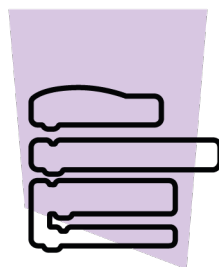
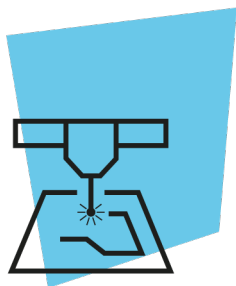
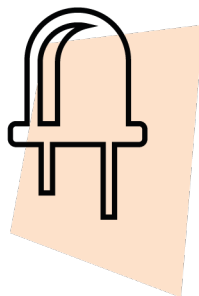
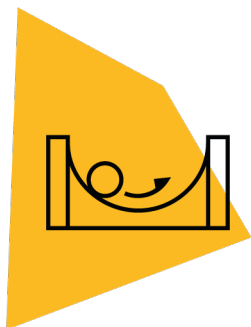
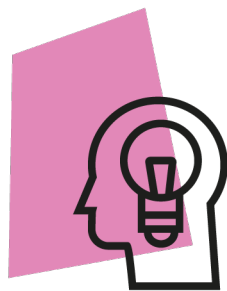


ROBÓTICA

Módulo 2



Robô Radio Controlado

AULA **41**

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

Adilson Carlos Batista

Cleiton Rosa

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

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 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 40
Braço Robótico
Via RF
(Rádio Frequência)

Aula 41
**Robô Radio
Controlado**

Aula 42
Feedbacks
+ Inventário III

Sumário

Introdução	2
Objetivos desta Aula	2
Competências Previstas na BNCC	3
Habilidades do Século XXI a Serem Desenvolvidas	4
Roteiro da aula	4
1. Contextualização	4
2. Conteúdo	5
3. Feedback e Finalização	9
Referências	10



Introdução

É muito comum os Robôs controlados por rádio (ondas) serem utilizados principalmente para executar ações que envolvem riscos de vida para o ser humano, como desarmar bombas, mandar imagens de lugares com risco de desmoronamento, se o robô tiver câmera acoplada, como sempre ocorre ao investigar planetas como Marte, por exemplo.

Nesta aula, por meio do Módulo RF433 (Emissor - Receptor) iremos programar um robô via rádio para realizar movimentos controlados por um Joystick.



Objetivos desta Aula

- Retomar o Módulo RF433 e o Kit Chassi 2WD;
- Programar um robô rádio controlado;



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.



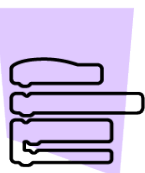
Habilidades do Século XXI a Serem Desenvolvidas

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



Lista de Materiais

- 02 Placas Arduino Uno R3;
- 01 Motor Shield;
- 01 Joystick Shield;
- 01 Cabo USB;
- 06 Jumpers Fêmea-Fêmea;
- 03 Jumpers Macho-Macho;
- 01 Transmissor RF 433MHz;
- 01 Receptor RF 433MHz;
- 01 Kit Chassi 2WD;
- 02 Suportes de pilhas com conector P4;
- Notebook;
- Software Arduino IDE.



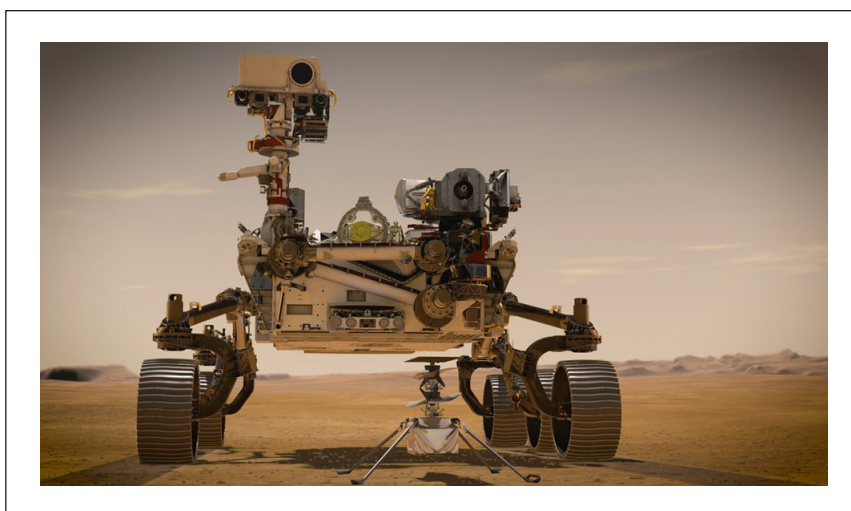
Roteiro da Aula

1. Contextualização:

Um robô, via rádio, poderá executar diferentes ações, dependendo dos componentes que estejam acoplados a ele, no caso dele ter uma câmera, por exemplo, poderá registrar, com imagens, os lugares por onde o robô passar, mostrando os obstáculos que serão encontrados no trajeto, caso tenham braços robóticos, podem coletar objetos ou executar ações.

A NASA faz uso de robôs como esses para fazer verificações de solo, um exemplo é o Robô Perseverance (figura 1), um explorador enviado ao planeta vermelho em 18 de fevereiro de 2021, para pousar na cratera de Jezero e buscar vestígios de vida. Segundo os cientistas, esse lugar foi um lago há bilhões de anos, o robô tem por missão coletar amostras do solo, observar a geologia e transformar dióxido de carbono em oxigênio.

