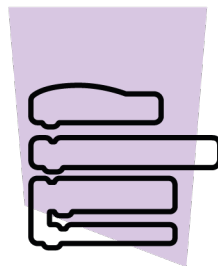
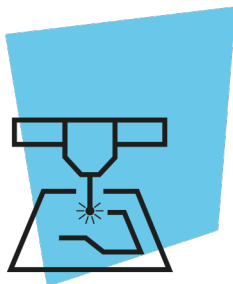
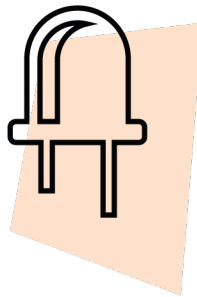
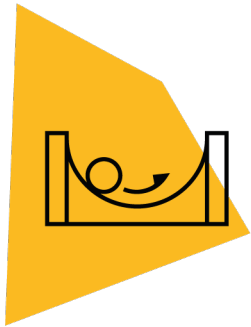
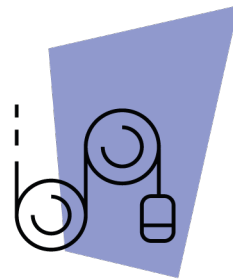
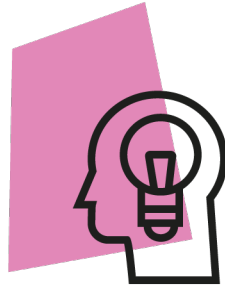


ROBÓTICA

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 Wirelless
- 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 - IoT
com Atuador (LED)

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

39 Robô Wireless



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

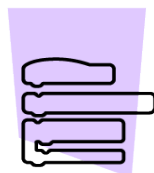
- Pensamento crítico;
- Afinidade digital;
- Resiliência;
- Resolução de problemas;
- Colaboração;
- Comunicação.



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).

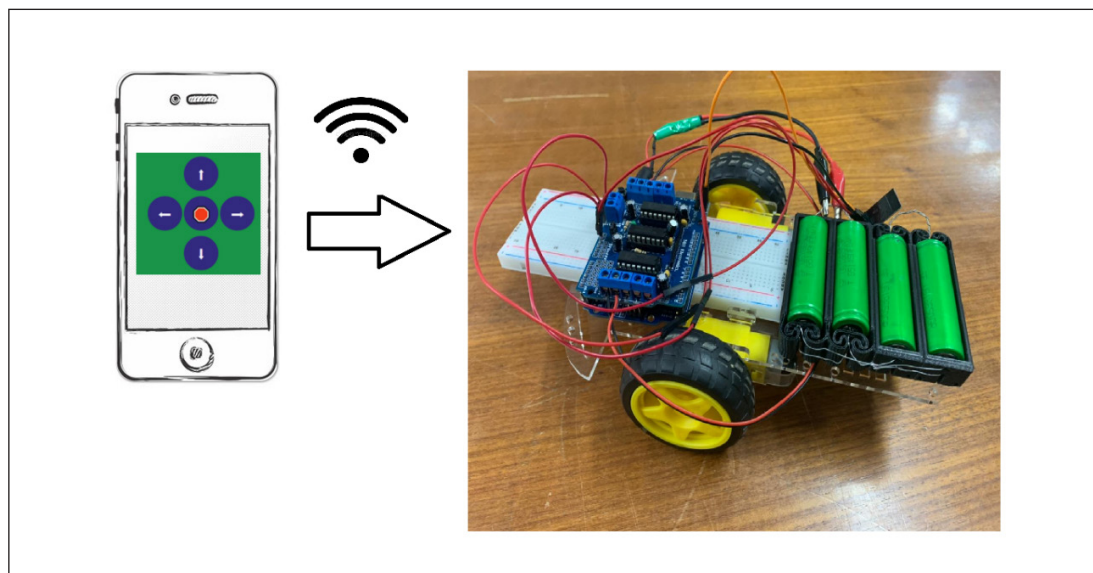


Figura 1 - Teclas de controle do robô



Fonte: SEED / DTI / CTE, 2022

Figura 2 - Esquema de funcionamento da aula.

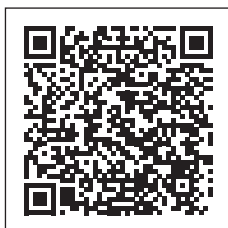


Fonte: SEED / DTI / CTE, 2022



Para Saber Mais...

Robôs inteligentes para manter a produtividade em alta pós pandemia.



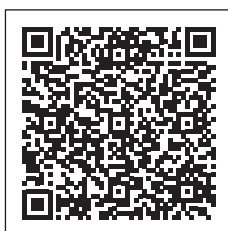
<https://exame.com/bussola/precisa-se-de-robos-inteligentes-para-manter-a-produtividade-em-alta/>

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



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

Pandemia disparou interesse por robôs.



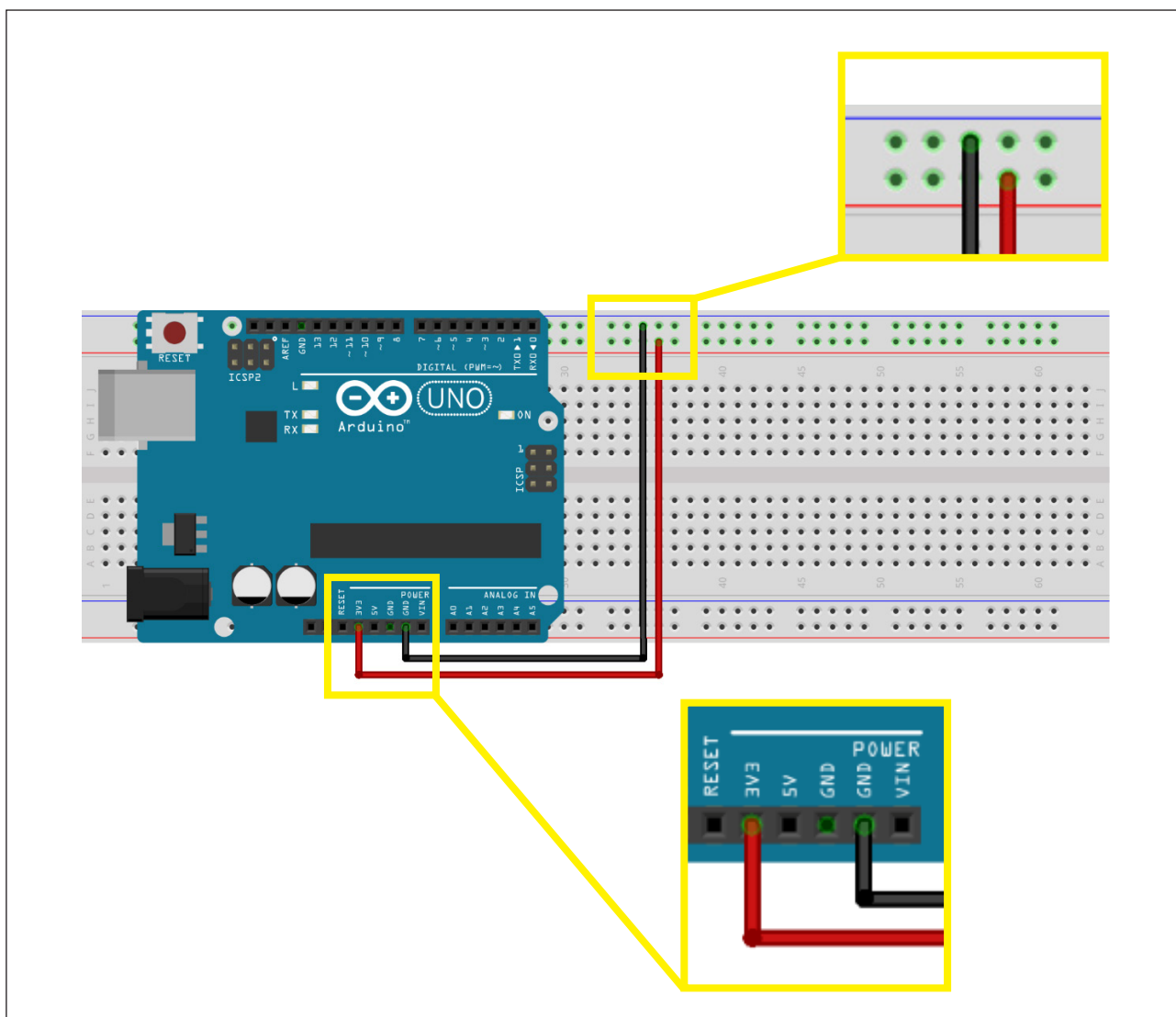
<https://www.uol.com.br/tilt/noticias/afp/2021/01/13/ces-2021-pandemia-disparou-interesse-por-uso-de-robos.htm>



