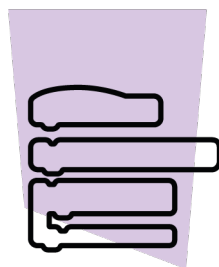
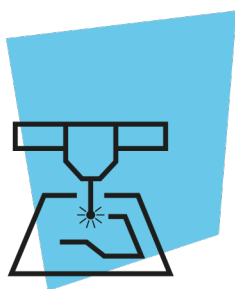
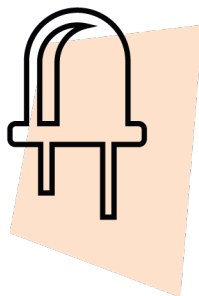
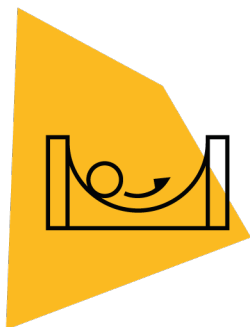
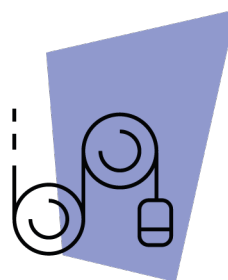
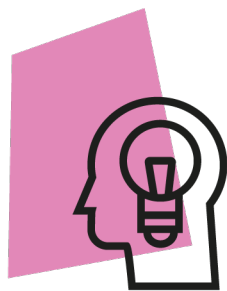


ROBÓTICA

Módulo 2



Braço Robótico Via RF

AULA 40

GOVERNADOR DO ESTADO DO PARANÁ

Carlos Massa Ratinho Júnior

SECRETÁRIO DE ESTADO DA EDUCAÇÃO

Renato Feder

DIRETOR DE TECNOLOGIA E INOVAÇÃO

Andre Gustavo Souza Garbosa

COORDENADOR DE TECNOLOGIAS EDUCACIONAIS

Marcelo Gasparin

Produção de Conteúdo

Cleiton Rosa

Michelle dos Santos

Validação de Conteúdo

Cleiton Rosa

Revisão Textual

Adilson Carlos Batista

Leitura Crítica e Normalização Bibliográfica

Ricardo Hasper

Projeto Gráfico e Diagramação

Edna do Rocio Becker

Ilustração

Jocelin Vianna (Educa Play)

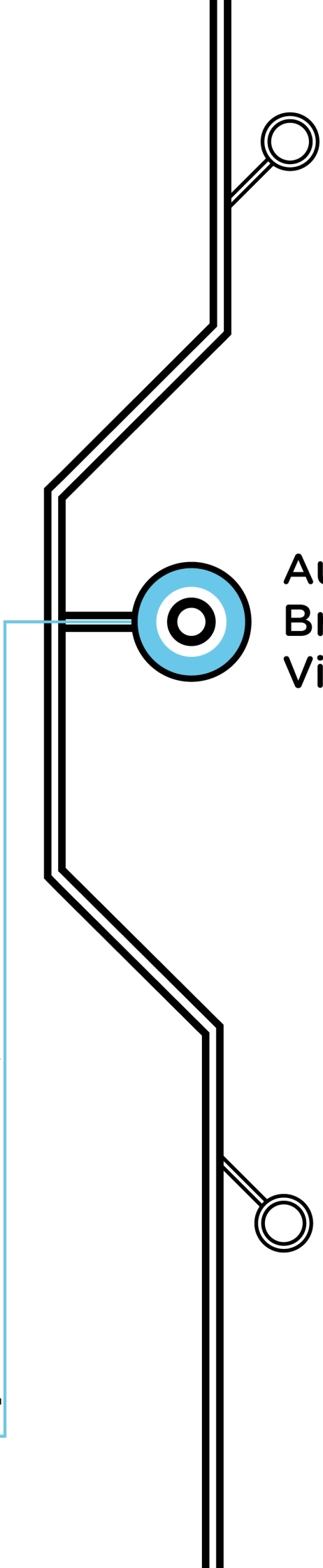
2022

- 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 39 Robô Wireless

Aula 40 Braço Robótico Via RF

Aula 41 Robô Radio Controlado



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	25
Videotutorial	26
Referências	27



Introdução

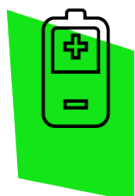
Em um mundo altamente digital, sistemas automatizados como o Braço Robótico pode contribuir para evitar a exposição do homem às atividades de risco que podem causar sérios danos à sua saúde, bem como utilizá-lo para alcançar objetos ou executar funções mais onerosas, seja no comércio ou até mesmo em nossas casas.

Nesta aula, aprenderemos a manipular o Braço Robótico através de controle remoto utilizando a Radiofrequência.



Objetivos desta Aula

- Desenvolver um novo sistema de manipulação do Braço Robótico;
- Buscar novas aplicações do Braço Robótico;
- Estimular o estudo para novos projetos envolvendo o Braço Robótico e a Robótica.



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.