Figura 1 - Robô Perseverance enviado a Marte



Fonte: Nasa, 2022

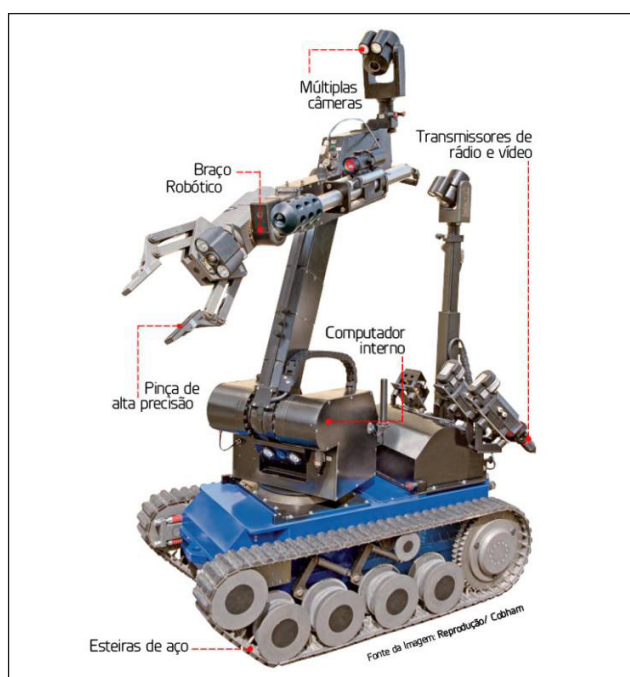
ROBÔ RÁDIO CONTROLADO

Os equipamentos utilizados nestas explorações são de grande frequência e fazem usos de satélites ao redor do planeta Terra, diferente dos módulos que iremos utilizar em nosso protótipo que alcança pequenas distâncias.

Os Robôs Controlados por rádio são muito importantes para a Defesa Civil na verificação de lugares com problemas de desmoronamento ou até mesmo serem utilizados pela polícia para averiguar casos de desarmamento de bombas – robô antibomba.

Os robôs antibombas funcionam por meio de um controle remoto, eles são mais baixos, largos e com rodas esteiras para passar por desníveis e pedregulhos. Um exemplo é o Cobham Teodor, esse possui duas esteiras de aço para locomoção, um braço robótico, uma pinça de grande precisão e câmeras com capacidade de visão noturna, um transmissor de rádio e vídeo e um computador interno capaz de consertar pequenos problemas que possam ocorrer no sistema. Esse modelo pesa mais ou menos 375 kg e é preparado para enfrentar temperaturas extremas, que variam de $-20\text{ }^{\circ}\text{C}$ a $60\text{ }^{\circ}\text{C}$ (figura 2).

Figura 2 - Cobham Teodor: robô antibomba



Fonte: Tecmundo.com.br, 2022

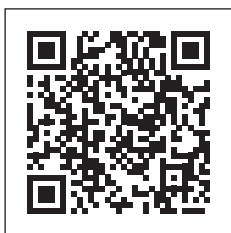
Nesta aula, iremos trabalhar com um protótipo utilizando o Arduino, o Módulo RF433 que poderá enviar sinal ao robô para que receba os comandos e execute ações desejadas pelo usuário por meio do Joystick, indo para frente, para trás, esquerda ou direita.

Caso queira retomar o uso dos Módulos RF433, rever as Aulas 32 e 33 - Módulo RF 433HMz, deste módulo.



Para Saber Mais...

Assista à chegada do Robô Perseverance, da Nasa, ao solo de Marte



<https://www.youtube.com/watch?v=s5mpGncr7EE>

Leia a reportagem sobre “Robô Perseverance da Nasa pousa em Marte após sete meses de viagem”



<https://g1.globo.com/ciencia-e-saude/noticia/2021/02/18/robo-perseverance-da-nasa-pousa-em-marte.ghtml>



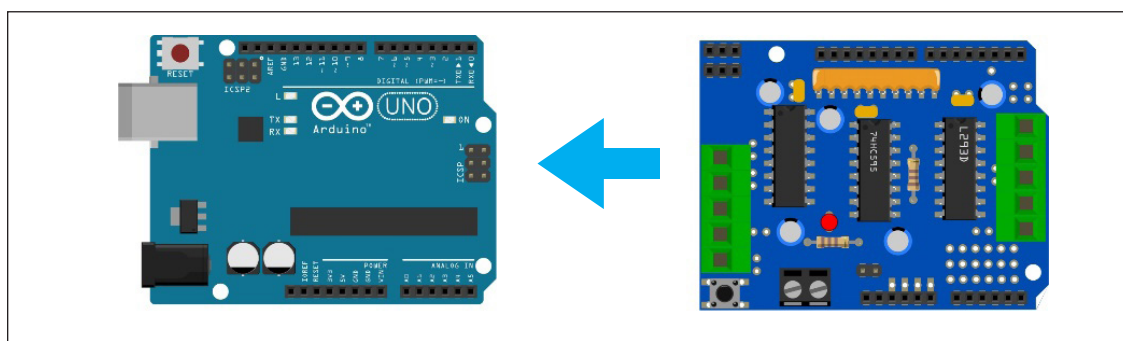
2. Montagem e Programação:

Para este projeto serão necessárias duas montagens, uma que envolve o emissor de ondas (montagem 1) e a outra que envolve o receptor para realizar os comandos (montagem 2). Para tanto, faremos a mesma dinâmica trabalhada na **Aula 33 - Módulo RF 433MHz - II**, onde duas equipes deverão trabalhar juntas para a montagem dos dois dispositivos.

2.1 Montagem do Receptor

Inicie a montagem encaixando a placa Motor Shield sobre a placa Arduino, como indicado na figura 3.

