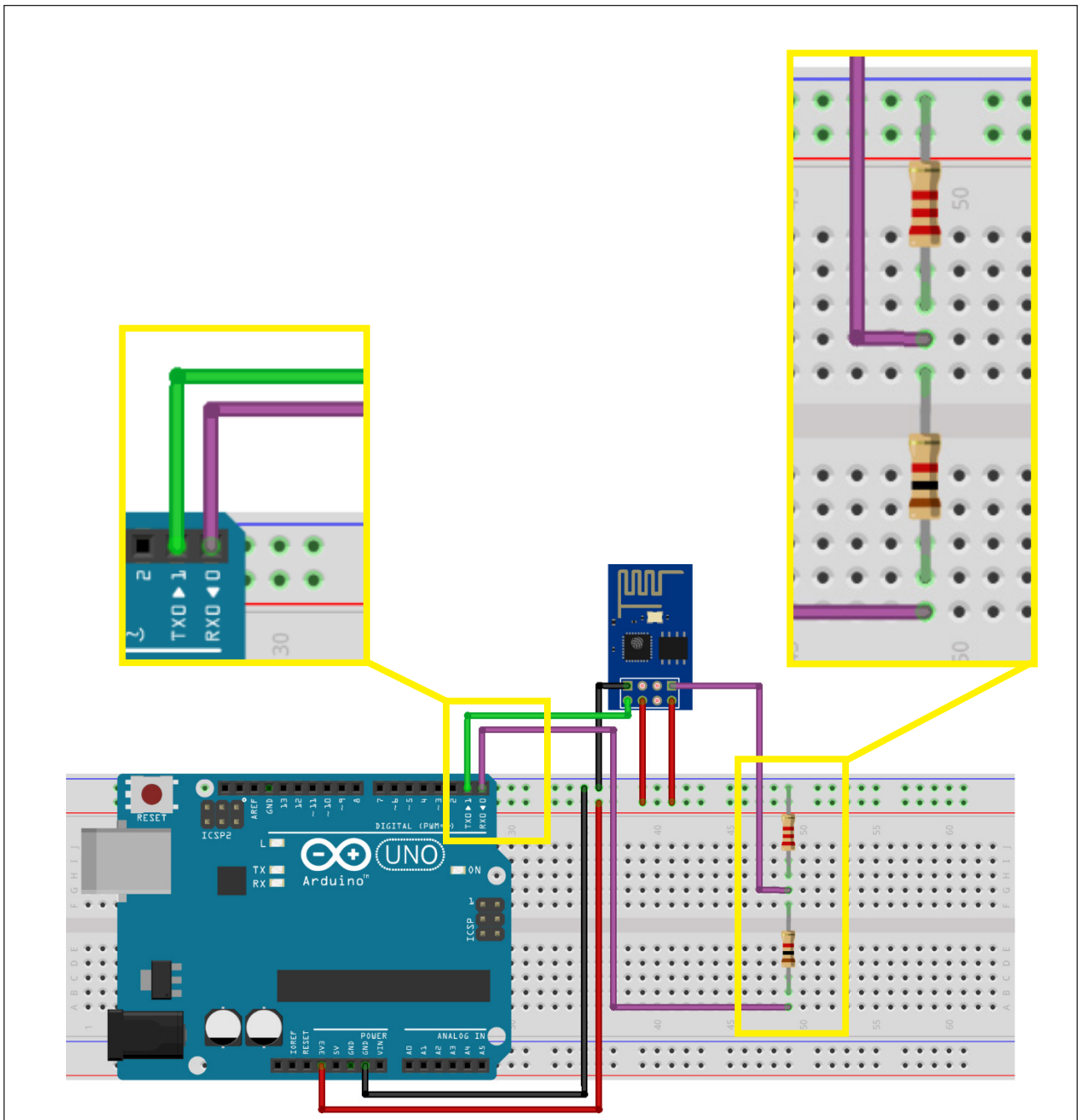
Figura 7 – Conexão do pino **RX** do Módulo Wireless ao divisor de tensão



Fonte: Fritzing

Conecte, através de 1 Jumper Macho-Macho, o divisor de tensão à **porta digital 0** do Arduino, como mostrado na figura 8.