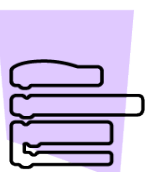
Habilidades do Século XXI a Serem Desenvolvidas

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



Lista de Materiais

- 02 Placas Arduino Uno R3;
- 01 Módulo RF 433mhz;
- 01 Sensor Shield V5.0;
- 01 Joystick Shield;
- 01 Braço Robótico;
- 06 Jumpers Fêmea-Fêmea;
- 03 Jumpers Macho-Macho;
- 01 Cabo USB;
- Notebook;
- Software Arduino IDE.



Roteiro da Aula

1. Contextualização:

As tecnologias vêm para trazer muito mais do que facilidades e conforto, podendo suprir a necessidade de programar remotamente qualquer objeto, como, por exemplo, um Braço Robótico.

Controlar e poder monitorar um Braço Robótico remotamente possibilita realizar o manuseio de uma variedade de objetos a uma distância segura e rápida.

Imagine desenvolver um protótipo manipulável que possa auxiliar em atividades em ambientes e demandas hostis para a vida humana, assim como áreas inacessíveis ou mesmo adaptá-lo para o uso em processos que utilizem máquinas e equipamentos passíveis de controle remotos. Estes manipuláveis podem ser usados como barreiras adicionais à disseminação de doenças contagiosas, protegendo e evitando a população da exposição à contaminação.

Para otimizar a funcionalidade do Braço Robótico, nesta aula, utilizaremos como controle remoto o Joystick Shield, estudado na **Aula 19 – Joystick Shield**, operando, neste projeto, via Radiofrequência.

Diferenças entre Controle Remoto Infravermelho (Ir) e Radiofrequência (Rf)

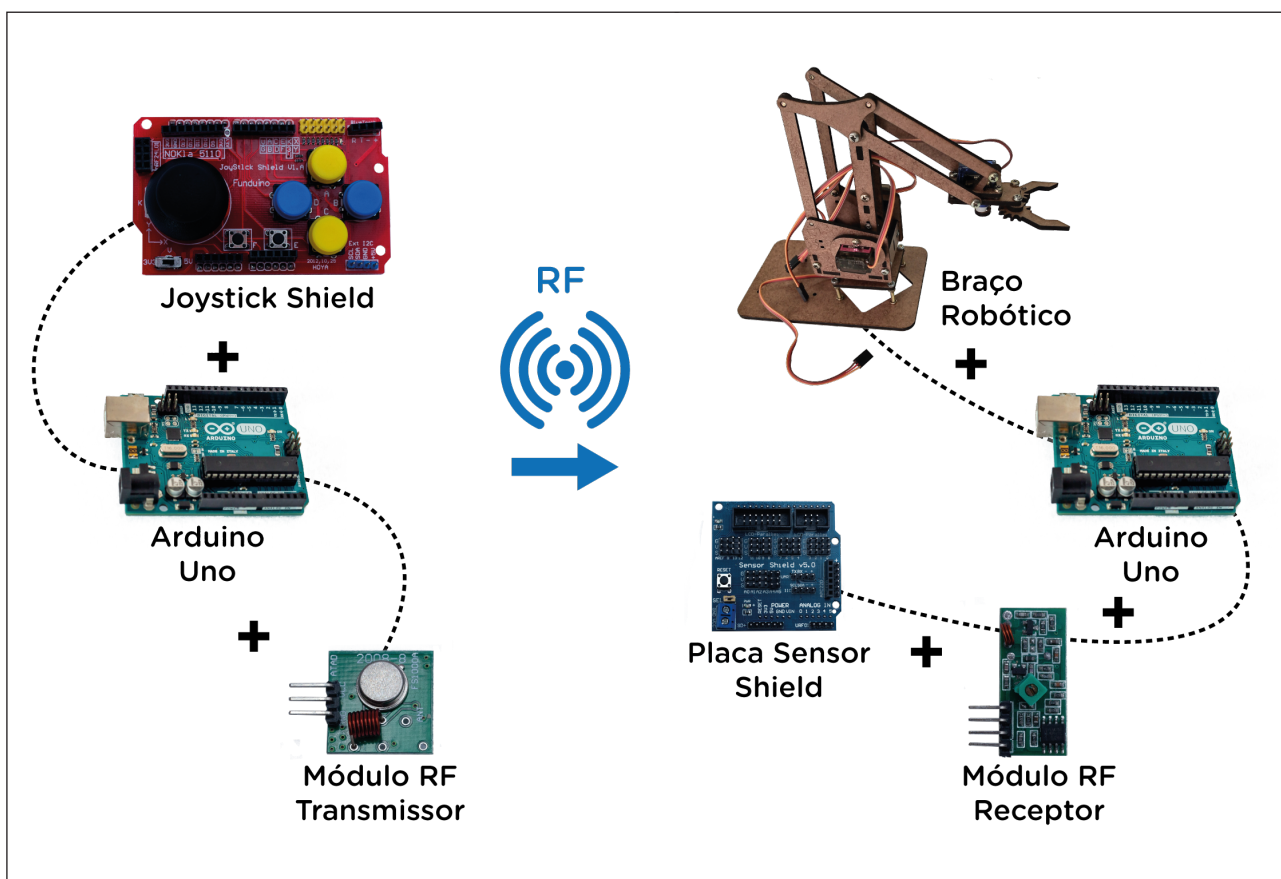
O controle remoto ou comando à distância é um dispositivo eletrônico usado para realizar uma operação remota sobre um dispositivo. Podem operar em Infravermelho (Ir) ou Radiofrequência (Rf). Afinal, qual o melhor?