Figura 3 - Encaixe da placa Motor Shield sobre o Arduino



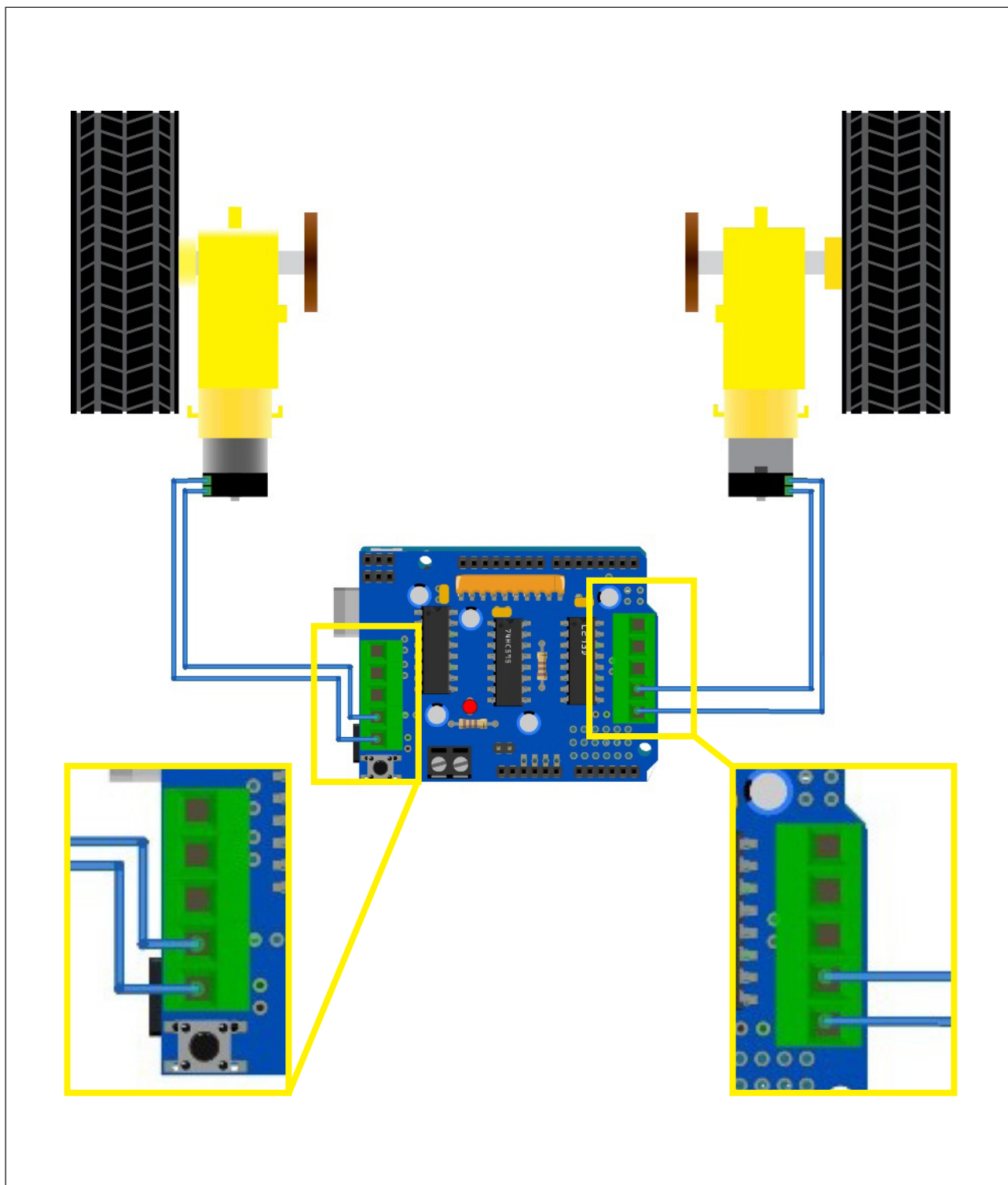
Fonte: Fritzing, 2022

Com auxílio de uma pequena chave, fixe os terminais dos Motores do Chassi 2WD nos conectores M2 e M3 da placa Motor Shield, como indicado na figura 4.



Atenção, não se preocupe com a ordem dos fios, caso os motores não estejam girando na direção correta, é possível fazer a inversão dos fios posteriormente.

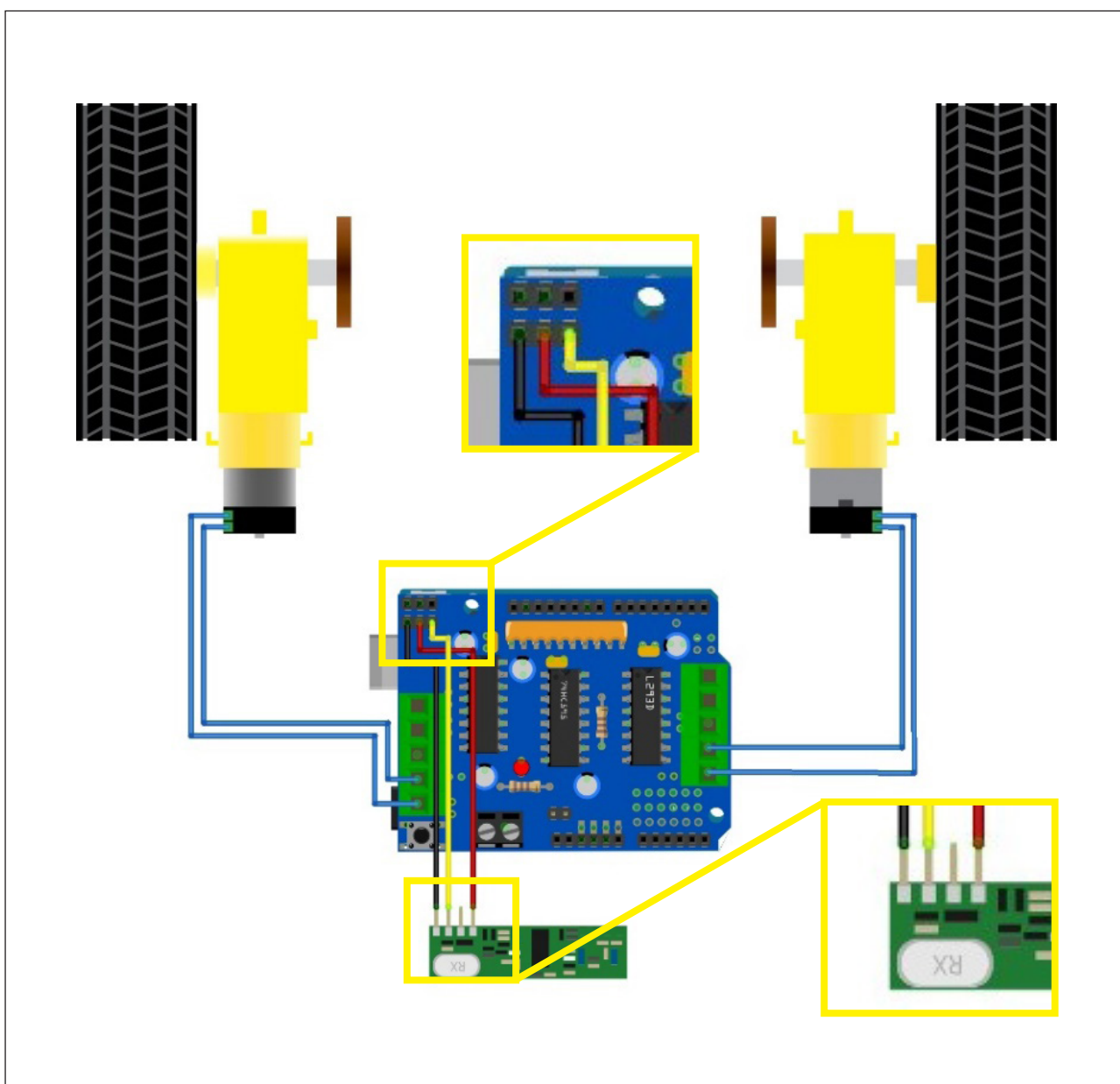
Figura 04 - Conectando os motores à Placa Motor Shield