Figura 8 - Conexão do divisor de tensão à porta **digital 0** do Arduino

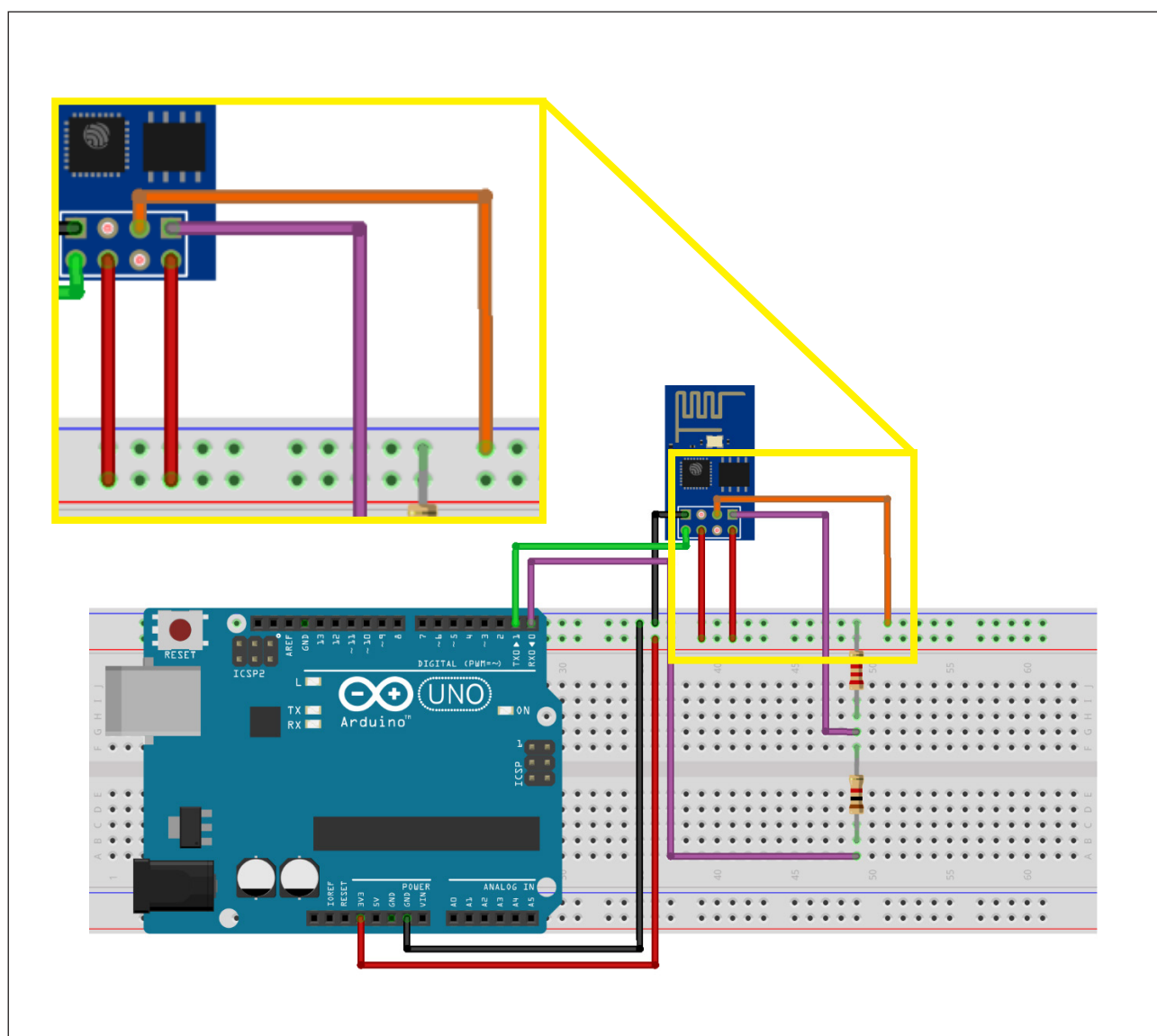


39 Robô Wireless

Com outro Jumper Macho-Fêmea, conecte o pino **GPIO0** do Módulo Wireless à linha lateral AZUL da Protoboard, como mostrado na figura 9.

Observação: Esse Jumper é utilizado somente para a gravação do programa. Após, será removido para o normal funcionamento do Módulo Wireless.

Figura 9 – Conexão do pino **GPIO0** do Módulo Wireless à Protoboard



Fonte: Fritzing, 2022



Agora, vamos programar o Módulo Wireless!

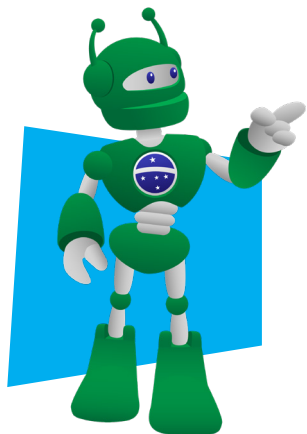
Antes de programarmos o Módulo Wireless, precisamos garantir que o conversor USB serial do Arduino não esteja sendo utilizado por códigos anteriormente gravados, basta carregarmos um Sketch com as funções **setup()** e **loop()** vazias.

Com o Arduino conectado ao Notebook, no Software Arduino IDE, selecione a **Placa: Arduino UNO** e a **porta de comunicação COM** que o Notebook atribuiu ao Arduino através dos menus **Ferramentas > Placa** e **Ferramentas > Porta** e carregue o código mostrado no quadro 1 em seu Arduino e prossiga.

Quadro 1 - Código-fonte para limpar a memória do Arduino

```
void setup() {  
  }  
void loop() {  
}
```

Feito isso, novamente 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** instalada, 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, conforme apresentado no quadro 2.



Atenção, no campo "**ALTERAR**", na linha 16 do código, você deverá alterar o nome da rede WiFi 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 a sua rede, lembrando que a senha precisa ter no mínimo 8 caracteres.

Quadro 2 - 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, girar no */
/* sentido anti-horário e parar. */
/* **** */
/* Inclui as bibliotecas do Módulo WiFi. */
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>

/* Define o nome e senha da rede wifi do robô. */
const char *nome_da_rede = "ALTERAR";
const char *senha = "12345678";

/* Inicia o servidor na porta padrão 80. */
ESP8266WebServer servidor(80);
```

```

/* Variável que armazenará o código html da página 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-scale=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;  
    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: #ffff;
}

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

#botao4 {
    background-color: #d35400;
    color: #ffff;
}

#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">ROBÓTICA PARANÁ</h2>
    <h2 class="prevent-select">Aula 39 - Robô Wireless</h2>

    <table class="joystick-table">
```

```

<tr>
  <td class="joystick-cell"></td>
  <td class="joystick-cell"><button class="prevent-select
joystick-button" id="botao1"></button></td>
  <td class="joystick-cell"></td>
</tr>
<tr>
  <td class="joystick-cell"><button class="prevent-select
joystick-button" 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-button" id="botao3"></button></td>
</tr>
<tr>
  <td class="joystick-cell"></td>
  <td class="joystick-cell"><button class="prevent-select
joystick-button" 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;
  let botaoAtivo = null; // Variável para controlar o botão pressionado

  function iniciarAcao(botao, comando, corPressionado, corSolto) {
    if (botaoAtivo === null) {
      botaoAtivo = botao; // Define o botão ativo
      botao.style.backgroundColor = corPressionado;
    }
  }

```