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



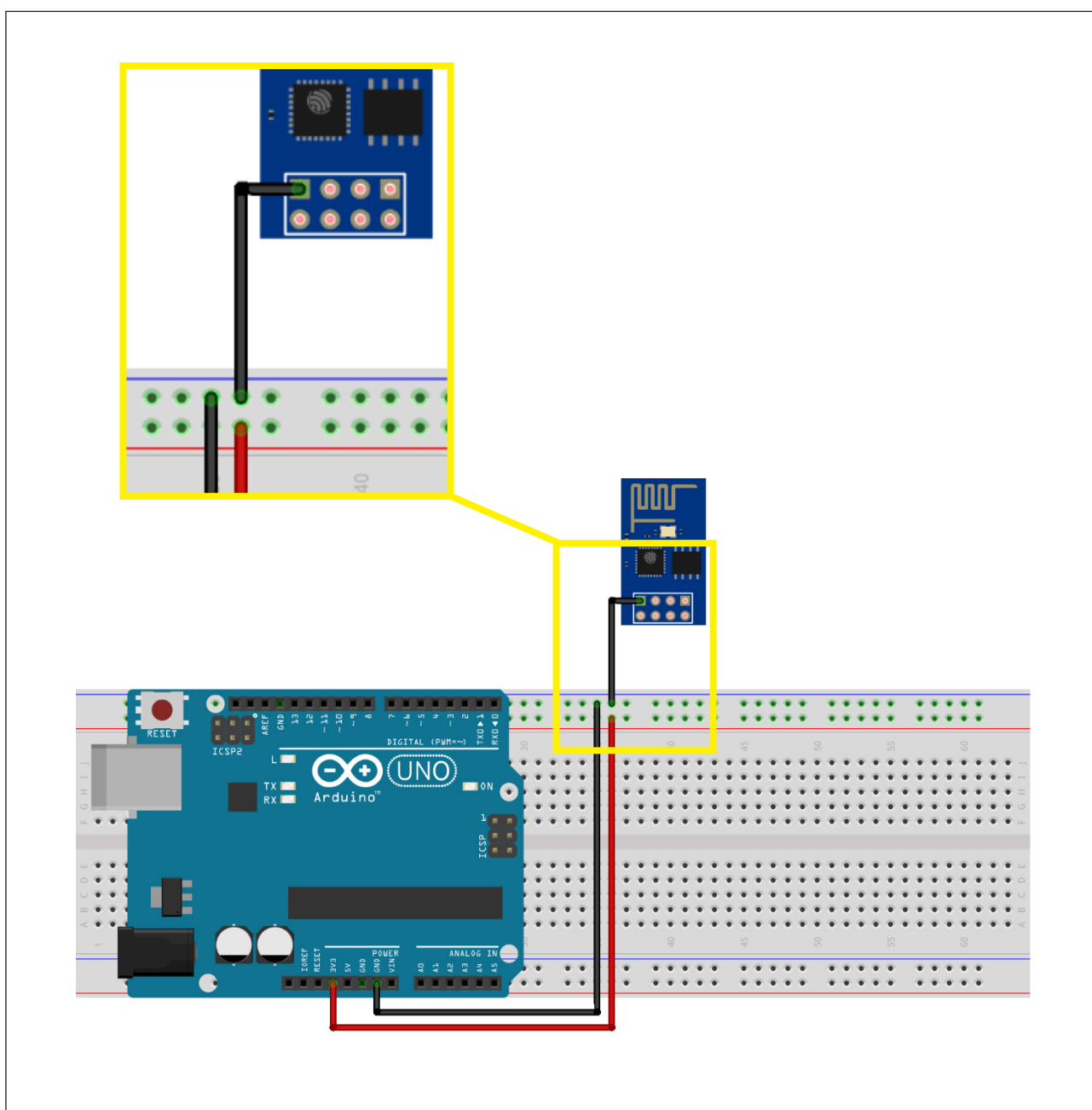
Fonte: Fritzing



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



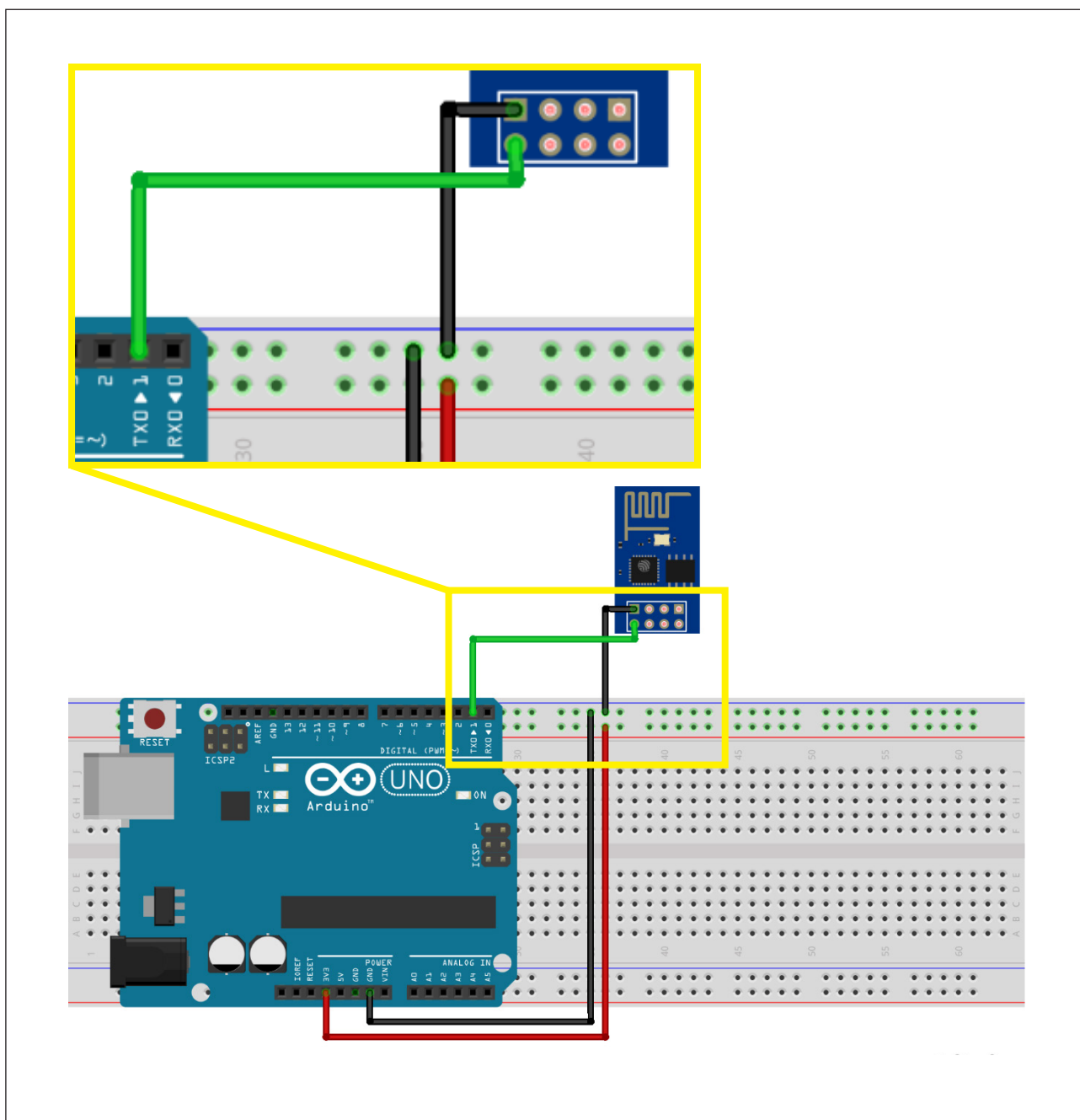
Fonte: Fritzing



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

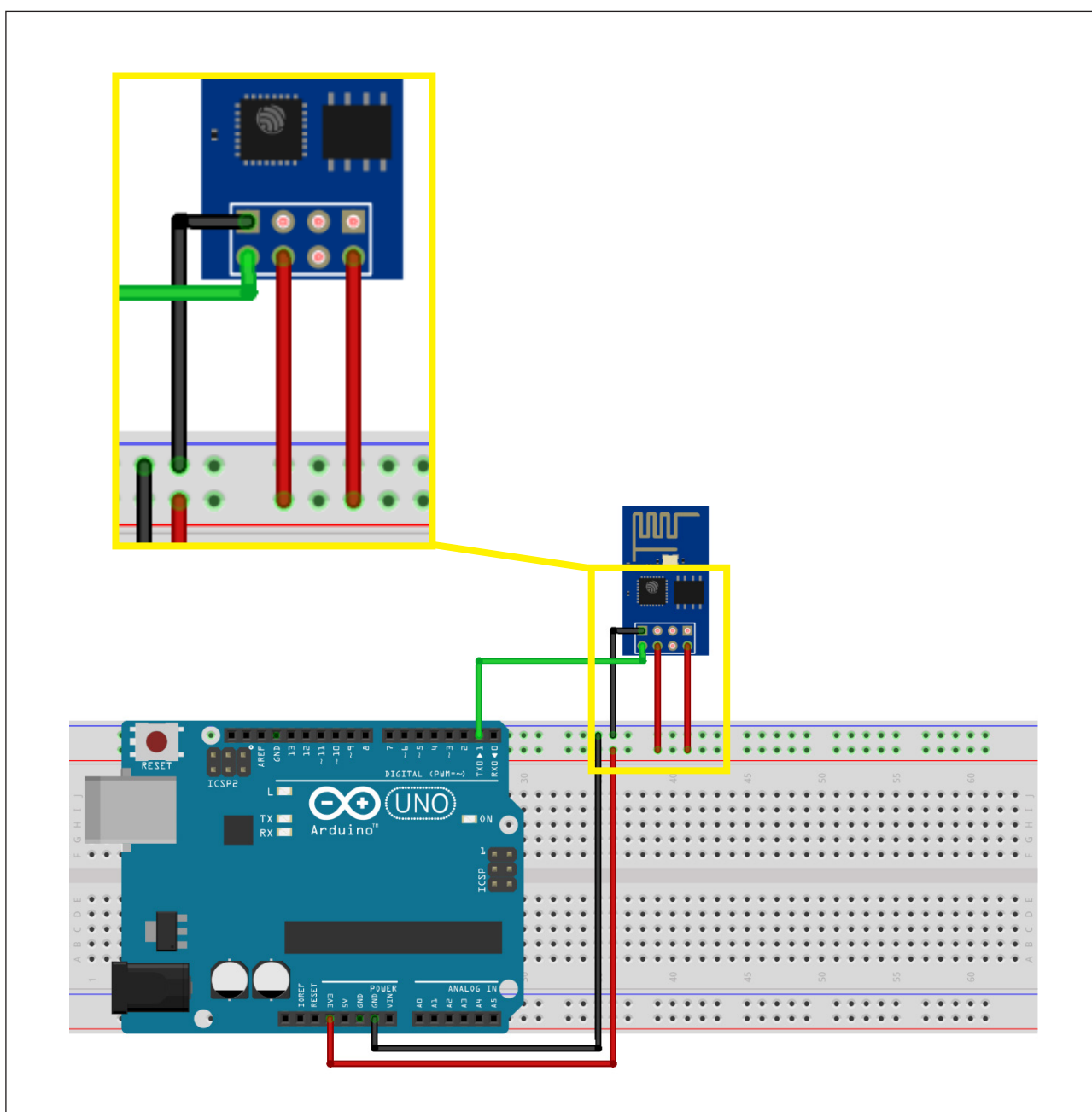


Fonte: Fritzing

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 Proto-board

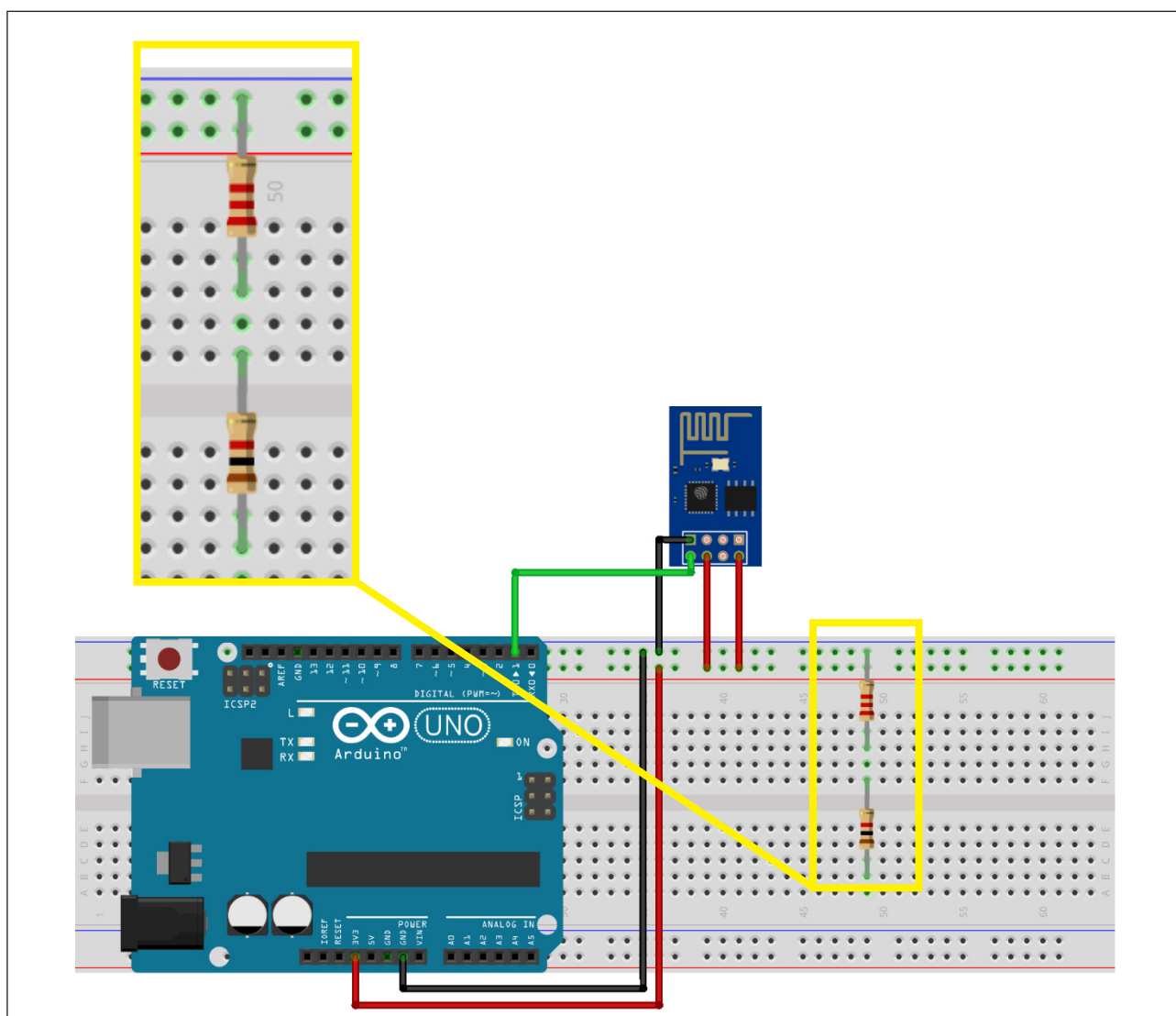


Fonte: Fritzing

39 Robô Wireless

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

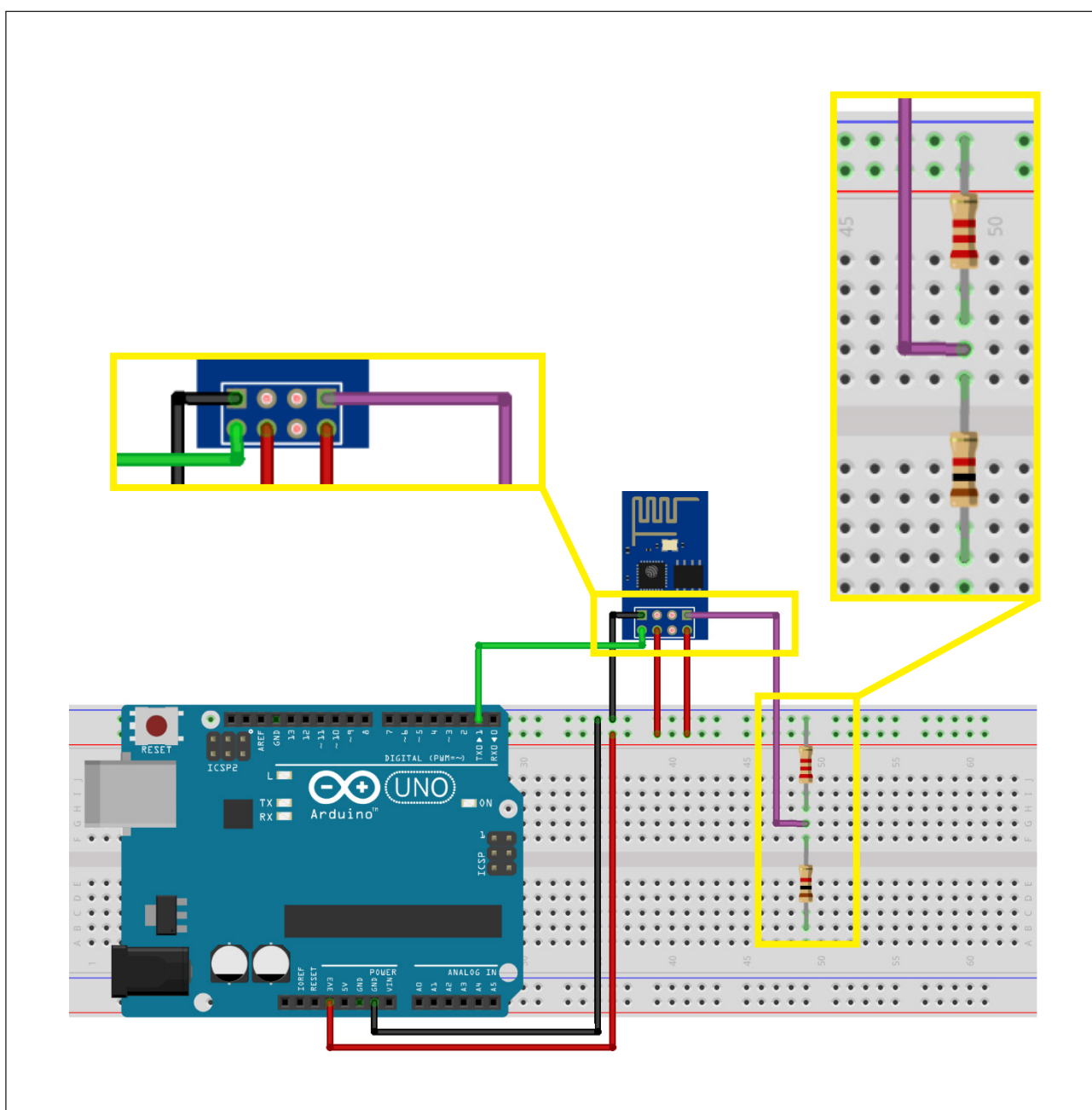


Fonte: Fritzing

39 Robô Wireless

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

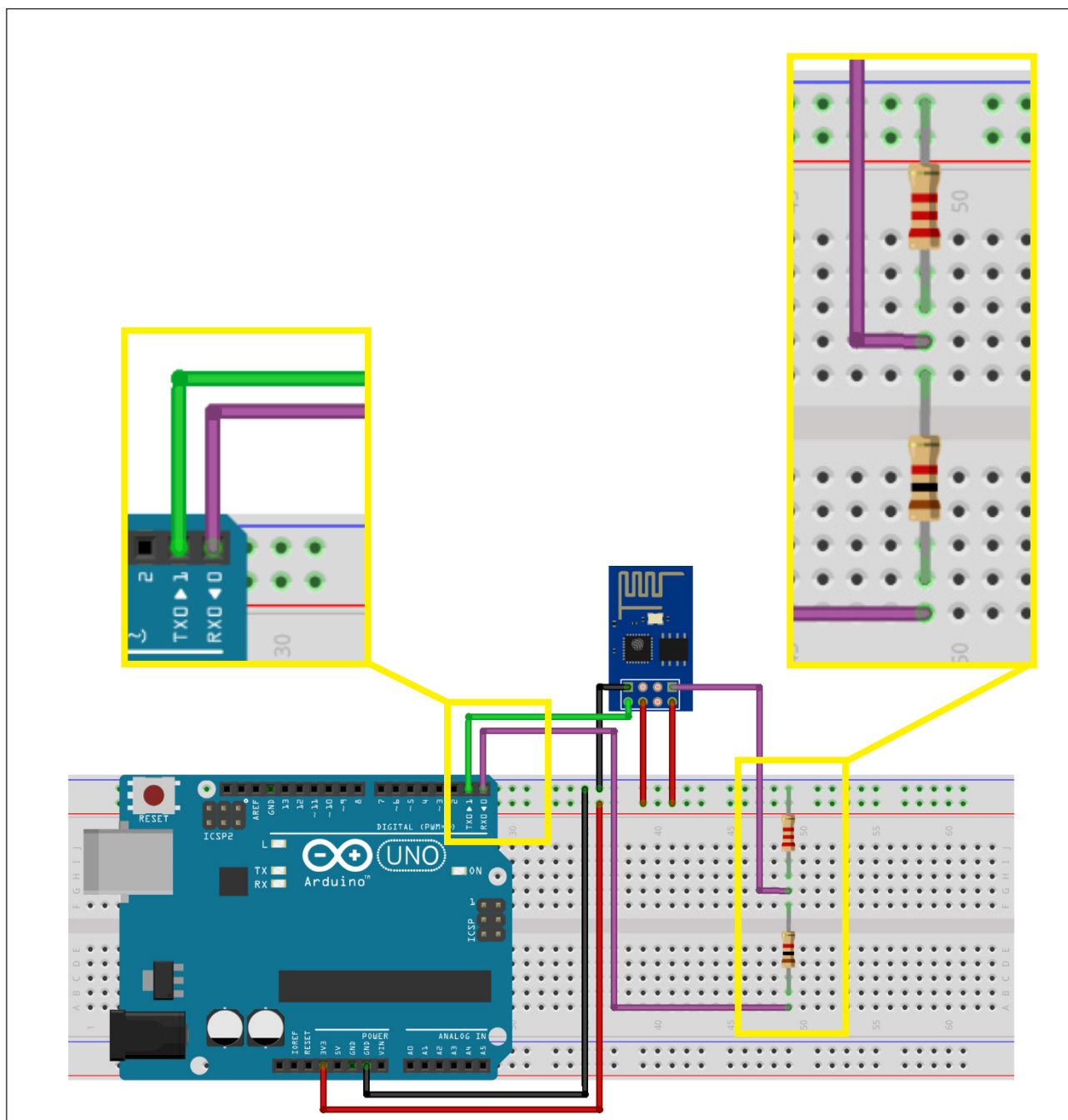


Fonte: Fritzing

39 Robô Wireless

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



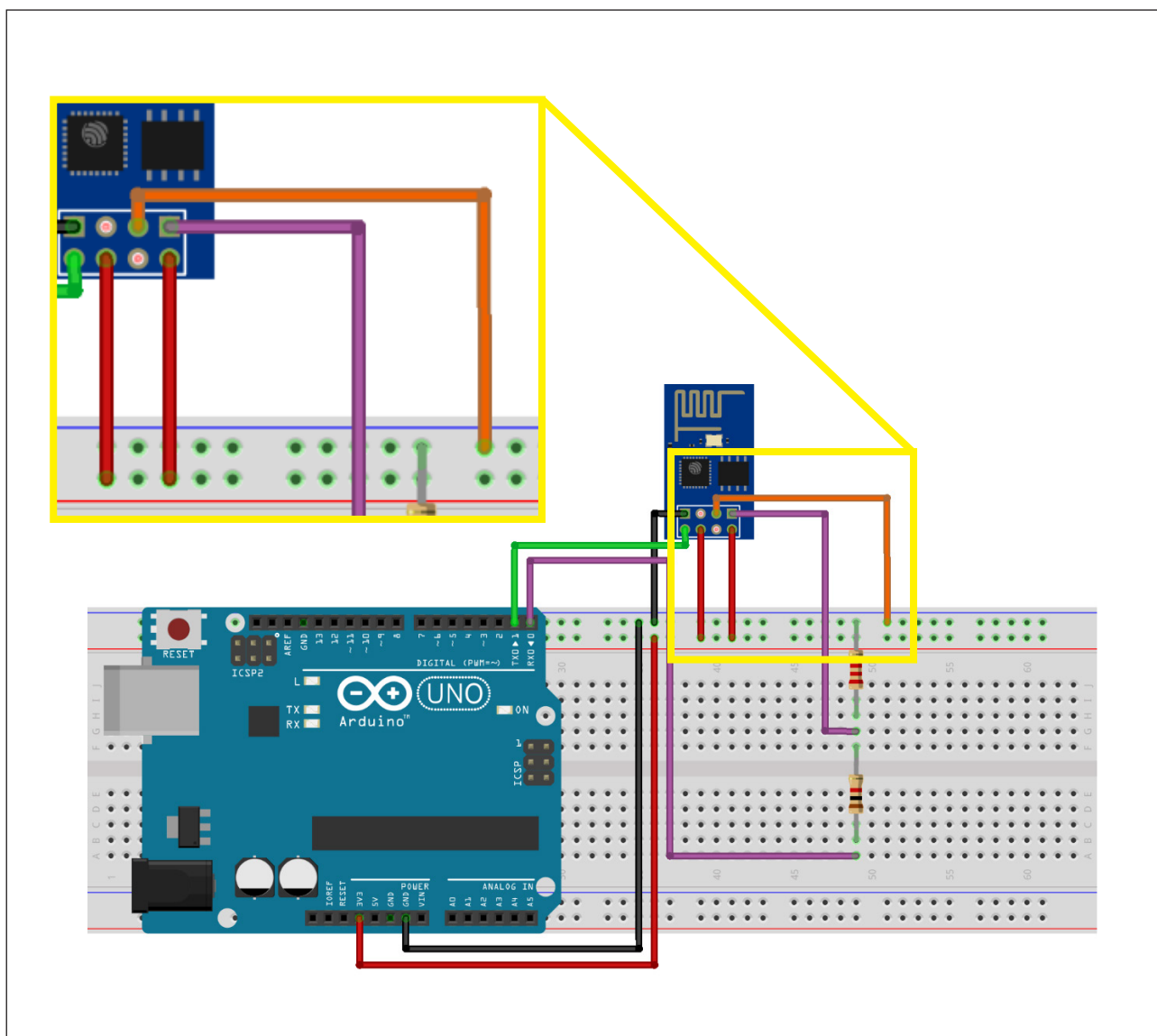
Fonte: Fritzing

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

 Agora, vamos programar!

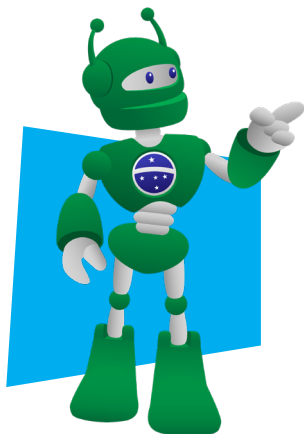
Antes de programarmos o Módulo Wireless, precisamos garantir que o conversor USB serial do Arduino não esteja sendo utilizado por códigos anteriormente gravados, basta carregarmos um Sketch com as funções **setup()** e **loop()** vazias.

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.

39 Robô Wireless

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 <WiFiClient.h>
#include <ESP8266WebServer.h>
/* Defina um nome para a sua rede.                       */
const char *nome_da_rede = "ALTERAR";

```

```
/* Defina uma senha para a sua rede (mínimo 8 caracteres). */
const char *senha = "12345678";