ROBÔ RÁDIO CONTROLADO

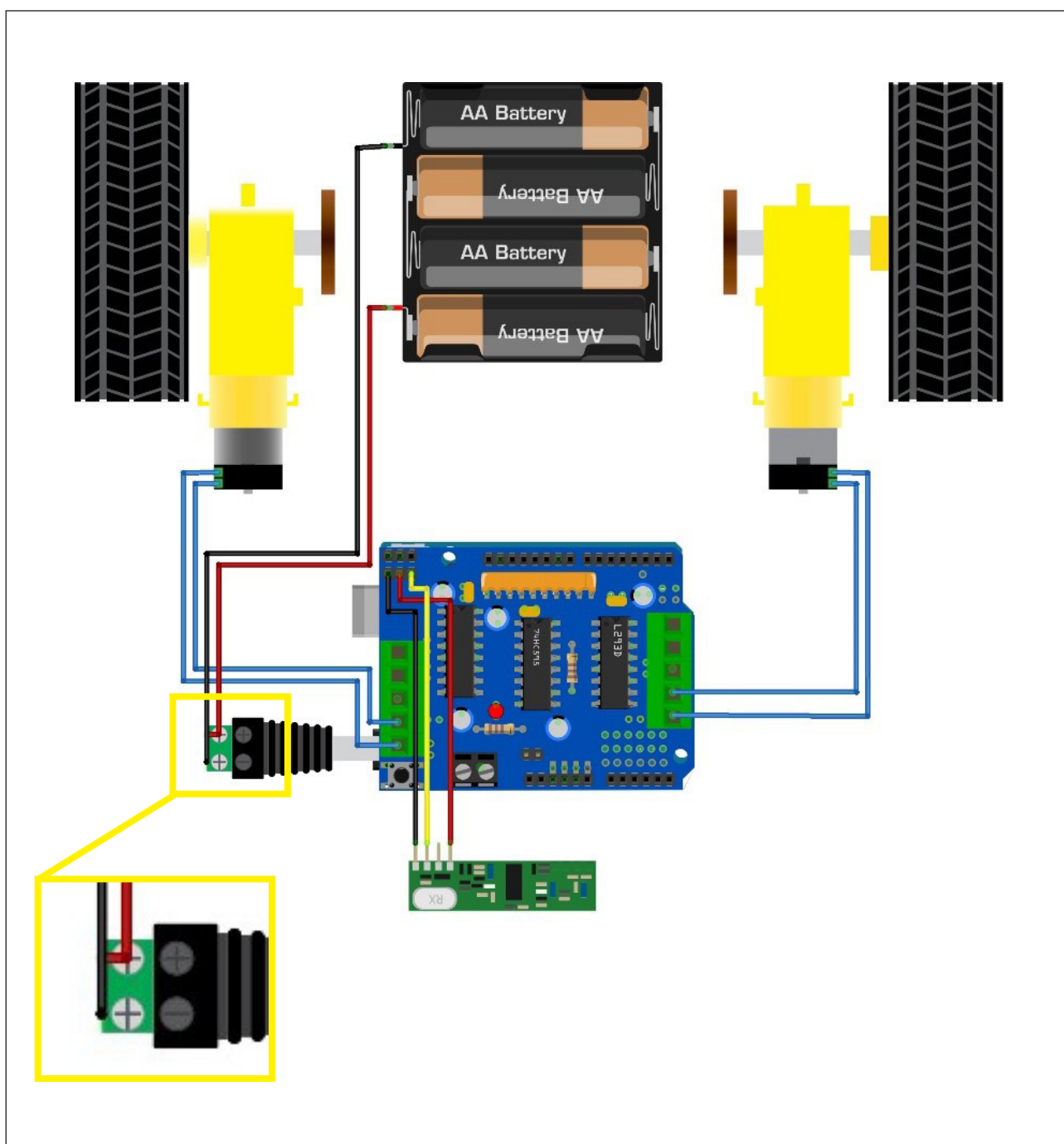
Utilize 3 Jumpers Fêmea-Fêmea para conectar o Módulo Receptor RF 433MHz à placa Motor Shield da seguinte maneira: 1 Jumper do pino - (com sinal negativo) da Motor Shield ao pino GND do Módulo RF, 1 Jumper do pino + da Motor Shield ao pino VCC do Módulo RF e outro Jumper do pino S da Motor Shield ao pino de dados do Módulo RF (qualquer um dos dois pinos centrais), como indicado na figura 5.

Figura 5 - Conectando o Módulo Receptor RF 433MHz à placa Motor Shield



Para finalizar a montagem do emissor, utilize o suporte de pilhas do Chassi para alimentar o circuito elétrico do seu robô, como indicado na figura 6.