```
        botao.style.transform = 'translateY(2px)';
        botao.style.boxShadow = 'none';
        intervalID = setInterval(() => {
            fetch('/pressionado', { method: 'POST', body: comando });
        }, 100);
    }
}

function pararAcao(botao, corSolto) {
    if (botaoAtivo === botao) { // Só libera se for o botão ativo
        botao.style.backgroundColor = corSolto;
        botao.style.transform = 'translateY(0)';
        botao.style.boxShadow = 'none';
        clearInterval(intervalID);
        fetch('/pressionado', { method: 'POST', body: '0' }); // Envia o
comando de parada
        botaoAtivo = null; // Libera para que outro botão possa ser
pressionado
    }
}

// Funções de eventos para cada botão
botao1.addEventListener('touchstart', () => iniciarAcao(botao1, '1',
'#2980b9', '#3498db'));
botao1.addEventListener('touchend', () => pararAcao(botao1, '#3498db'));

botao2.addEventListener('touchstart', () => iniciarAcao(botao2, '2',
'#ff8c66', '#ff5733'));
botao2.addEventListener('touchend', () => pararAcao(botao2, '#ff5733'));

botao3.addEventListener('touchstart', () => iniciarAcao(botao3, '3',
'#1d8348', '#27ae60'));
botao3.addEventListener('touchend', () => pararAcao(botao3, '#27ae60'));
```

```
        botao4.addEventListener('touchstart', () => iniciarAcao(botao4, '4',
        '#d35400', '#f39c12'));
        botao4.addEventListener('touchend', () => pararAcao(botao4, '#f39c12'));
    </script>

</body>
</html>

)===";

void setup() {
    /* Cria o ponto de acesso com o nome e senha definida. */
    WiFi.softAP(nome_da_rede, senha);
    /* Requisição entregue ao cliente que acessar o endereço. */
    servidor.on("/", []() {
        servidor.send(200, "text/html", pagina);
    });
    /* Requisição dos botões pressionados. */
    servidor.on("/pressionado", HTTP_POST, []() {
        String estado = servidor.arg("plain");
        Serial.println(estado);
        servidor.send(200, "text/plain", "OK");
    });
    /* Inicia o servidor. */
    servidor.begin();
    /* Inicia a comunicação serial. */
    Serial.begin(9600);
}

void loop() {
    /* função que monitora os clientes conectados entregando as requisições
    solicitadas. */
    servidor.handleClient();
}
```

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 10.

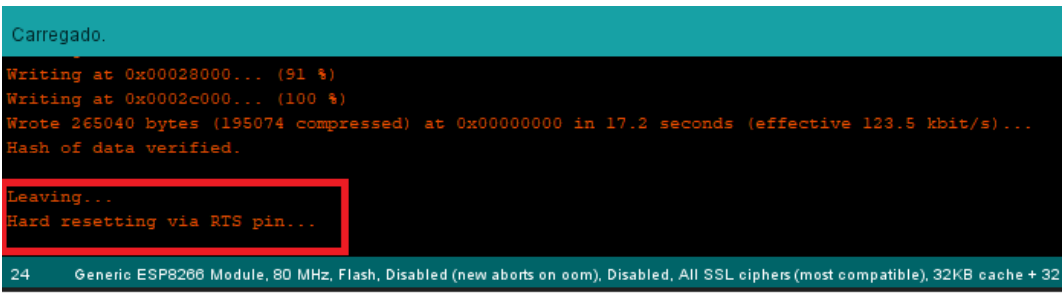
Figura 10 – Gravação do código ao Módulo Wireless



```
Carregando...
Running stub...
Stub running...
Configuring flash size...
Auto-detected Flash size: 1MB
Compressed 265040 bytes to 195074...
Writing at 0x00000000... (8 %)
Writing at 0x00004000... (16 %)
24 Generic ESP8266 Module, 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB SRAM
```

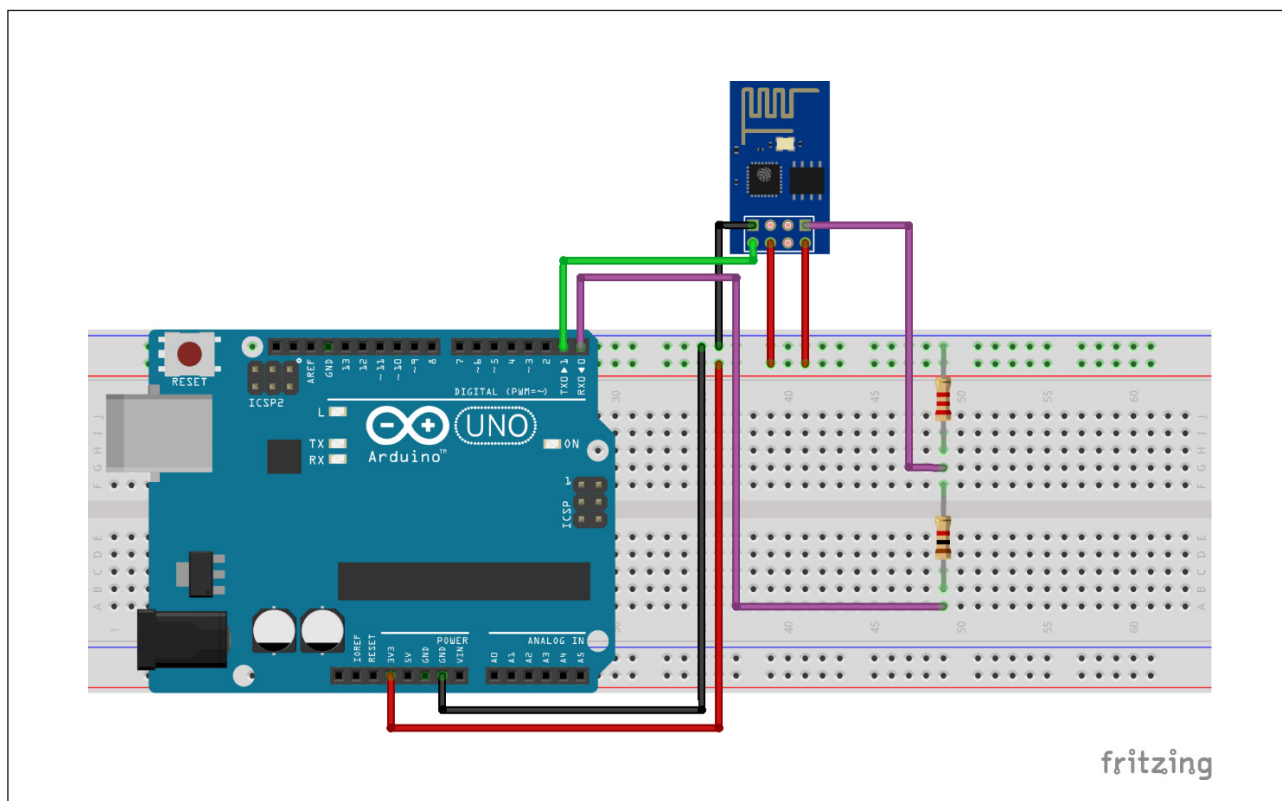
Ao terminar o processo de gravação, será apresentada a mensagem **Hard resetting via RTS pin...**, (figura 11).

Figura 11 – Término da gravação do código ao Módulo 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 + 32KB SRAM
```