Os controles remotos de radiofrequência são mais utilizados, pois é preciso um leque de ação superior, por possuir maior longitude de ondas que as infravermelhas. Os infravermelhos são recomendados quando se utiliza apenas um dispositivo, visto que não abrem tanto o sinal que emite como para alcançar vários dispositivos.



Manipulando as chaves botão ou alavanca do Joystick Shield, você verificará que, conforme o movimento, ocorrerá a mobilidade do Braço Robótico. As conexões do Joystick Shield serão responsáveis por “capturar” o movimento feito e encaminhar o comando para o Arduino, que emitirá um comando de movimento para os Servos Motores. Enquanto o Joystick Shield for acionado para uma das posições, o Servo Motor estará se movimentando. Quando parar de acionar o Joystick Shield, o Servo Motor também irá parar (figura 1).

Figura 1 - Esquema de funcionamento da aula



Fonte: SEED/DTI/CTE, 2022



Para Saber Mais...

Braço Robótico faz ultrassom a distância durante pandemia.



<https://canaltech.com.br/espaco/telemedicina-espacial-braco-robotico-faz-ultrassom-a-distancia-durante-pandemia-164431/>

Einstein utiliza Braço Robótico para tratar paciente com Covid infartado.

<https://www1.folha.uol.com.br/equilibrioesaude/2020/06/einstein-utiliza-braco-robotico-para-tratar-paciente-com-covid-infartado.shtml>



2. Montagem e Programação:

Como neste projeto são utilizadas duas placas Arduino Uno (uma para o controle e outra para o Braço Robótico), utilizaremos nesta aula a mesma dinâmica trabalhada na **Aula 33 - Módulo RF 433MHz - II**, onde duas equipes deverão trabalhar juntas para a montagem dos dois dispositivos.

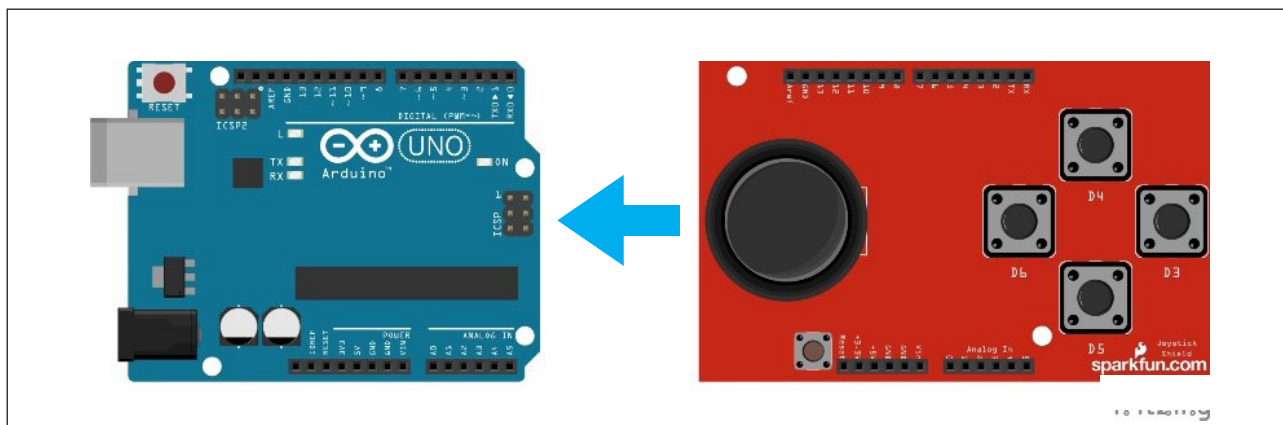
A montagem será realizada em duas etapas: montagem do Controle e montagem do Braço Rádio Controlado.

2.1 Montagem do Controle

Inicie a montagem preparando 3 Jumpers Macho-Fêmea, conectando 3 Jumpers Macho-Macho a outros 3 Jumpers Fêmea-Fêmea e reserve-os.

A seguir, encaixe a placa Joystick Shield sobre a placa Arduino, como mostrado na figura 2