/* Define o pino GPIO2 do ESP-01 para envio dos pulsos. */
int pino_Pulsos = 2;
/* Cria um servidor web na porta 80. */
ESP8266WebServer servidor(80);
/* Variáveis que armazenarão os códigos da página HTML. */
String pagina = "";

void setup()
{
  /* Página HTML. */
  pagina = "<html>\n"
    "\n"
    "<style>\n"
    "body {\n"
    /* Código html para a cor de fundo da página. */
    "background: #009541; \n"
    "}\n"
    /* Código html para configuração dos botões. */
    ".button {\n"
    "background-color: #312682;\n"
    "border: none;\n"
    "color: white;\n"
    "padding: 15px 32px;\n"
    "text-align: center;\n"
    "display: inline-block;\n"
    "font-size: 64px;\n"
    "margin: 4px 2px;\n"
```

```
"cursor: pointer;\n"
"border-radius: 100%;\n"
"width: 150px;\n"
"height: 150px;\n"
"}\n"
"</style>\n"
"<section><center>\n"
"<br><br>\n"
/* Código html para os títulos da página. */
"<font size=\"6\"> \n"
"<h1>#129302; "
"ROB&Oacute;TICA PARAN&Aacute; "
"#129302; </h1>\n"
"<h1>#x1F3AE; "
"Aula 39 - Rob&ocirc; Wireless "
"#128246;</h1>\n"
/* Código html para o botão Frente. */
"<h1>Movimento Pulsado</h1>"
"<p><a href=\"/frente_p\">"
"<button class=\"button button\">"
"#x2B06;</button></a>\n"
/* Código html para o botão Esquerda. */
"<p><a href=\"/esquerda_p\">"
"<button class=\"button button\">"
"#11013;</button></a>\n"
/* Código html para o botão Pare. */
"<a href=\"/pare\">"
"<button class=\"button button\">"
"#x1F6D1;</button></a>\n"
```

```

/* Código html para o botão Direita */
"<a href=\"/direita_p\">"
"<button class=\"button button\">"
"&#10145;</button></a>\n"
/* Código html para o botão Ré. */