A seguir, reiniciamos o Módulo Wireless removendo o cabo USB do computador ou do Arduino e também o Jumper conectado entre o pino **GPI00** do Módulo e a linha lateral **AZUL** da Protoboard (figura 12).

Figura 12 – Jumper removido do Pino **GPIO0** do Módulo Wireless

Fonte: Fritzing, 2022

Aproximadamente 30 segundos após reiniciar o Módulo Wireless, você poderá observar através de um dispositivo (Smartphone, Notebook, Tablet etc.) que foi criada uma rede WI-FI com o nome que você escolheu antes de carregar o código. Conecte à sua rede inserindo a senha “12345678”, caso não tenha feito a alteração no código.

Após a conexão, abra o navegador de seu dispositivo e acesse o endereço de IP padrão do Módulo (<http://192.168.4.1>), com isso, abrirá a página criada no código-fonte da programação (figura 13).

Figura 13 – Acesso da página HTML criada no Módulo Wireless



Fonte: SEED / DTI / CTE, 2022

Dicas:

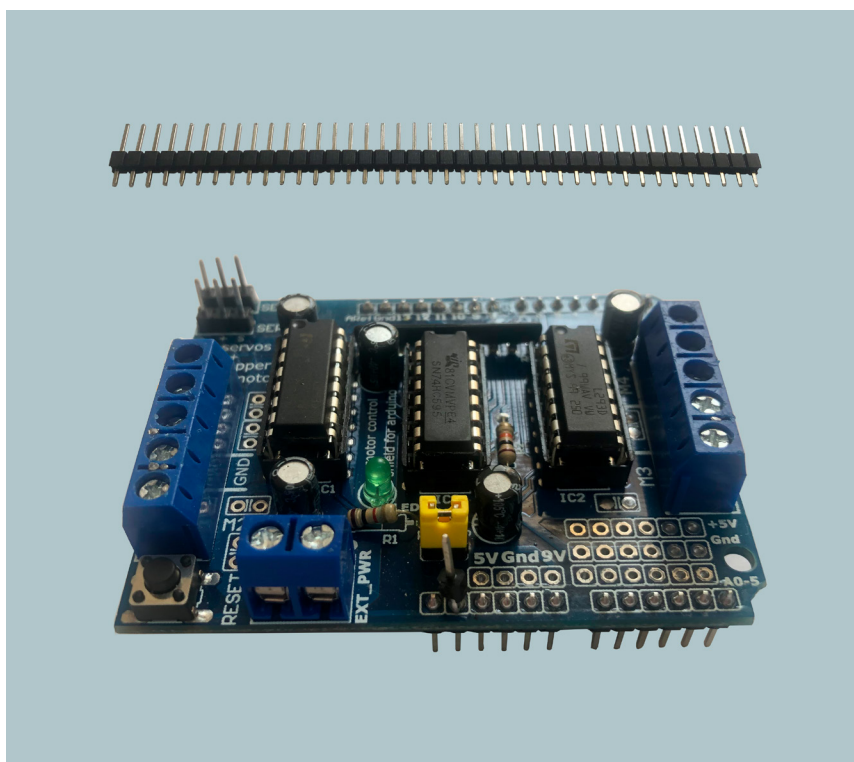
- Para criar suas páginas HTML acesse o site w3school através do link: <https://www.w3schools.com/>
- Para converter a página HTML na linguagem que o Software Arduino IDE compreenda (C/C++), utilize a ferramenta online através do link http://tomeko.net/online_tools/cpp_text_escape.php?lang=en

Realizada esta primeira etapa, já podemos desmontar o protótipo utilizado para gravação do Módulo Wireless e iniciar a montagem do nosso robô Wireless.

Antes de iniciarmos a montagem do nosso robô Wireless, certifique-se de que o Kit Chassi 2WD esteja corretamente montado, ou seja, os motores e as rodas, fixados à estrutura de acrílico. Caso precise montar ou fazer alguns ajustes, você poderá consultar o guia de montagem do KIT Chassi 2WD através do link: [Manual de montagem do kit chassi 2WD](#)