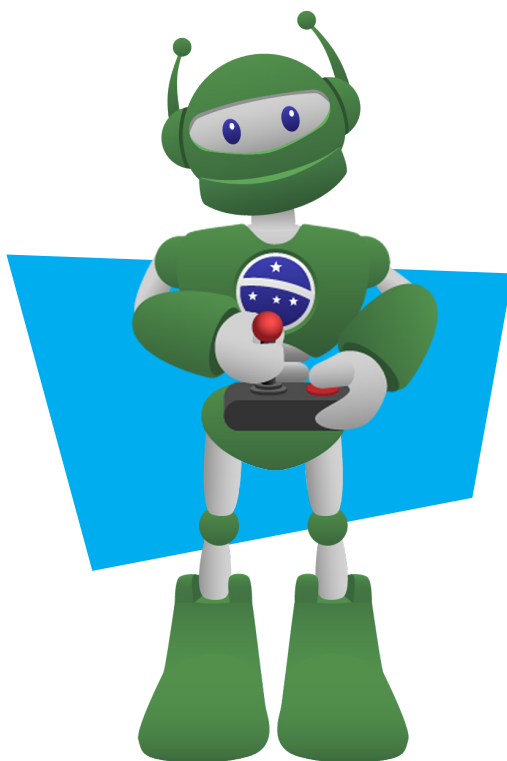
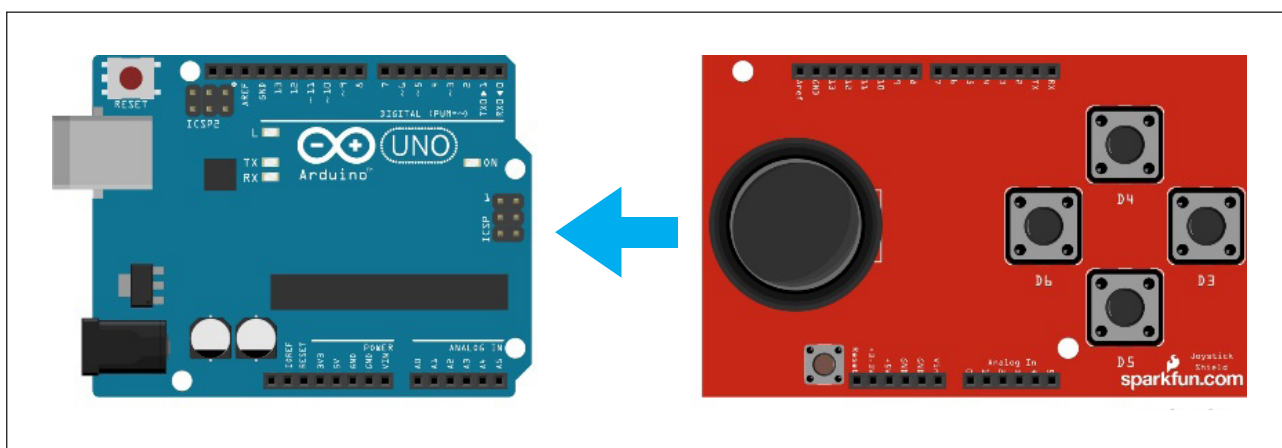
Figura 6 - Alimentação do circuito do robô



2.2 Montagem do Emissor de Ondas

Inicie a montagem do emissor para controlar o seu robô encaixando a placa Joystick Shield sobre a placa Arduino, como indicado na figura 7.

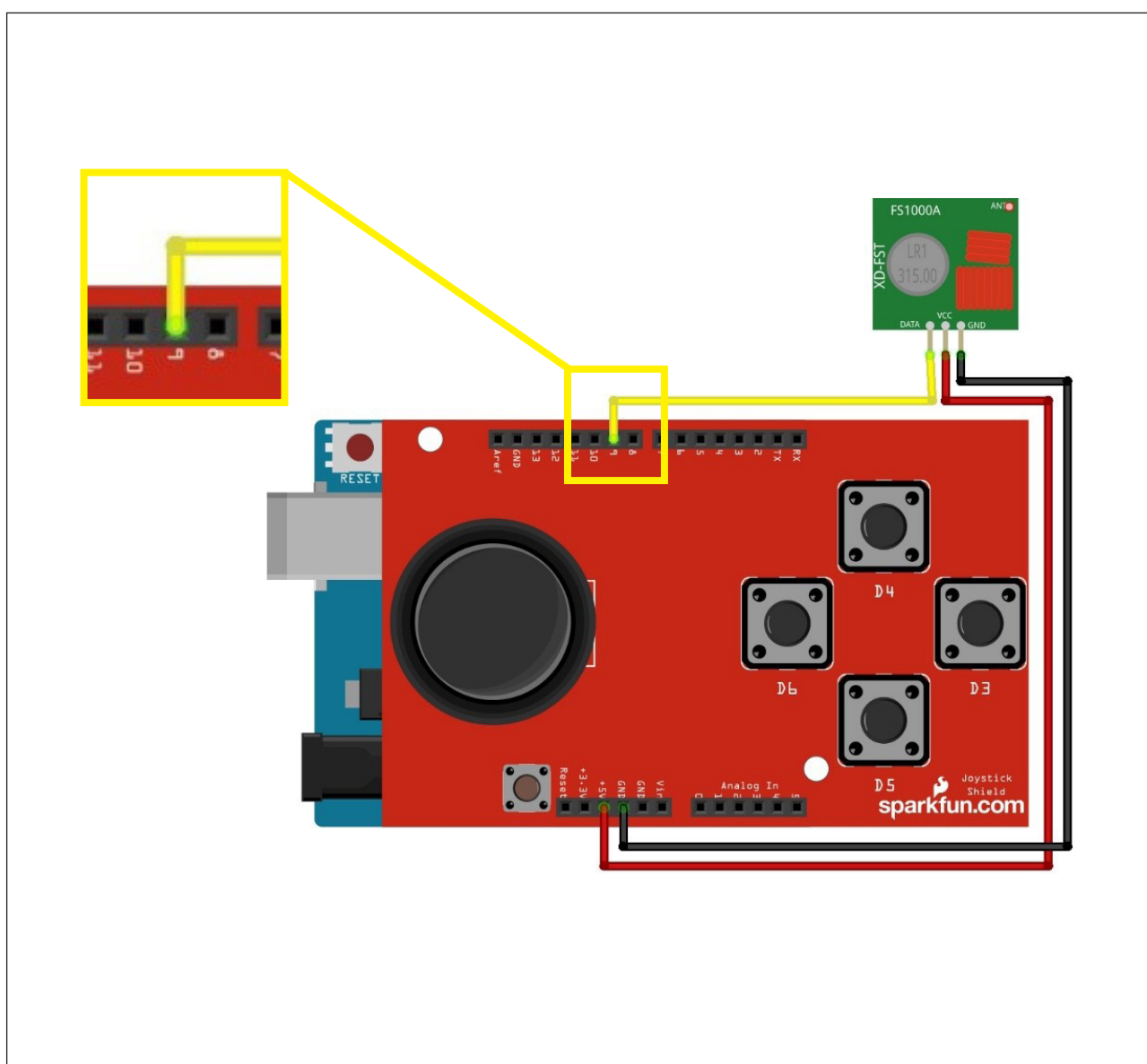
Figura 7 - Encaixe da placa Joystick Shield sobre o Arduino