"<p><a href=\"/re_p\">"
"<button class=\"button button\">"
"&#11015;</button></a>\n"
"<BR>"
"<BR>"
"<h1>Movimento Cont&iacute;nuo</h1>"
"</font>\n"
"<p><a href=\"/frente_c\">"
"<button class=\"button button\">"
"&#x2B06;</button></a>\n"
/* Código html para o botão Pare. */
"<p><a href=\"/pare\">"
"<button class=\"button button\">"
"&#x1F6D1;</button></a>\n"
/* Código html para o botão Ré. */
"<p><a href=\"/re_c\">"
"<button class=\"button button\">"
"&#11015;</button></a>\n";

/* Fim dos códigos da página HTML. */
/* Define o pino dos pulsos como saída. */
pinMode(pino_Pulsos, OUTPUT);
/* Inicia com o pino_Pulsos desligado. */
digitalWrite(pino_Pulsos, LOW);

```

```
/* Cria o ponto de acesso com o nome e a senha definida. */
WiFi.softAP(nome_da_rede, senha);
/* Requisição da página principal, envia ao cliente a */
/* página de controle do robô. */
servidor.on("/", []() {
    servidor.send(200, "text/html", pagina);
});
/* Requisição do botão FRENTE, envia ao Arduino */
/* um pulso de 10 milissegundos. */
servidor.on("/frente_p", []() {
    servidor.send(200, "text/html", pagina);
    pulso(10);
});
/* Requisição do botão ESQUERDA, envia ao Arduino */
/* um pulso de 20 milissegundos. */
servidor.on("/esquerda_p", []() {
    servidor.send(200, "text/html", pagina);
    pulso(20);
});
/* Requisição do botão PARE, envia ao Arduino */
/* um pulso de 30 milissegundos. */
servidor.on("/pare", []() {
    servidor.send(200, "text/html", pagina);
    pulso(30);
});
/* Requisição do botão DIREITA, envia ao Arduino */
/* um pulso de 40 milissegundos. */
servidor.on("/direita_p", []() {
```