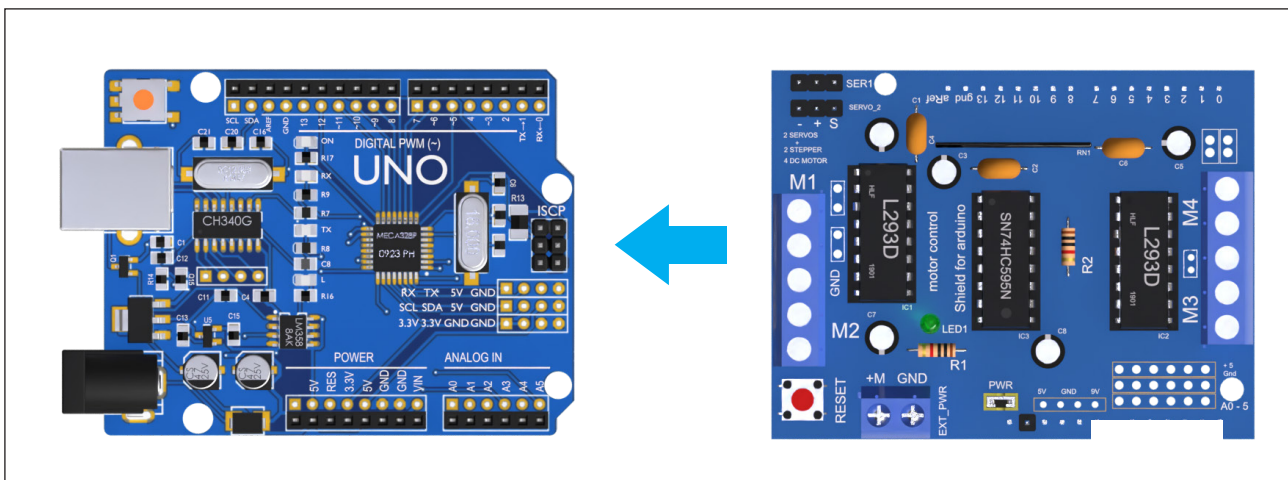
Precisamos também soldar um pino na placa Motor Shield para que possamos ter acesso ao pino 3,3V do Arduino. Utilizando um terminal da barra de pinos (figura 18), que acompanha o pacote com a placa Arduino, faça a solda no 2º pino da esquerda para a direita da parte inferior da Motor Shield, como indicado na figura 14.

Figura 14 - Barra de pinos e soldagem na placa Arduino



Agora, estamos prontos para dar início à montagem do robô Wireless. Comece preparando 3 Jumpers Macho-Fêmea, para isso, encaixe 3 Jumpers Macho-Macho a 3 Jumpers Fêmea-Fêmea. A seguir, encaixe a placa Motor Shield L293D sobre a placa Arduino Uno R3, cuidando para o correto encaixe dos pinos e posicione o conjunto de placas sobre uma Placa Protoboard, conforme a figura 15.

Figura 15: Encaixe da placa Motor Shield L293D sobre o Arduino



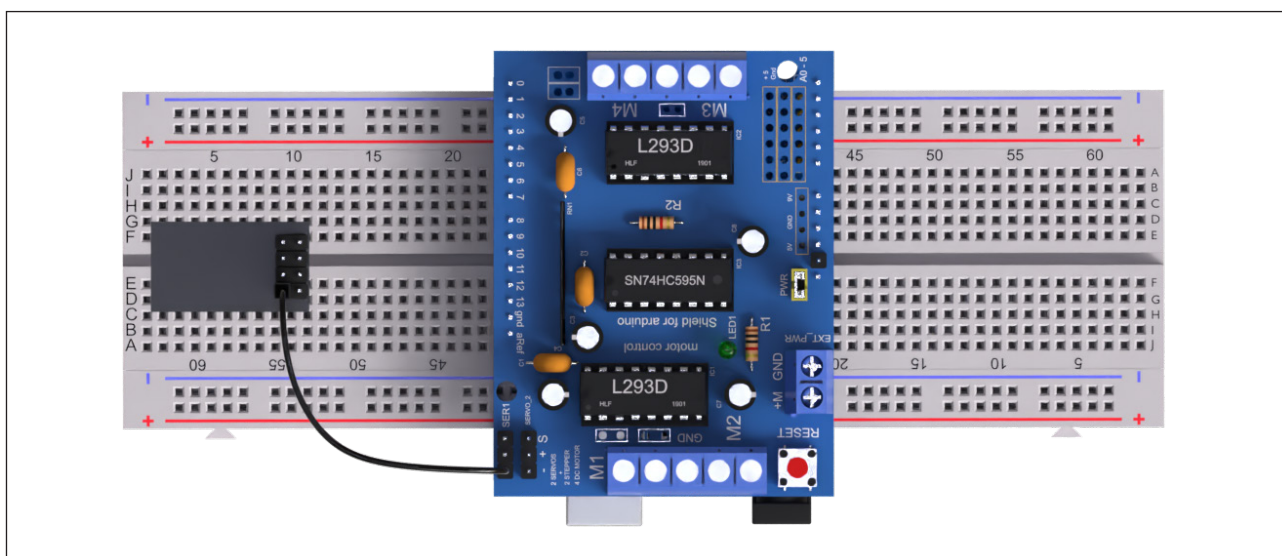
Fonte: Fritzing, 2022



39 Robô Wireless

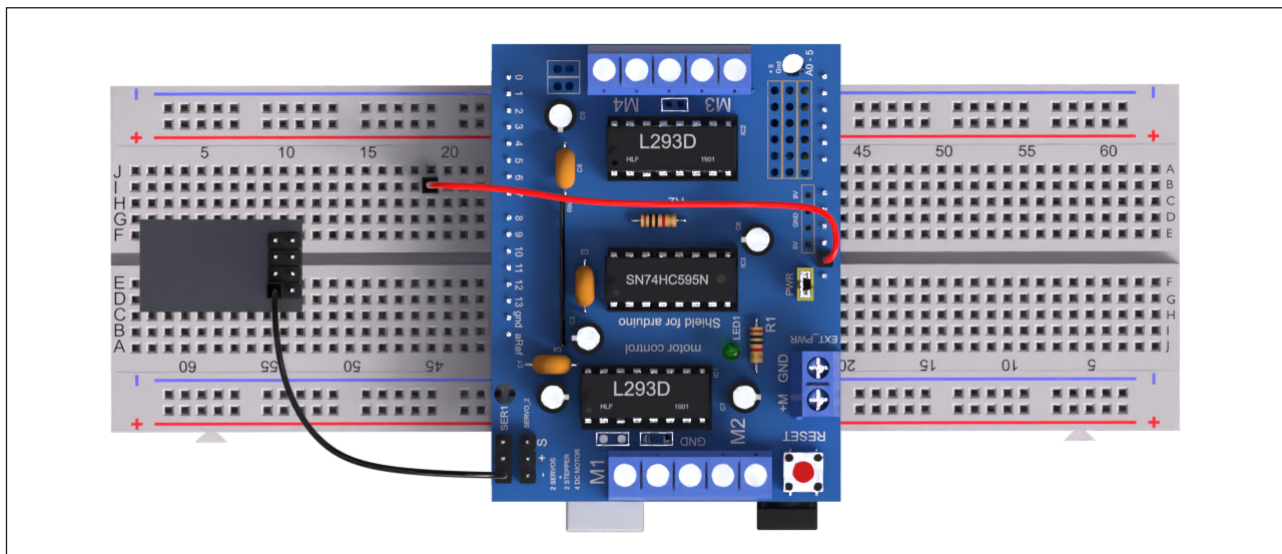
Com um Jumper Fêmea-Fêmea conecte o pino GND do módulo Wireless a um dos pinos negativos (símbolo -) da Motor Shield, conforme mostrado na figura 16.