Figura 2 - Encaixe da Joystick Shield sobre o Arduino

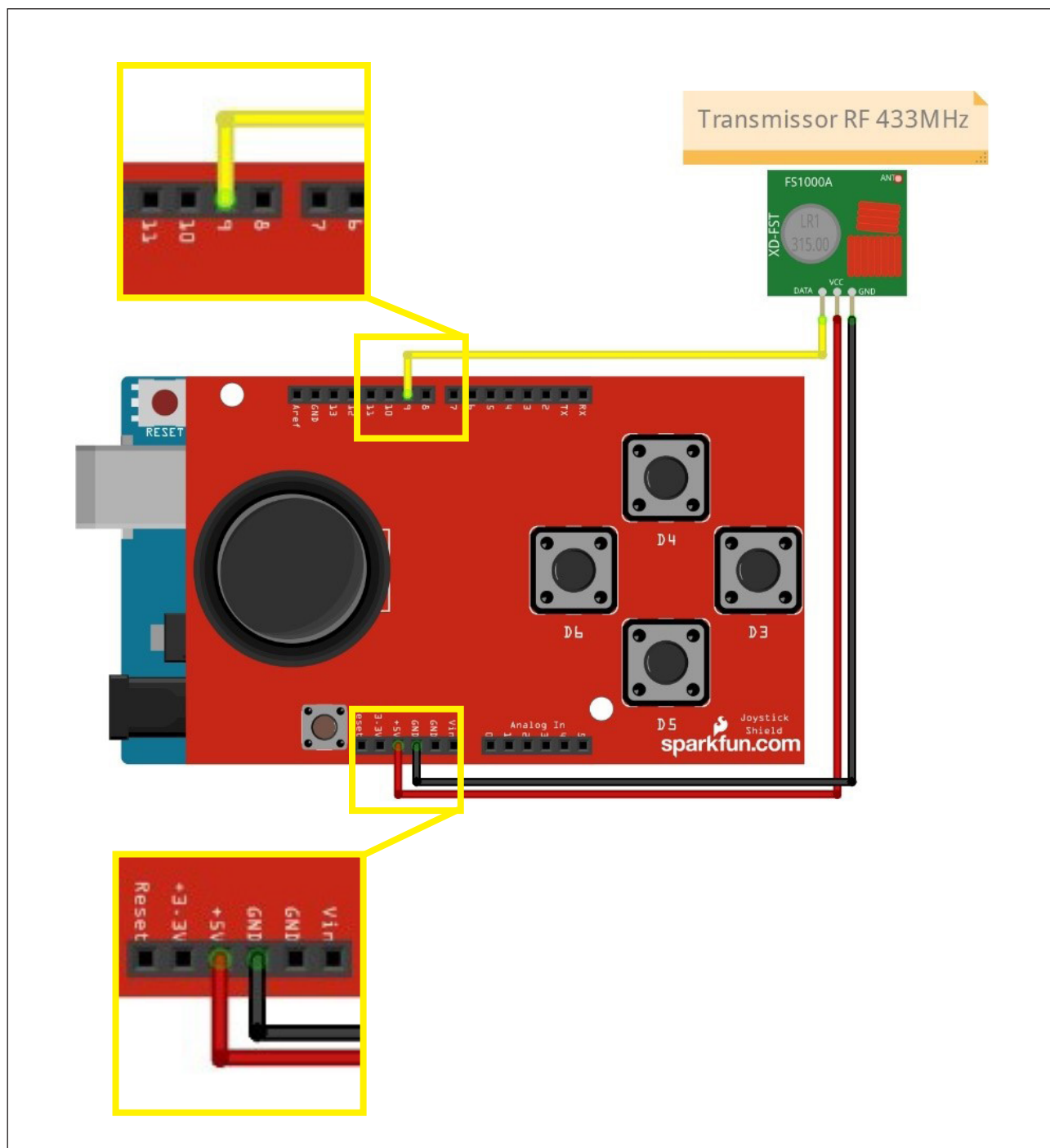


Fonte: Fritzing, 2022

Finalize a montagem conectando o Módulo Transmissor RF 433MHz da seguinte maneira: 1 Jumper Macho-Fêmea entre os pinos GND do Módulo e GND da placa Joystick Shield, 1 Jumper Macho-Fêmea entre os pinos VCC do Módulo e 5V da Joystick Shield e o outro Jumper Macho-Fêmea entre os pinos DATA do Módulo e D9 da Joystick Shield (figura 3).