```
servidor.send(200, "text/html", pagina);
pulso(40);
});
/* Requisição do botão Ré, envia ao Arduino */
/* um pulso de 50 milissegundos. */
servidor.on("/re_p", []() {
    servidor.send(200, "text/html", pagina);
    pulso(50);
});
/* Requisição do botão FRENTE contínuo, envia ao */
/* Arduino um pulso de 60 milissegundos. */
servidor.on("/frente_c", []() {
    servidor.send(200, "text/html", pagina);
    pulso(60);
});
/* Requisição do botão RÉ contínuo , envia ao */
/* Arduino um pulso de 70 milissegundos. */
servidor.on("/re_c", []() {
    servidor.send(200, "text/html", pagina);
    pulso(70);
});
/* Inicia o servidor. */
servidor.begin();
}
```

```
void loop()
{
  /* Função que monitora se o cliente fez uma requisição */
  /* e envia o pulso correspondente. */
  servidor.handleClient();
}

/* Função responsável por enviar o pulso ao Arduino com a */
/* duração informada de acordo com a requisição: */
/* Seguir em frente -> Pulso de 10 milissegundos. */
/* Girar à esquerda -> Pulso de 20 milissegundos. */
/* Parar -> Pulso de 30 milissegundos. */
/* Girar à direita -> Pulso de 40 milissegundos. */
/* Ré -> Pulso de 50 milissegundos. */
/* Frente contínuo -> Pulso de 60 milissegundos. */
/* Ré contínuo -> Pulso de 70 milissegundos. */

void pulso(int tempo) {
  /* Ativa o pino do pulso. */
  digitalWrite(pino_Pulsos, HIGH);
  /* Aguarda a duração do pulso. */
  delay(tempo);
  /* Desativa o pino do pulso. */
  digitalWrite(pino_Pulsos, LOW);
}
```

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.

39 Robô Wireless

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
```

Fonte: SEED / DTI / CTE, 2022

Ao terminar o processo de gravação, será apresentada a mensagem **Hard resetting via RTS pin...**, (figura 11), então, devemos reiniciar o Módulo Wireless removendo o cabo USB do computador ou do Arduino e também o Jumper conectado entre o pino **GPIO0** do Módulo e a linha lateral **AZUL** da Protoboard (figura 14).

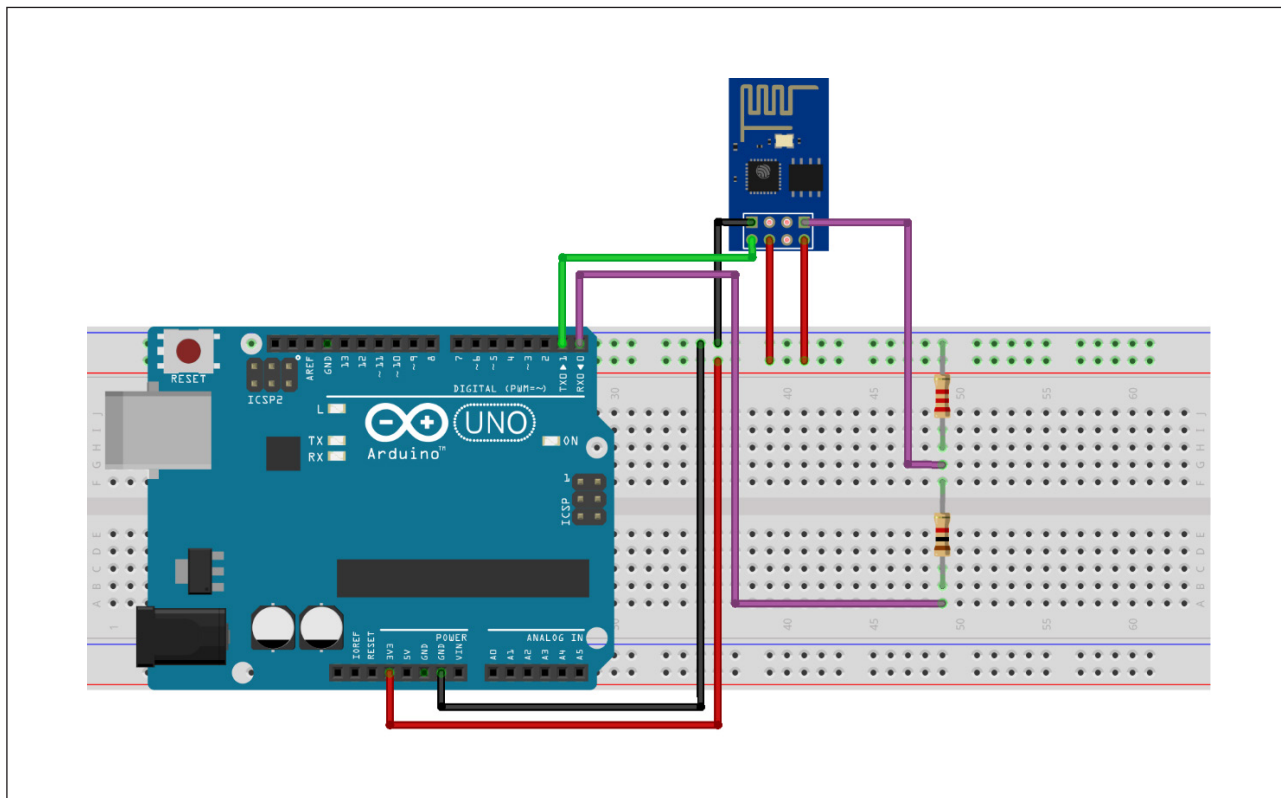
Figura 11 – Término da gravação do código ao Modulo Wireless

```
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 + 32KB SRAM
```

Fonte: SEED / DTI / CTE, 2022

39 Robô Wireless

Figura 12 - Jumper removido do Pino GPIO do Módulo Wireless



Fonte: Fritzing

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:

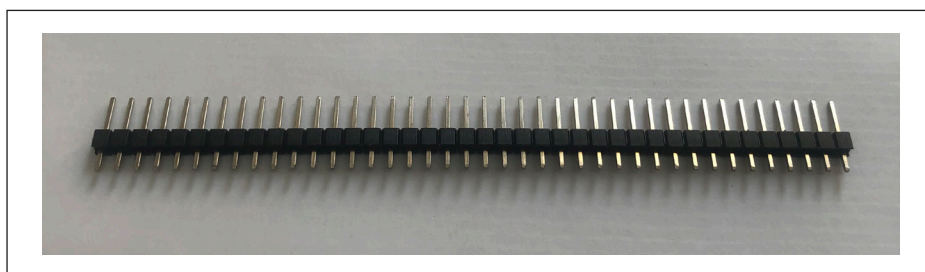
- Para criar suas páginas HTML acesse o site w3school através do link: <https://www.w3schools.com/>
- Para converter a página HTML na linguagem que o Software Arduino IDE compreenda (C/C++), utilize a ferramenta online através do link http://tomeko.net/online_tools/cpp_text_escape.php?lang=en

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

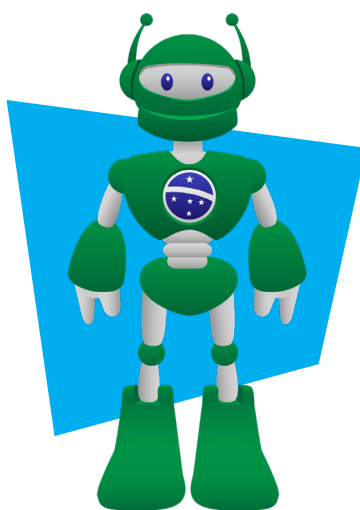
Antes de iniciarmos a montagem do nosso robô Wireless, certifique-se de que o Kit Chassi 2WD esteja corretamente montado, ou seja, os motores e as rodas, fixados à estrutura de acrílico. Caso precise montar ou fazer alguns ajustes, você poderá consultar o guia de montagem do KIT Chassi 2WD através do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

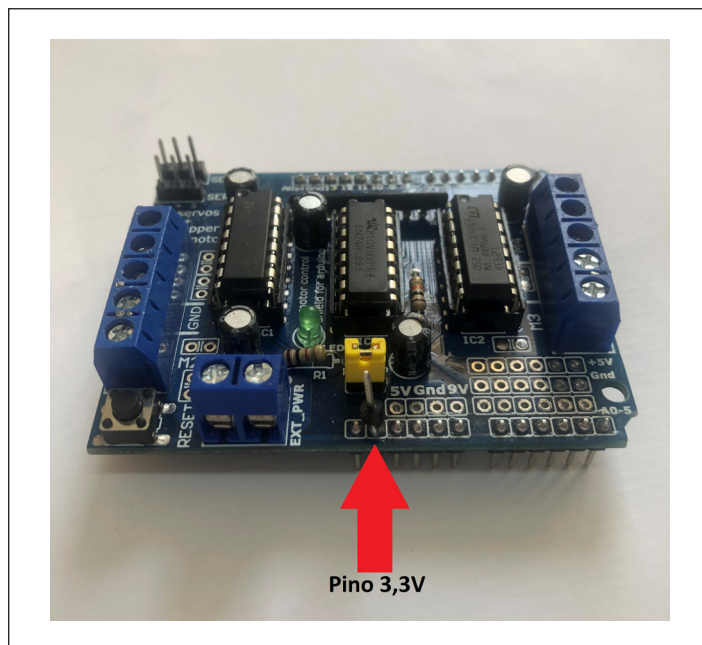


Fonte: SEED / DTI / CTE, 2022



39 Robô Wireless

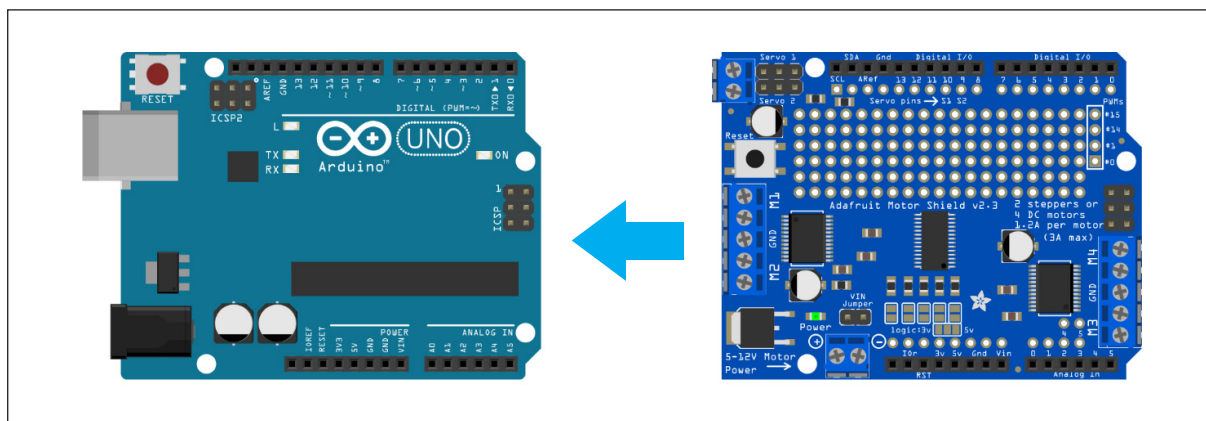
Figura 19 - Pino soldado para acesso do 3,3V do Arduino