Figura 16 - Conexão do pino GND do módulo Wireless à Placa Motor Shield



Utilize um Jumper Macho-Fêmea para conectar o **pino 3,3V** da Motor Shield (aquele soldado anteriormente) a uma coluna de furos da placa Protoboard, como indicado na figura 17.

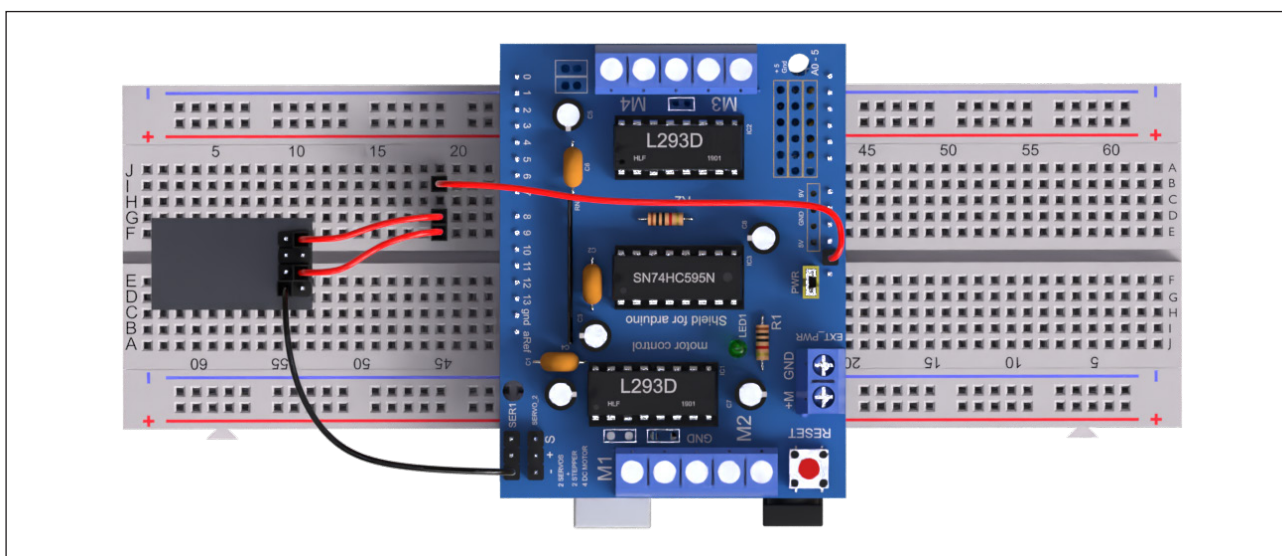
Figura 17 - Conexão do pino 3,3V Placa Motor Shield à Protoboard



39 Robô Wireless

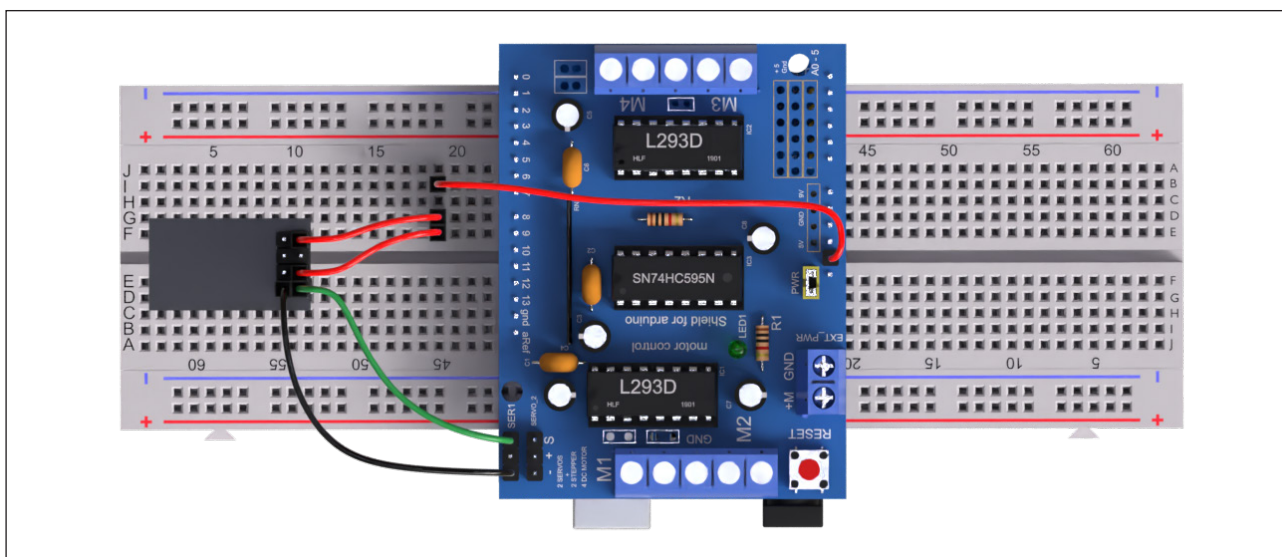
Utilize mais dois Jumpers Macho-Fêmea para conectar os pinos **CH_PD** e **3,3V** do módulo Wireless à mesma coluna de furos que foi conectado o Jumper anterior (**pino 3,3V**), como indicado na figura 18.