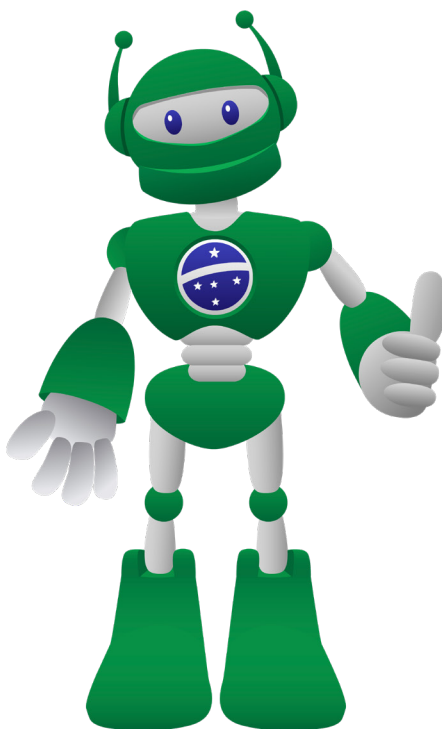
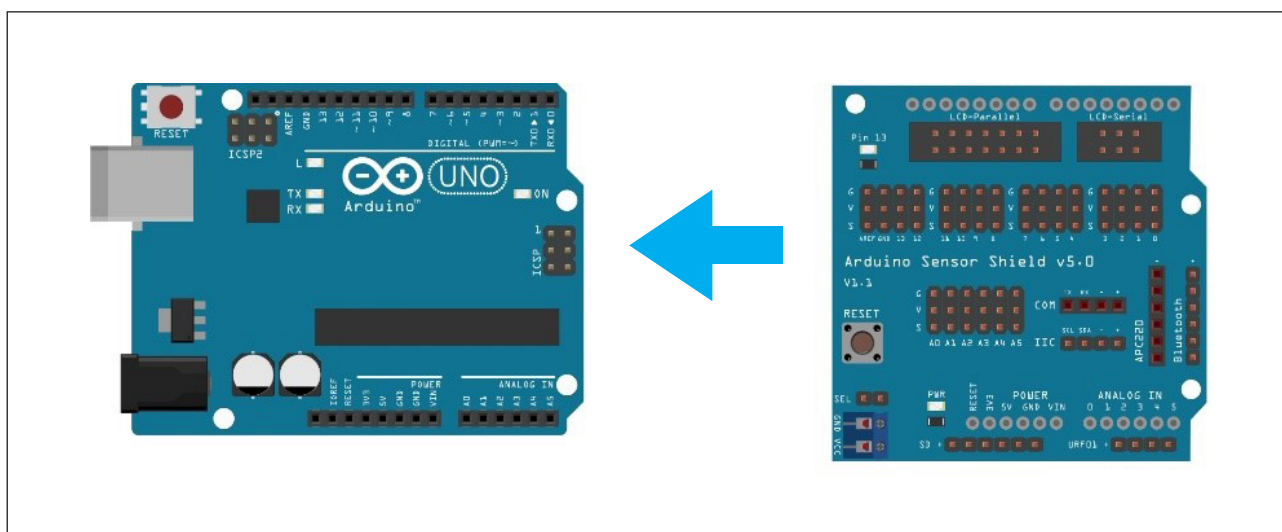
Figura 03 - Conexão do Módulo Transmissor RF 433MHz à Joystick Shield



2.1 Montagem do Braço Rádio Controlado

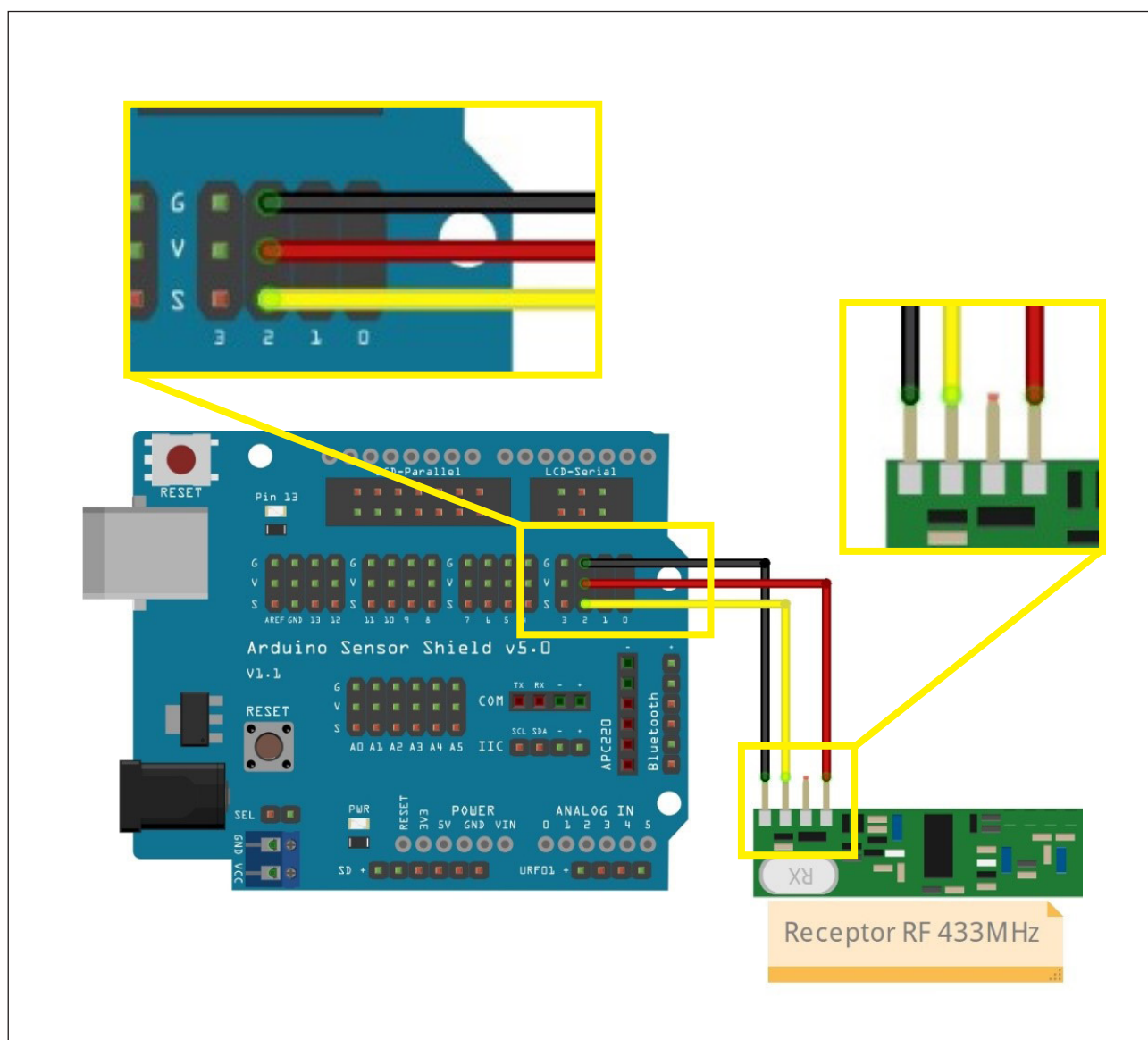
Inicie a montagem encaixando sobre a placa Arduino a placa Sensor Shield V5.0, como mostrado na figura 4.