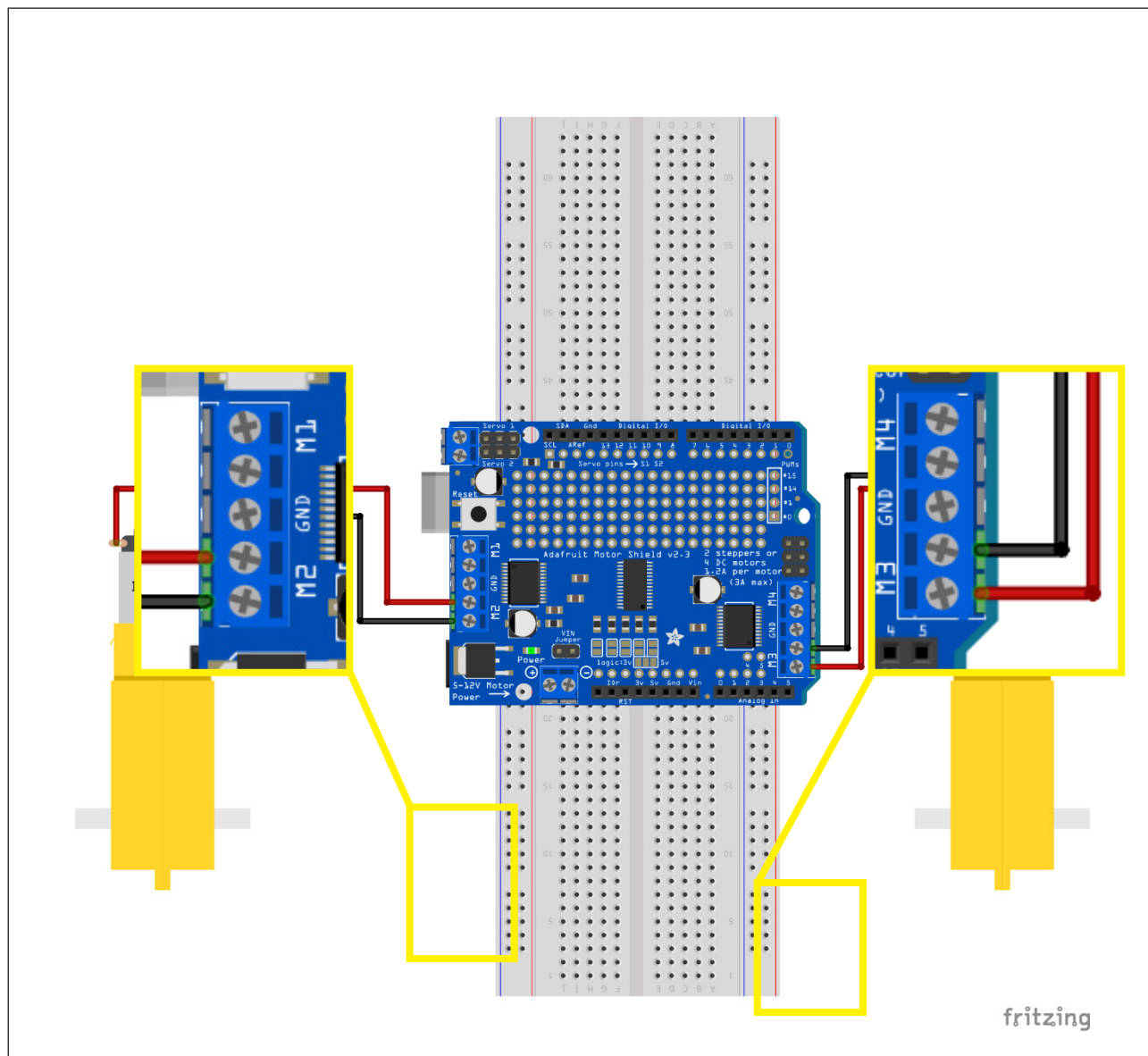
Fonte: SEED / DTI / CTE, 2022

Agora, estamos prontos para dar início à montagem do robô Wireless. Comece preparando 2 Jumpers Macho-Fêmea encaixando 2 Jumpers Macho-Macho à 2 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 20.

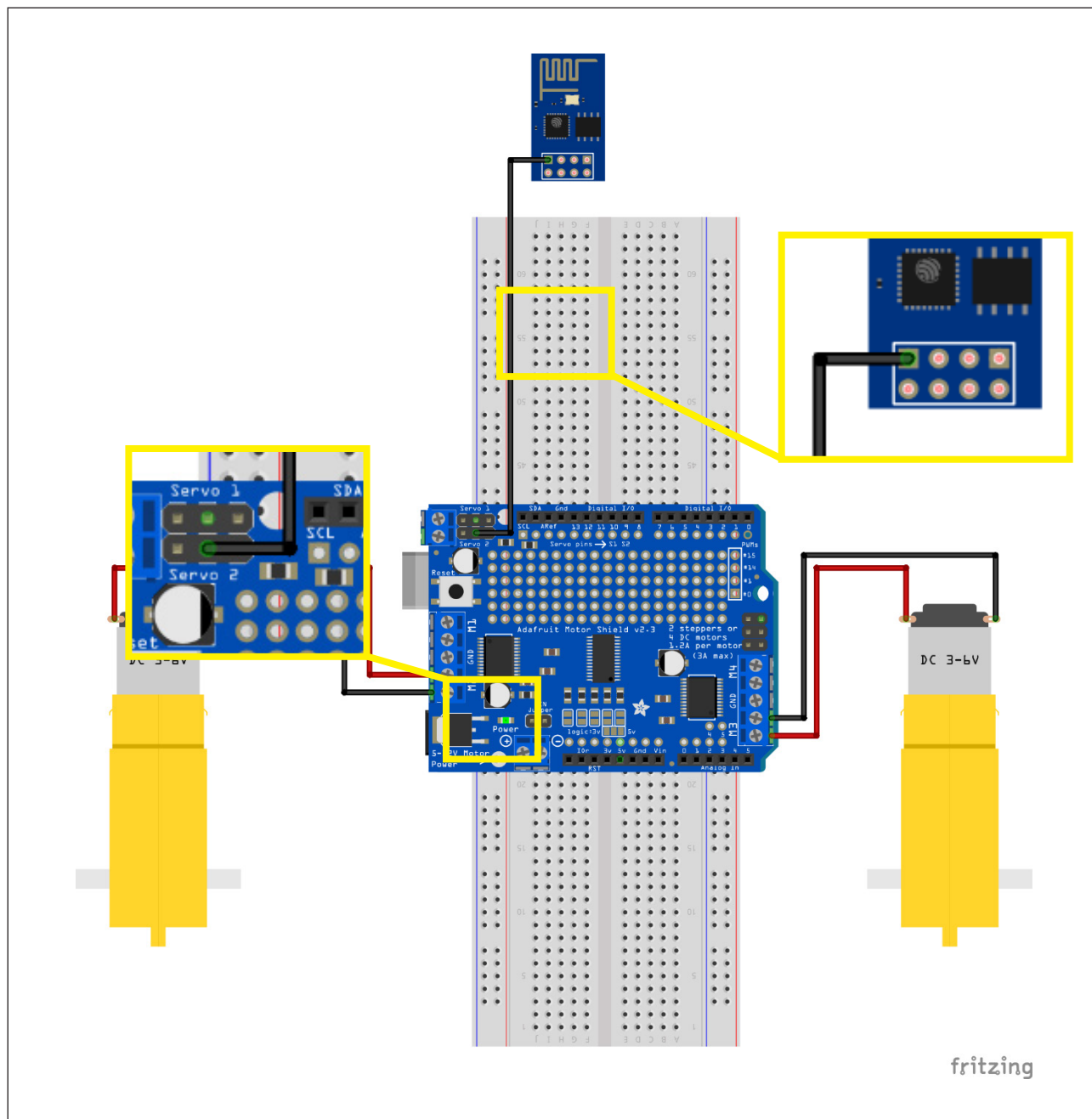
Figura 20: Encaixe da placa Motor Shield L293D sobre o Arduino