Figura 18 - Conexão dos pinos CH_PD e 3,3V do Módulo Wireless à Protoboard



Utilize um Jumper Fêmea-Fêmea para conectar o pino **GPIO1** do módulo ao primeiro **pino S** da Motor Shield, conforme a figura 19.

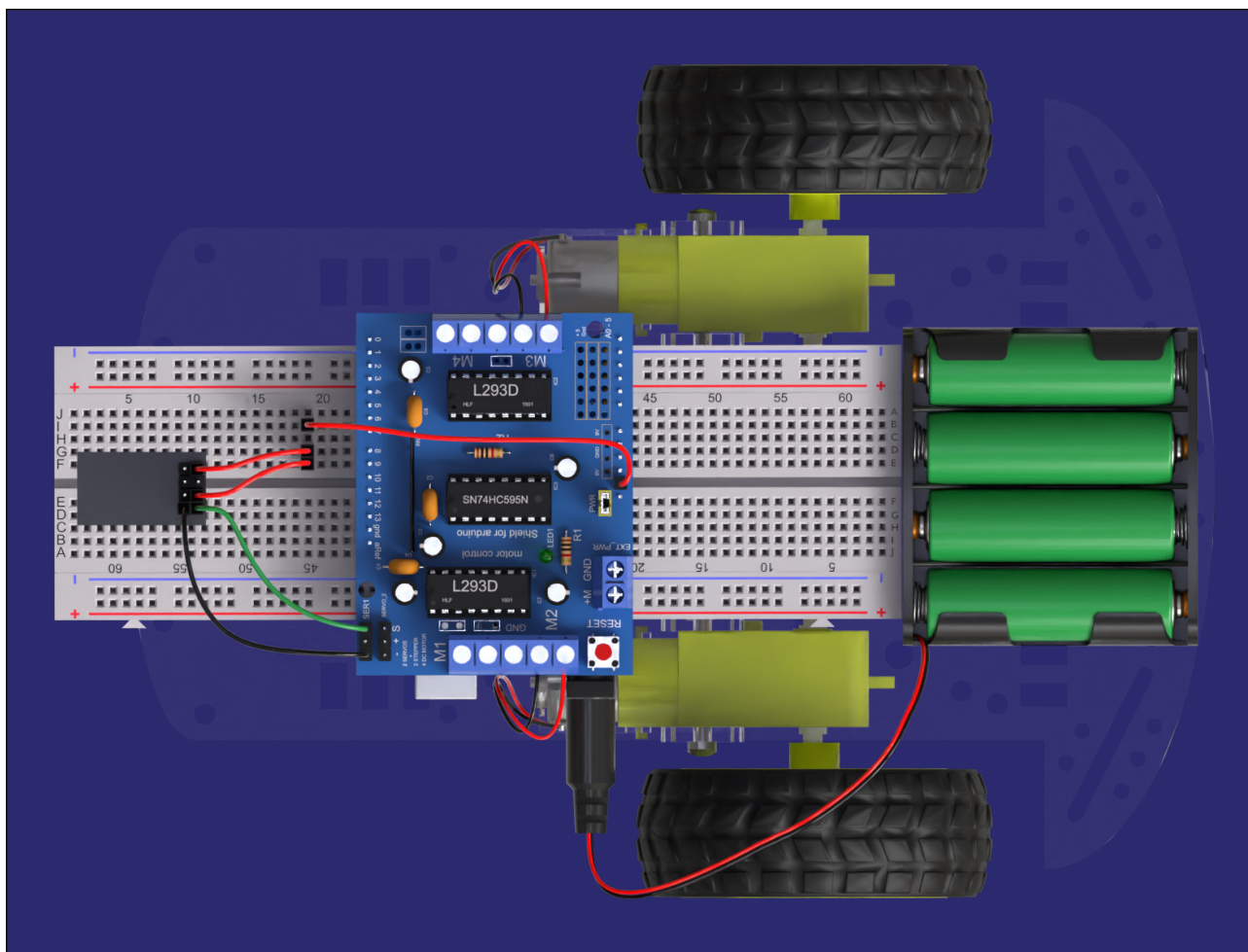
Figura 19 - Conexão do pino GPIO1 do Módulo Wireless ao pino S da Motor Shield



39 Robô Wireless

Conecte a Bateria ao Arduino através do conector Jack P4, conforme a figura 20.

Figura 20 - Conexão da Bateria ao Arduino

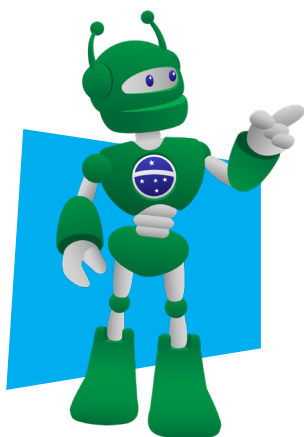


Fonte: Fritzing, 2022



Agora, vamos programar o Arduino!

Antes de programarmos o Arduino, precisamos alterar a configuração da placa no Software Arduino IDE. Com o Arduino conectado ao Notebook, selecione no menu **Ferramentas > Placa: "Generic ESP8266 Module" > Arduino AVR Boards > Arduino Uno** e a porta de comunicação COM que o Notebook atribuiu ao Arduino através do menu **Ferramentas > Porta > COMx**. Então, carregue o código mostrado no quadro 3 em seu Arduino e prossiga.