Figura 4 - Encaixe do Sensor Shield sobre o Arduino



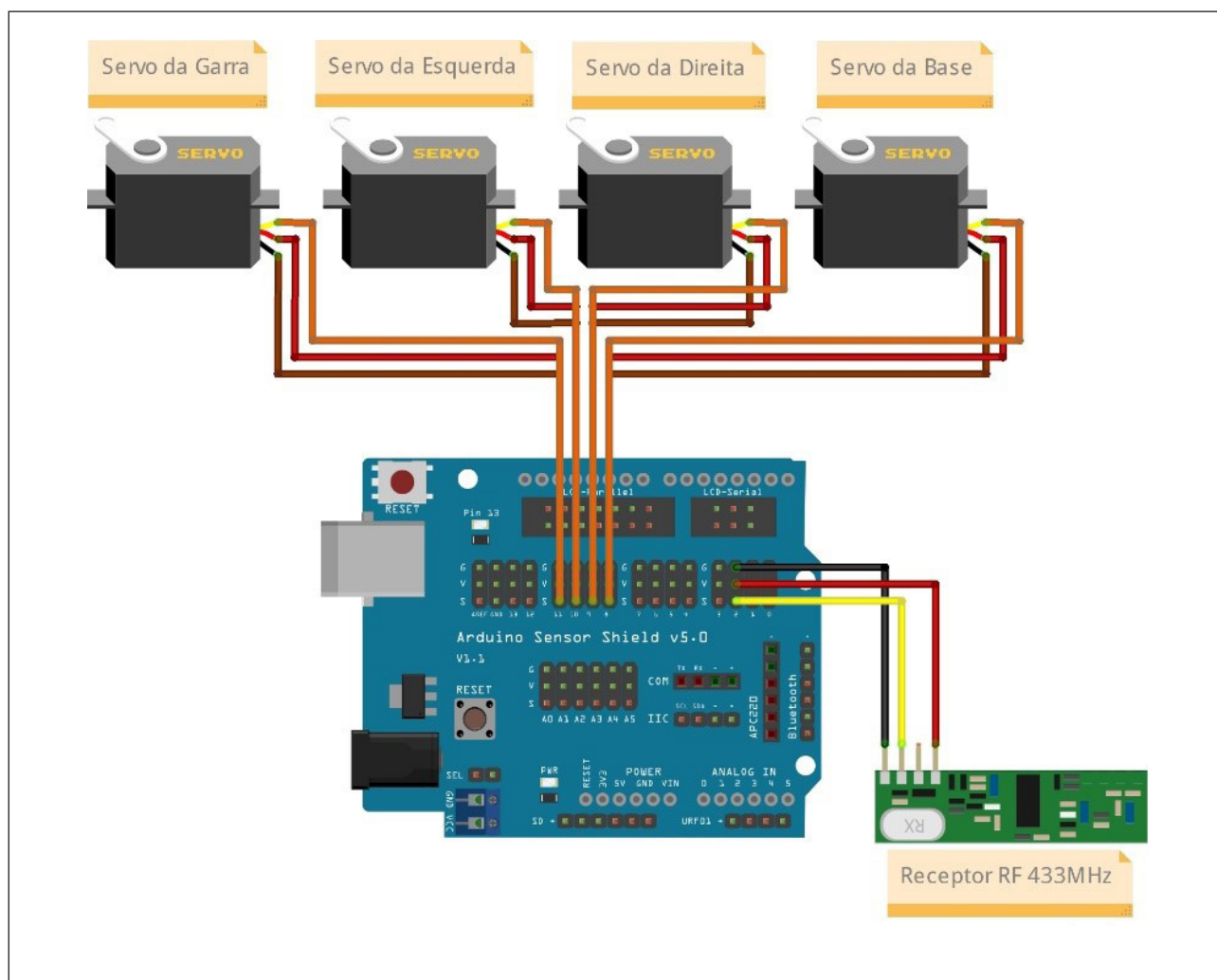
Utilizando 3 Jumpers Fêmea-Fêmea, conecte o Módulo Receptor RF 433MHz da seguinte maneira: 1 Jumper entre os pinos GND do Módulo e G da placa Sensor Shield, 1 Jumper entre os pinos VCC do Módulo e V da Sensor Shield, e o outro Jumper entre os pinos DATA do Módulo (qualquer um dos dois pinos centrais) e o S da coluna de pinos 2 da Sensor Shield (figura 5).