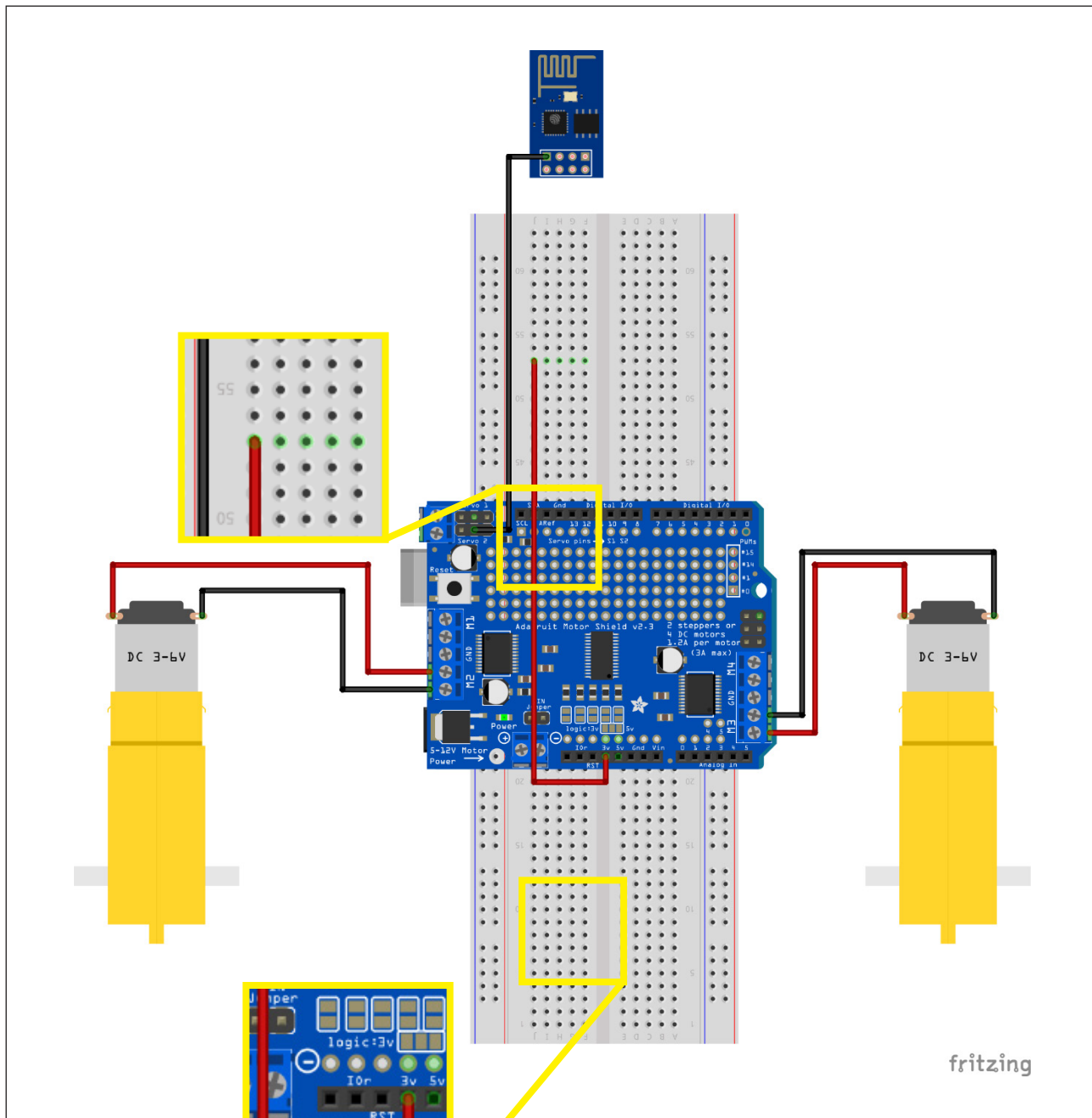
Fonte: Fritzing



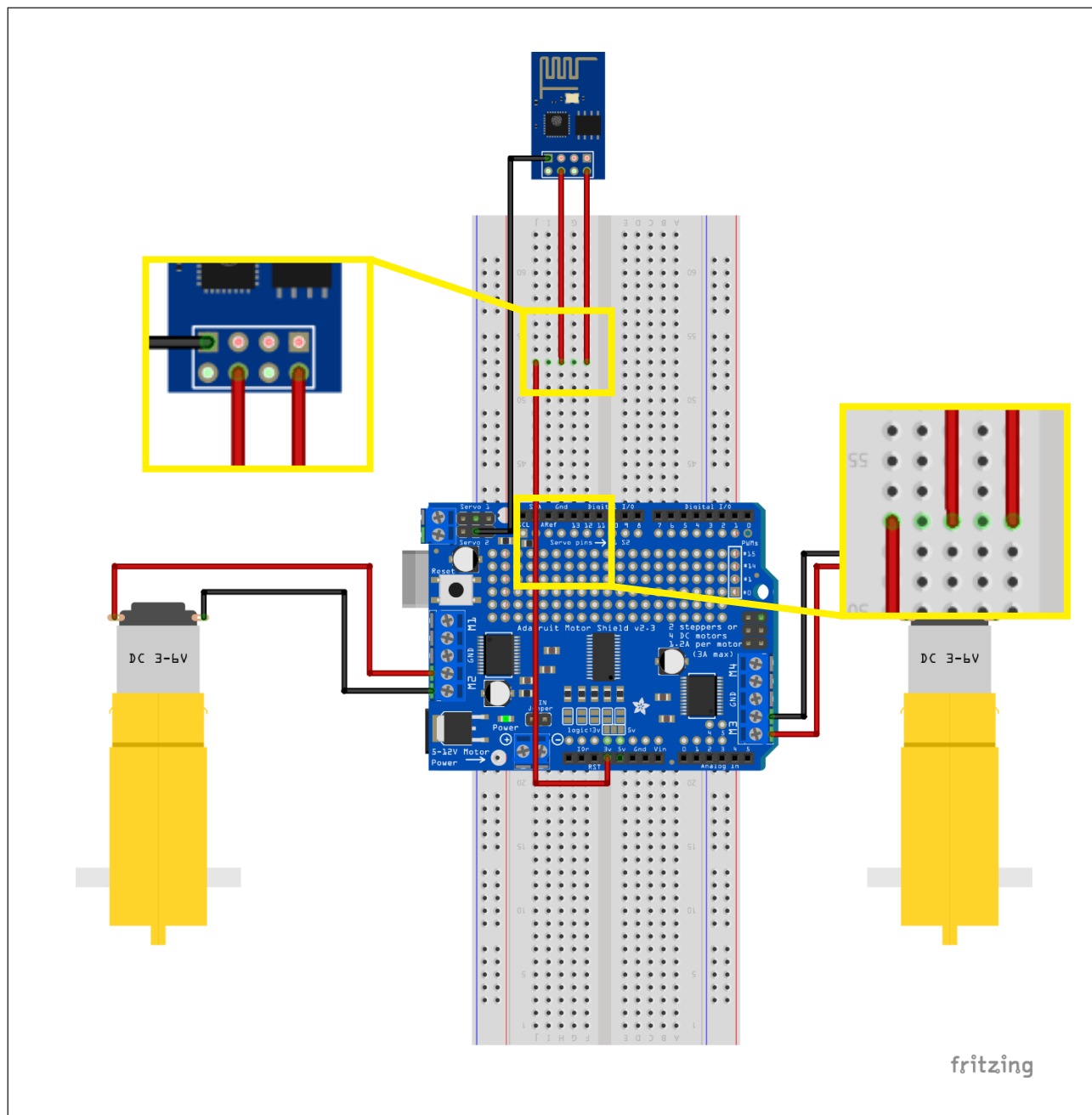
Fonte: Fritzing



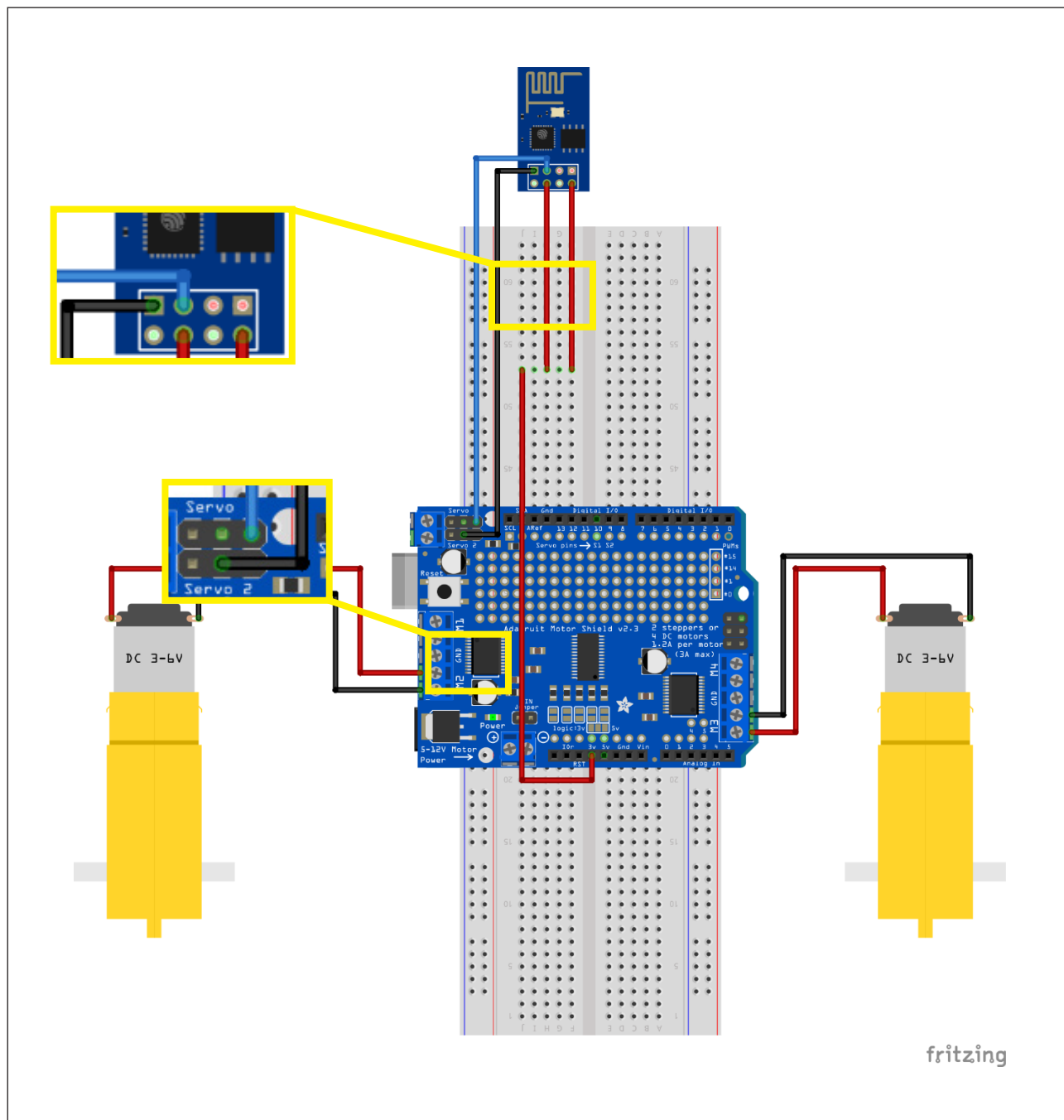
Fonte: Fritzing



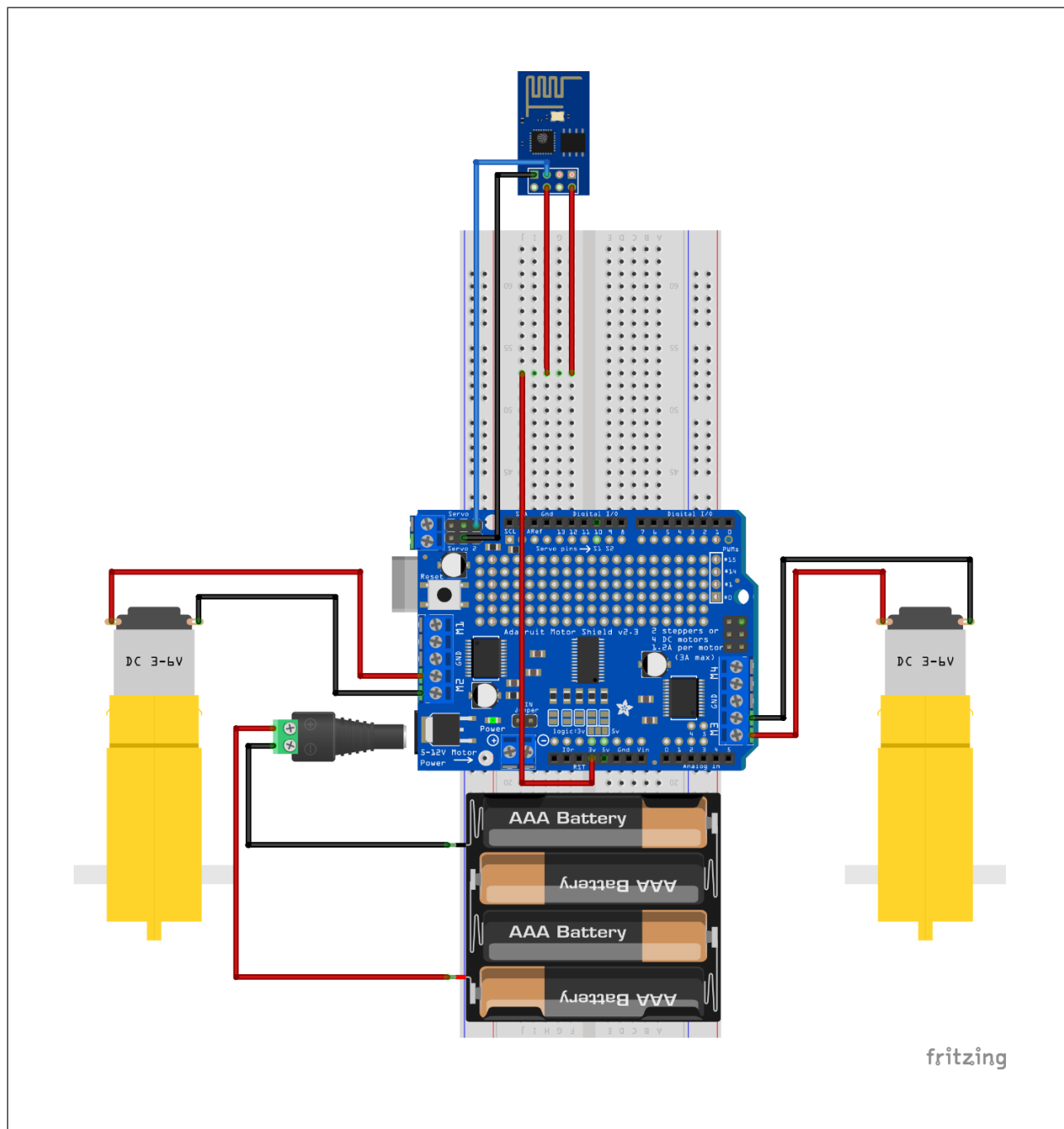
Fonte: Fritzing



Fonte: Fritzing



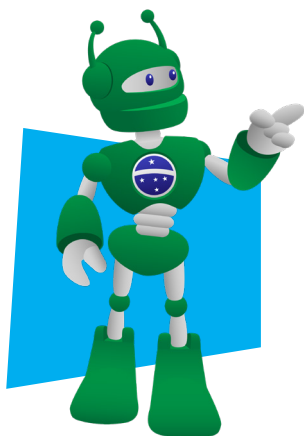
Fonte: Fritzing



Fonte: Fritzing

Agora, vamos programar!

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>

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

/* Variável que define a potência dos motores (0 a 255). */
int potencia = 140;

/* Define o pino que receberá os pulsos do módulo wireless. */
int pino_ESP = 10;

/* Variáveis utilizadas para determinar a duração dos */
/* pulsosemitidos pelo módulo wireless. */
unsigned long Tempo_Pulso;
unsigned long Tempo_Inicial_Pulso;
unsigned long Tempo_Final_Pulso;

void setup() {
    /* Configura o pino que receberá os pulsos como entrada. */
    pinMode(pino_ESP, INPUT);

    /* Inicia com os motores parados. */
    pare();
}

void loop() {
    /* Chama a função que checa a duração do pulso recebido. */
    checa_pulso();

    /* Se receber um pulso em torno de 10 milissegundos, */
    /* siga em frente por 500 milissegundos e pare. */
    if (Tempo_Pulso >= 5 && Tempo_Pulso <= 12) {
        frente();
        delay(500);
        pare();
    }
}
```

```
/* Se receber um pulso em torno de 20 milissegundos, */
/* gire à Esquerda por 200 milissegundos e pare. */
if (Tempo_Pulso >= 15 && Tempo_Pulso <= 22) {
    esquerda();
    delay(200);
    pare();
}
/* Se receber um pulso em torno de 30 milissegundos, */
/* pare o movimento. */
if (Tempo_Pulso >= 25 && Tempo_Pulso <= 32) {
    pare();
}
/* Se receber um pulso em torno de 40 milissegundos, */
/* gire à direita por 200 milissegundos e pare. */
if (Tempo_Pulso >= 35 && Tempo_Pulso <= 42) {
    direita();
    delay(200);
    pare();
}
/* Se receber um pulso em torno de 50 milissegundos, */
/* mova de ré por 500 milissegundos e pare. */
if (Tempo_Pulso >= 45 && Tempo_Pulso <= 52) {
    re();
    delay(500);
    pare();
}
```



```
/* Enquanto o pulso estiver em torno de 60 milissegundos, */
/* mantenha o movimento em marcha ré. */
while (Tempo_Pulso >= 55 && Tempo_Pulso <= 62) {
    frente();
    checa_pulso();
}