Atenção! Para esta programação é necessário a biblioteca **"AFMotor.h"**, instalada no Software Arduino IDE. Na **Aula 05 - Softwares Arduino IDE e mBlock** você encontrará orientações sobre como realizar a instalação de bibliotecas.

Quadro 3 - Código-fonte para limpar a memória do Arduino

```
/* **** */
/* 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, girar no sentido anti-horário e parar. */
/* **** */
/* Inclui a biblioteca que controla a Motor Shield. */
#include <AFMotor.h>
```

```
/* Inclui a biblioteca que emula novas portas seriais.      */
#include <SoftwareSerial.h>

// Define os pinos para RX e TX
const int RX_PIN = 10;   // Pino digital 10 como RX
const int TX_PIN = 9;    // Pino digital 9 como TX

SoftwareSerial NovaSerial(RX_PIN, TX_PIN); // RX, TX

/* Portas da Motor Shield que estão sendo utilizadas.      */
AF_DCMotor Motor_Esquerda(2); /* Motor da esquerda */
AF_DCMotor Motor_Direita(3);  /* Motor da direita */

/* Variável que define a potência dos motores (0 a 255).    */
int potencia = 255;

void setup() {
  /* Inicia a comunicação serial no SoftwareSerial. */
  NovaSerial.begin(9600);
  /* Inicia a comunicação serial padrão. */
  Serial.begin(9600);

  /* Define as potências para os motores. */
  Motor_Esquerda.setSpeed(potencia);
  Motor_Direita.setSpeed(potencia);
}

void loop() {

  /* Se receber dados pela porta NovaSerial, faça... */
```

```
if (NovaSerial.available()) {  
    /* Guarde o valor recebido na variável comando. */  
    char comando = NovaSerial.read();  
    /* Imprime no monitor serial o comando guardado. */  
    Serial.println(comando);  
    /* Chama a função de controle dos motores informando o comando. */  
    controlarMotor(comando);  
}  
}  
  
/* Função de controle dos motores que receberá o comando. */  
void controlarMotor(char comando) {  
    /* Atribui o controle adequado para cada comando. */  
    switch (comando) {  
        case '0': // Parar  
            Motor_Direita.run(RELEASE);  
            Motor_Esquerda.run(RELEASE);  
            break;  
        case '1': // Frente  
            Motor_Direita.run(FORWARD);  
            Motor_Esquerda.run(FORWARD);  
            break;  
        case '2': // Gire a esquerda  
            Motor_Direita.run(FORWARD);  
            Motor_Esquerda.run(BACKWARD);  
            break;  
        case '3': // Gire a direita  
            Motor_Direita.run(BACKWARD);  
            Motor_Esquerda.run(FORWARD);  
            break;  
    }
```

```
case '4': // Ré
    Motor_Direita.run(BACKWARD);
    Motor_Esquerda.run(BACKWARD);
    break;
}
```

Após a transferência do programa ao Arduino, desconecte o cabo USB e faça alimentação do seu Robô utilizando a Bateria 9V ou a case com pilhas através do conector Jack P4 do Arduino.

Utilize um smartphone ou o próprio Notebook para se conectar à rede WiFi criada pelo Módulo Wireless ESP01 com o nome e senha definido no código fonte do programa. Após a conexão, abra um navegador e acesse o controle do seu Robô através do endereço 192.168.4.1, como já mostrado na figura 17. Agora é só diversão!





Desafio:

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...?

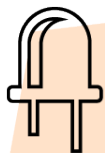
O projeto não funcionar, se atente a alguns dos possíveis erros:

- a.** Verifique se os Jumpers estão nos pinos certos, se estão na mesma coluna dos terminais dos componentes, fazendo assim as conexões;
- b.** Verifique se os Jumpers estão ligados aos pinos corretos no Arduino;
- c.** Reinicie o Módulo Wireless retirando por alguns segundos o pino 3,3V da Motor Shield e reconecte-o;
- d.** Procure manter os Jumpers interligados em furos mais próximos possíveis na Protoboard, isso evita o aumento da resistência elétrica que dificulta o fluxo de energia para o Módulo;
- e.** Verifique se a programação está adequada a cada porta digital e se as funções foram usadas corretamente;
- f.** Tente regravar o programa no Módulo Wireless.

3. Feedback e Finalização (15min):

- a.** Confira, compartilhando seu projeto com os demais colegas, se o objetivo foi alcançado.
- b.** Analise seu projeto desenvolvido, de modo a atender aos 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.

39 Robô Wireless



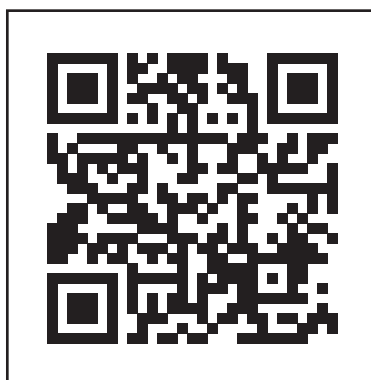
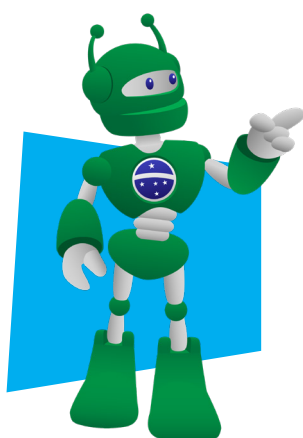
Videotutorial

Com o intuito de auxiliar na montagem e na programação desta aula, apresentamos um videotutorial, disponível em:



<https://rebrand.ly/a39robotica2>

Acesse, também, pelo QRCode:





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-caio-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-caio-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

Adilson Carlos Batista
Cleiton Rosa
Darice Alessandra Deckmann Zanardini
Edna do Rocio Becker
Marcelo Gasparin
Michelle dos Santos
Ricardo Hasper
Roberto Carlos Rodrigues
Simone Sinara de Souza

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 e do Esporte 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/)