Figura 5 - Conexão do Módulo Receptor RF 433MHz à placa Sensor Shield



Finalize a montagem conectando os 4 Servos Motores à placa Sensor Shield da seguinte maneira: o terminal do Servo da Base nos três pinos da porta digital 8 (a posição correta é fio laranja na linha S), o terminal do Servo da Direita nos três pinos da porta digital 9 (a posição correta é fio laranja na linha S), o terminal do Servo da Esquerda nos três pinos da porta digital 10 (a posição correta é fio laranja na linha S) e o terminal do Servo da Garra nos três pinos da porta digital 11 (a posição correta é fio laranja na linha S), como mostrado na figura 6.

Figura 6 - Conexão dos Servos Motores à placa Sensor Shield



Fonte: Fritzing, 2022



Agora, vamos programar!

Com os componentes eletrônicos montados, vamos programar nosso protótipo por codificação.

Nesta aula, serão utilizadas duas programações, uma para o controle do braço e outra para o receptor do Braço Robótico. Em ambas as programações, utilizaremos a biblioteca “**VirtualWire**” que auxiliará no controle dos Servos Motores (para recordar como instalar uma biblioteca, consulte a **Aula 05 - Softwares Arduino IDE e mBlock** do módulo 1).

Realize o download da biblioteca através do link: <https://github.com/song940/VirtualWire>

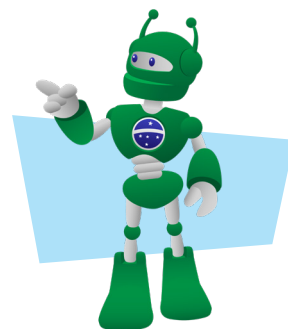
Para iniciar a programação, conecte a placa Arduino ao computador, através de um cabo USB, a fim de que ocorra a comunicação entre a placa Microcontroladora e o Arduino IDE.

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

Na linha 14 dos dois códigos, você deverá configurar a mesma velocidade de comunicação entre o módulo Transmissor e o Módulo Receptor, entretanto, deverá 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.

Atenção!

Ao copiar o código diretamente do pdf, evite quebra da página (e conseqüentemente erros na compilação), copiando o código por partes.



Quadro 1 - Linguagem de programação por código -
Controle do Braço Robótico

```
/* **** */
/* Aula 40 - Braço Robótico Rádio Controlado */
/* Programação do controle do braço robótico. */
/* Os quatro botões selecionam o movimento dos quatro */
/* servos motores e os eixos do joystick, enviam os */
/* comandos para movê-los. */
/* **** */

/* Definição da 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 braços. */
/* Valores recomendados: 2000, 2100, 2200, 2300, etc. */
#define velocidade 2000
/* Define o pino digital 9 do Arduino para enviar os */
/* comandos ao módulo transmissor RF 433MHz. */
#define Pino_Transmissor 9
/* Inclui a biblioteca de controle do módulo RF433MHz. */
#include <VirtualWire.h>
/* Criando as variáveis que armazenarão os dados dos */
/* eixos do joystick e dos quatro botões. */
int JoyX;
int JoyY;
int estadoBotaoA;
int estadoBotaoB;
int estadoBotaoC;
int estadoBotaoD;
/* Função para configurações. */
void setup()
```

```
{
  /* Função que informa à biblioteca qual o pino          */
  /* utilizado para a transmissão.                        */
  vw_set_tx_pin(Pino_Transmissor);
  /* Função que informa à biblioteca qual a velocidade */
  /* utilizada para a transmissão dos dados.          */
  vw_setup(velocidade);
  /* Configura as portas digitais de cada botão como  */
  /* entradas ativando 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 que realiza a leitura da */
  /* Joystick Shield.                                  */
  checar_joystick();
  /* Enquanto o eixo do joystick for deslocado para a */
  /* esquerda, faça...                                  */
  while (JoyX < 300) {
    /* Chame a função auxiliar para enviar o comando x.*/
    enviar_comando("x");
  }
  /* Enquanto o eixo do joystick for deslocado para a */
  /* direita, faça...                                  */
  while (JoyX > 700) {
    /* Chame a função auxiliar para enviar o comando X.*/
    enviar_comando("X");
  }
}
```

```
/* Enquanto o eixo do joystick for deslocado para */
/* trás, faça...*/
while (JoyY < 300) {
    /* Chame a função auxiliar para enviar o comando y.*/
    enviar_comando("y");
}
/* Enquanto o eixo do joystick for deslocado para */
/* frente, faça... */
while (JoyY > 700) {
    /* Chame a função auxiliar para enviar o comando Y.*/
    enviar_comando("Y");
}
/* Enquanto o botão A do joystick for pressionado, */
/* faça... */
while (estadoBotaoA == LOW) {
    /* Chame a função auxiliar para enviar o comando G.*/
    enviar_comando("G");
}
/* Enquanto o botão B do joystick for pressionado, */
/* faça... */
while (estadoBotaoB == LOW) {
    /* Chame a função auxiliar para enviar o comando D.*/
    enviar_comando("D");
}
/* Enquanto o botão C do joystick for pressionado, */
/* faça... */
while (estadoBotaoC == LOW) {
    /* Chame a função auxiliar para enviar o comando B.*/
    enviar_comando("B");
}
/* Enquanto o botão D do joystick for pressionado, */
/* faça... */
while (estadoBotaoD == LOW) {
    /* Chame a função auxiliar para enviar o comando E.*/
    enviar_comando("E");
}
/* Se não houver interação com o joystick Shield, */
/* chame a função auxiliar para enviar o comando N. */
enviar_comando("N");
}
```

```
/* Função auxiliar para envio dos comandos. */
void enviar_comando(const char *msg) {
    /* Enviando o comando. */
    vw_send((uint8_t *)msg, strlen(msg));
    /* Aguarda até que o comando seja enviado. */
    vw_wait_tx();
    /* Chame a função auxiliar para checar o joystick. */
    checar_joystick();
}
/* Função auxiliar para realizar a leitura do joystick.*/
void checar_joystick() {
    JoyX = analogRead(A0);
    JoyY = analogRead(A1);
    estadoBotaoA = digitalRead(2);
    estadoBotaoB = digitalRead(3);
    estadoBotaoC = digitalRead(4);
    estadoBotaoD = digitalRead(5);
}
```