/* Enquanto o pulso estiver em torno de 70 milissegundos, */
/* mantenha o movimento seguindo em frente. */
while (Tempo_Pulso >= 65 && Tempo_Pulso <= 72) {
    re();
    checa_pulso();
}
}
/***** Funções Criadas *****/
/* Função que para os motores. */
void pare()
{
    Tempo_Pulso = 30;
    Motor_Esquerda.run(RELEASE);
    Motor_Direita.run(RELEASE);
}
/* Função que aciona os motores para frente. */
void frente()
{
    Motor_Esquerda.setSpeed(potencia);
    Motor_Direita.setSpeed(potencia);
    Motor_Esquerda.run(FORWARD);
    Motor_Direita.run(FORWARD);
}
```

```
/* Função que aciona os motores para trás. */
void re()
{
    Motor_Esquerda.setSpeed(potencia);
    Motor_Direita.setSpeed(potencia);
    Motor_Esquerda.run(BACKWARD);
    Motor_Direita.run(BACKWARD);
}

/* Função que gira o robô para a esquerda. */
void esquerda()
{
    Motor_Esquerda.setSpeed(potencia);
    Motor_Direita.setSpeed(potencia);
    Motor_Esquerda.run(BACKWARD);
    Motor_Direita.run(FORWARD);
}

/* Função que gira o robô para a direita. */
void direita()
{
    Motor_Esquerda.setSpeed(potencia);
    Motor_Direita.setSpeed(potencia);
    Motor_Esquerda.run(FORWARD);
    Motor_Direita.run(BACKWARD);
}
```

```
/* Função que calcula a duração do pulso recebido */
void checa_pulso() {
  /* Se o pulso iniciar, faz... */
  if (digitalRead(pino_ESP) == HIGH) {
    /* Marca o tempo do início do pulso. */
    Tempo_Inicial_Pulso = millis();
    /* Aguarda enquanto o pulso estiver sendo enviado. */
    while (digitalRead(pino_ESP) == HIGH) {}
    /* Marca o tempo que o pulso terminou. */
    Tempo_Final_Pulso = millis();
    /* Calcula a duração do pulso (T_final - T_inicial). */
    Tempo_Pulso = Tempo_Final_Pulso - Tempo_Inicial_Pulso;
  }
  delay(1);
}
```

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:

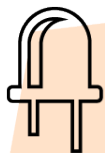
i. Colaboração e cooperação: Você e os membros de sua equipe interagiram entre si, compartilhando ideias que promoveram a aprendizagem e o desenvolvimento deste projeto?

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



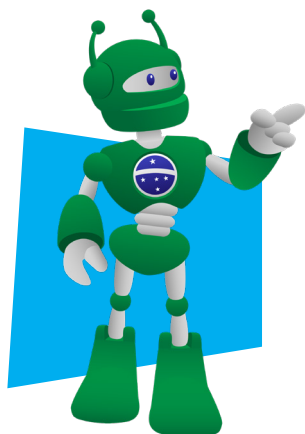
Videotutorial

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:



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/)