ROBÔ RÁDIO CONTROLADO

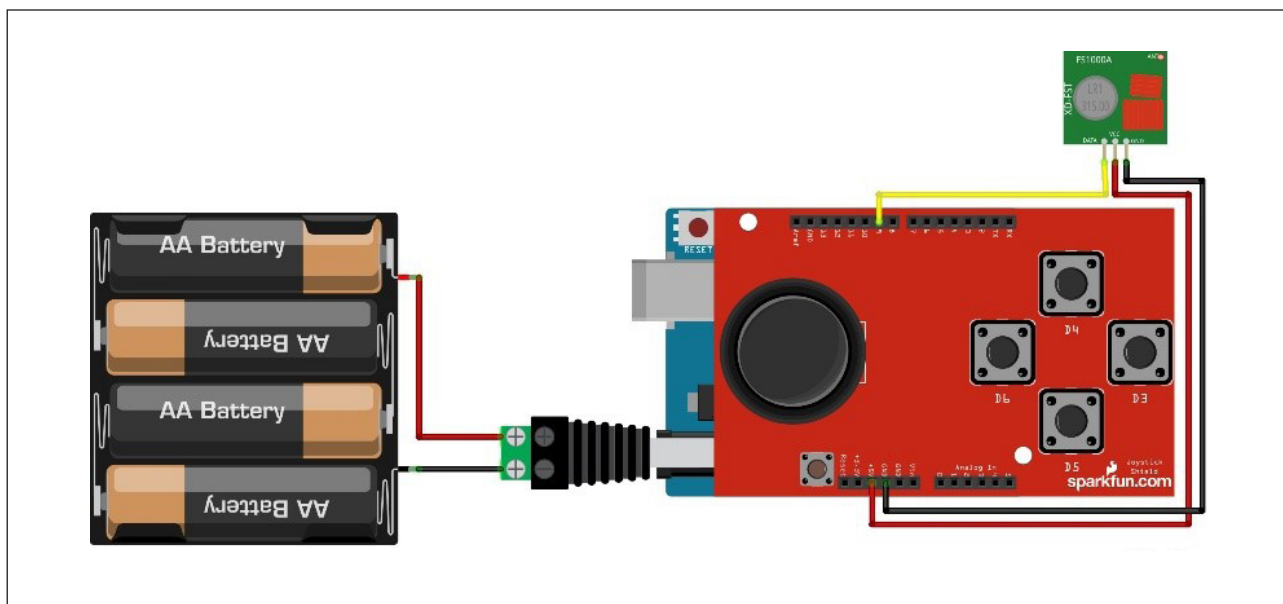
Utilize 3 Jumpers Fêmea-Fêmea para conectar o Módulo Transmissor RF 433MHz à placa Joystick Shield da seguinte maneira: 1 Jumper do pino GND da Joystick Shield ao pino GND do Módulo RF (pino da direita), 1 Jumper do pino 5V da Joystick Shield ao pino VCC do Módulo RF (pino central) e outro Jumper da porta digital D9 da Joystick Shield ao pino DATA do Módulo RF (pino da esquerda), como indicado na figura 8.

Figura 8 - Conectando o Transmissor RF 433MHz à placa Joystick Shield



Para finalizar a montagem, utilize um suporte de pilhas do Chassi para alimentar o circuito elétrico do seu robô, como indicado na figura 9.

Figura 9 - Alimentação do circuito do robô



Fonte: Fritzing, 2022



Agora, vamos programar!

Nesta aula, novamente, utilizaremos duas placas Arduino (uma para transmitir e outra para receber os comandos) sendo necessário que duas equipes trabalhem juntas para a montagem e programação do projeto.

Com os componentes eletrônicos montados, vamos programar nosso Módulo RF 433MHz para transmitir e receber as informações ao robô.

I. Linguagem de programação por código

Para iniciar a programação, conecte a placa Arduino ao computador, através de um cabo USB, para que ocorra a comunicação entre a placa micro controladora e o software Arduino IDE.

No software IDE, crie um sketch e lembre-se de selecionar a porta que o computador atribuiu ao Arduino, então, escreva ou copie e cole o código-fonte de programação, conforme apresentado no quadro 1.

Na linha 13 dos dois códigos, você deverá configurar a mesma velocidade de comunicação entre o Módulo Transmissor e o Módulo Receptor, esse número precisa ser diferente da velocidade definida por outras equipes, isso é necessário para não haver interferência no controle de seus protótipos. Os valores recomendados são: 2000, 2100, 2200, 2300 etc.

Robô - Receptor RF 433MHz

Quadro 1 - Código-fonte da programação na linguagem do Arduino (Wiring)

```
/*
*****
/* Aula 41 - Robô Rádio Controlado - Receptor */
/* Programação do robô constituído pelo Chassi 2WD e */
/* controlado através dos Módulos RF 433MHz e Joystick */
/* Shield. */
*****

/* Define a velocidade de comunicação (em bits por */
/* segundo) entre os Módulos Transmissor e Receptor. */
/* Cada equipe deverá configurar uma velocidade para */
/* evitar a interferência no controle dos seus robôs. */
/* Valores recomendados: 2000, 2100, 2200, 2300 etc. */
#define velocidade 2000

/* Define o pino 9 do Arduino para receber os comandos */
/* do receptor RF 433MHz. */
#define Pino_Receptor 9

/* Inclui a biblioteca de controle do Módulo RF433MHz. */
#include <VirtualWire.h>
```

```
/* Inclui a biblioteca de controle da Motor Shield. */
#include <AFMotor.h>
/* Variável utilizada para formar os comandos */
/* recebidos via RF 433MHz. */
byte comando[VW_MAX_MESSAGE_LEN];
/* Variável para o tamanho dos comandos recebidos. */
byte tamanho_comando = VW_MAX_MESSAGE_LEN;
/* Variável que armazenará o comando completo. */
char recebido;

/* Cria o objeto de controle do motor da direita na */
/* saída 2 da Motor Shield. */
AF_DCMotor Motor_Direita(2);
/* Cria o objeto de controle do motor da esquerda na */
/* saída 3 da Motor Shield. */
AF_DCMotor Motor_Esquerda(3);

/* Função das configurações. */
void setup() {
    /* Configura o pino de entrada de dados do receptor. */
    vw_set_rx_pin(Pino_Receptor);
    /* Configura a velocidade de comunicação já definida.*/
    vw_setup(velocidade);
    /* Inicializa o receptor RF. */
    vw_rx_start();
}
```

```
/* Looping Principal. */
void loop() {
  /* Chama a função auxiliar que receberá os comandos. */
  receber();
  /* Enquanto o comando recebido for "R", faça... */
  while (recebido == 'R') {
    /* Chame a função auxiliar mover para trás. */
    mover_para_tras();
    /* Atualiza o comando recebido. */
    receber();
  }
  /* Enquanto o comando recebido for "F", faça... */
  while (recebido == 'F') {
    /* Chame a função auxiliar mover para frente. */
    mover_para_frente();
    /* Atualiza o comando recebido. */
    receber();
  }
  /* Enquanto o comando recebido for "E", faça... */
  while (recebido == 'E') {
    /* Chame a função auxiliar virar à esquerda. */
    virar_a_esquerda();
    /* Atualiza o comando recebido. */
    receber();
  }
}
```

```
/* Enquanto o comando recebido for "D", faça... */
while (recebido == 'D') {
    /* Chame a função auxiliar virar à direita. */
    virar_a_direita();
    /* Atualiza o comando recebido. */
    receber();
}
/* Enquanto o comando recebido for "N", faça... */
while (recebido == 'N') {
    /* Chame a função auxiliar permanecer parado. */
    permanecer_parado();
    /* Atualiza o comando recebido. */
    receber();
}
/* Se o comando recebido for "1", faça... */
if (recebido == '1') {
    /* Defina a potência dos motores para 150. */
    Motor_Direita.setSpeed(130);
    Motor_Esquerda.setSpeed(130);
}
/* Se o comando recebido for "2", faça... */
if (recebido == '2') {
    /* Defina a potência dos motores para 190. */
    Motor_Direita.setSpeed(155);
    Motor_Esquerda.setSpeed(155);
}
```

```
/* Se o comando recebido for "3", faça... */
if (recebido == '3') {
    /* Defina a potência dos motores para 220. */
    Motor_Direita.setSpeed(220);
    Motor_Esquerda.setSpeed(220);
}
/* Se o comando recebido for "4", faça... */
if (recebido == '4') {
    /* Defina a potência dos motores para 255. */
    Motor_Direita.setSpeed(255);
    Motor_Esquerda.setSpeed(255);
}
}

/* Função auxiliar de captura dos comandos recebidos. */
void receber() {
    if (vw_get_message(comando, &tamanho_comando))
    {
        for (int i = 0; i < tamanho_comando; i++)
        {
            /* Armazena na variável o comando recebido. */
            recebido = comando[i];
        }
    }
}
```

```
/* Função auxiliar para o robô mover para frente.      */
void mover_para_frente() {
    Motor_Direita.run(FORWARD);
    Motor_Esquerda.run(FORWARD);
}
/* Função auxiliar para o robô permanecer parado.      */
void permanecer_parado() {
    Motor_Direita.run(RELEASE);
    Motor_Esquerda.run(RELEASE);
}
/* Função auxiliar para o robô mover para trás.        */
void mover_para_tras() {
    Motor_Direita.run(BACKWARD);
    Motor_Esquerda.run(BACKWARD);
}
/* Função auxiliar para o robô virar à direita.        */
void virar_a_direita() {
    Motor_Direita.run(BACKWARD);
    Motor_Esquerda.run(FORWARD);
}
/* Função auxiliar para o robô virar à esquerda.      */
void virar_a_esquerda() {
    Motor_Direita.run(FORWARD);
    Motor_Esquerda.run(BACKWARD);
}
```

Quadro 2 - Controle do Robô - Transmissor RF 433MHz

```
/* **** */
/* Aula 41 - Robô Rádio Controlado - Transmissor */
/* Programação do robô constituído pelo Chassi 2WD e */
/* controlado através dos módulos RF 433MHz e Joystick */
/* Shield. */
/* **** */

/* Define a velocidade de comunicação (em bits por */
/* segundo) entre os módulos transmissor e receptor. */
/* Cada equipe deverá configurar uma velocidade para */
/* evitar a interferência no controle dos seus robôs. */
/* Valores recomendados: 2000, 2100, 2200, 2300 etc. */
#define velocidade 2000
/* Define o pino 9 do Arduino para receber os comandos */
/* do receptor RF 433MHz. */
#define Pino_Transmissor 9
/* Inclui a biblioteca de controle do módulo RF433MHz. */
#include <VirtualWire.h>
/* Variáveis que armazenarão os dados da Joystick Shield*/
int JoyX;
int JoyY;
int estadoB1;
int estadoB2;
int estadoB3;
int estadoB4;
```



```
/* Função para as configurações. */
void setup()
{
  /* Configura o pino de transmissão de dados. */
  vw_set_tx_pin(Pino_Transmissor);
  /* Configura a velocidade de comunicação já definida.*/
  vw_setup(velocidade);
  /* Ativa as portas dos botões como entradas e ativa */
  /* os Resistores internos. */
  pinMode(2, INPUT_PULLUP);
  pinMode(3, INPUT_PULLUP);
  pinMode(4, INPUT_PULLUP);
  pinMode(5, INPUT_PULLUP);
}
/* Looping Principal. */
void loop() {
  /* Chama a função auxiliar para verificar o estado */
  /* da Joystick Shield. */
  checar_joystick();

  /* Enquanto o eixo do Joystick estiver à esquerda, faça... */
  while (JoyX < 300) {
    /* Envie para a função auxiliar o caractere E. */
    enviar_comando("E");
  }
}
```

```
/* Enquanto o eixo do Joystick estiver à direita, faça... */
while (JoyX > 700) {
    /* Envie para a função auxiliar o caractere D. */
    enviar_comando("D");
}

/* Enquanto o eixo do Joystick estiver para trás, faça... */
while (JoyY < 300) {
    /* Envie para a função auxiliar o caractere R. */
    enviar_comando("R");
}

/* Enquanto o eixo do Joystick estiver para frente, faça... */
while (JoyY > 700) {
    /* Envie para a função auxiliar o caractere F. */
    enviar_comando("F");
}

/* Se pressionar o botão A do Joystick, faz... */
if (estadoB1 == LOW) {
    /* Envie para a função auxiliar o número 4. */
    enviar_comando("4");
}

/* Se pressionar o botão B do Joystick, faz... */
if (estadoB2 == LOW) {
    /* Envie para a função auxiliar o número 3. */
    enviar_comando("3");
}
```

```
/* Se pressionar o botão C do Joystick, faz... */
if (estadoB3 == LOW) {
    /* Envie para a função auxiliar o número 2. */
    enviar_comando("2");
}

/* Se pressionar o botão D do Joystick, faz... */
if (estadoB4 == LOW) {
    /* Envie para a função auxiliar o número 1. */
    enviar_comando("1");
}

/* Se o Joystick permanecer no centro, faz... */
if (( 300 < JoyX < 700) && ( 300 < JoyY < 700)) {
    /* Envie para a função auxiliar o caractere N. */
    enviar_comando("N");
}
}

/* Função auxiliar que realiza a leitura do Joystick. */
void checar_joystick() {
    JoyX = analogRead(A0);
    JoyY = analogRead(A1);
    estadoB1 = digitalRead(2);
    estadoB2 = digitalRead(3);
    estadoB3 = digitalRead(4);
    estadoB4 = digitalRead(5);
}
```

```
/* Função auxiliar para envio dos caracteres.          */
void enviar_comando(const char *msg) {
    /* Enviando o caractere.                            */
    vw_send((uint8_t *)msg, strlen(msg));
    /* Aguarda até que o caractere seja enviado.       */
    vw_wait_tx();
    /* Chama a função auxiliar para verificar o estado */
    /* do Joystick Shield.                             */
    checar_joystick();
}
```

Com o código-fonte inserido no Arduino IDE, compile o programa pressionando o botão **Verificar** para averiguar se não há erros de sintaxe. Estando o código correto, o próximo passo é realizar a transferência do programa para o Arduino. Pressione o botão **Carregar**, para realizar upload do programa para o Arduino.

Após a transferência do programa para o Arduino, o módulo RF 433 estará com o transmissor e o receptor do módulo em funcionamento para que o robô obedeça aos comandos do Joystick, deslocando-se para a esquerda, para a direita, para frente, para trás e parando.



Desafio:

Que tal acrescentar recursos adicionais ao seu robô? Por exemplo, uma buzina com o Buzzer, LEDs para indicar a intensão de conversões, vamos lá! Dê um toque exclusivo para seu robô rádio controlado!



E se...?

O projeto não funcionar, atente-se 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. Verifique se a programação está adequada às portas digitais.



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 aos requisitos para programar o Robô Rádio Controlado.

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?

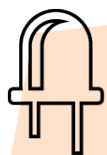
ii. Pensamento Crítico e Resolução de Problemas: você conseguiu identificar os problemas, analisar informações e tomar decisões de modo a contribuir para o projeto desenvolvido?

d. Reúna todos os componentes utilizados nesta aula e os organize novamente, junto aos demais, no kit de robótica.

AULA

41

ROBÔ RÁDIO CONTROLADO



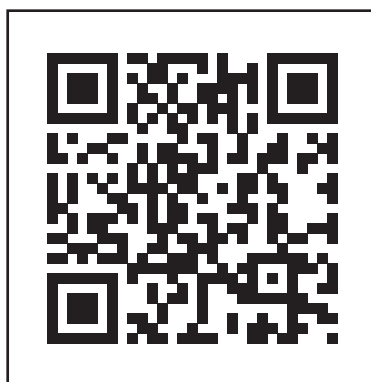
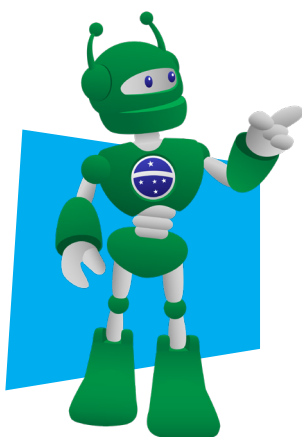
Videotutorial

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



<https://rebrand.ly/a41robotica2>

Acesse, também, pelo QRCode:





Referências

ARDUINO. Site oficial. **Ambiente de Programação do Arduino**. Disponível em: <https://create.arduino.cc/editor>. Acesso em: 14 abr. 2022.

ARDUINO. Site oficial. **Documentação de Referência da Linguagem Arduino**. Funções. Disponível em: <https://www.arduino.cc/reference/pt/>. Acesso em: 16 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: 10 abr. 2022.

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

MODELLI, Laís. **Ciência e Saúde**. G1. Robô perseverance da Nasa pousa em mar-te. Disponível em: <https://g1.globo.com/ciencia-e-saude/noticia/2021/02/18/robo-perseverance-da-nasa-pousa-em-marte.ghtml>. Acesso em: 10 mar. 2022.

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: 22 abr. 2022.

TECMUNDO. **Como funcionam os robôs antibombas?** Disponível em: <https://www.tecmundo.com.br/robotica/38103-como-funcionam-os-robos-antibombas-.htm>. Acesso em: 12 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 - Compartilhável 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