Quadro 02 - Linguagem de programação por código -
Receptor do Braço Robótico

```

/*****/
/* Aula 40 - Braço Robótico Rádio Controlado */
/* Programação do receptor para o braço robótico. */
/* O módulo irá receber os comandos enviados pelo */
/* transmissor e a partir destes, será feito o controle*/
/* dos movimentos de cada Servo Motor. */
/*****/

/* Definição da 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 braços. */
/* Valores recomendados: 2000, 2100, 2200, 2300, etc. */
#define velocidade 2000
/* Define o pino digital 9 do Arduino para enviar os */
/* comandos ao módulo transmissor RF 433MHz. */
#define Pino_Receptor 2
/* Inclui a biblioteca de controle do módulo RF433MHz. */
/* http://www.airspayce.com/mikem/arduino/VirtualWire/VirtualWire-1.27.zip */
#include <VirtualWire.h>
/* Inclui a biblioteca de controle dos servos. Não é */
/* utilizar a biblioteca Servo padrão do Arduino IDE */
/* pois ela entra em conflito com a biblioteca */
/* VirtualWire. Uma solução é utilizar a biblioteca */
/* ServoTimer2.h que pode ser baixada através do link: */
/* https://github.com/nabontra/ServoTimer2 */
#include <ServoTimer2.h>
/* Criando objetos de controle para cada Servo Motor. */
ServoTimer2 servo_base;
ServoTimer2 servo_direita;
ServoTimer2 servo_esquerda;
ServoTimer2 servo_garra;
/* Variável que definirá o Servo a ser controlado. */
char funcao_selecionada;
/* Variável de incremento da posição do Servo Motor. */
/* Poderá alterar para 1, 2, 3, etc. Quanto maior, mais*/
/* rápido o movimento. */

```

```
int incremento = 4;
/* Variável que armazenará a posição em graus do Servo.*/
int posicao;
/* Posição inicial dos quatro servos. Para a biblioteca*/
/* ServoTimer2 1400 equivale a posição de 90° */
int posicao_base = 1400;
int posicao_direita = 1400;
int posicao_esquerda = 1400;
int posicao_garra = 1400;
/* 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 decodificado. */
char recebido;
/* Função de configuração. */
void setup() {
  /* Função que informa à biblioteca qual o pino */
  /* utilizado para a recepção dos dados. */
  vw_set_rx_pin(Pino_Receptor);
  /* Função que informa à biblioteca qual a velocidade */
  /* utilizada para a recepção dos dados. */
  vw_setup(velocidade);
  /* Função que inicializa o receptor RF 433 MHz. */
  vw_rx_start();
  /* Define as portas digitais para cada Servo Motor. */
  servo_base.attach(8);
  servo_direita.attach(9);
  servo_esquerda.attach(10);
  servo_garra.attach(11);
  /* Inicia com todos os Servos na posição de 90°. */
  servo_base.write(angulo(90));
  servo_direita.write(angulo(90));
  servo_esquerda.write(angulo(90));
  servo_garra.write(angulo(90));
}
```

```
/* Função looping principal. */
void loop() {
  /* Chama a função auxiliar para receber os comandos. */
  /* e definir em que modo o braço será operado. */
  receber_comando();
  /* Chama a função auxiliar para definir o modo de */
  /* operação: */
  /* No modo B será controlado o Servo da Base. */
  /* No modo E será controlado o Servo da Esquerda. */
  /* No modo D será controlado o Servo da Direita. */
  /* No modo G será controlado o Servo da Garra. */
  escolha_modos();
  /* Enquanto o modo de operação for B, faça... */
  while (funcao_selecionada == 'B') {
    /* Se o joystick for deslocado para a direita e o */
    /* ângulo do Servo ainda é maior que 0°, faça... */
    if (recebido == 'X' && posicao_base > angulo(0)) {
      /* Mova o servo da base para a posição atual. */
      servo_base.write(posicao_base);
      /* Decremente o valor da posição da base. */
      posicao_base = posicao_base - incremento;
      /* Pequena pausa. */
      delay(5);
    }
    /* Se o joystick for deslocado para a esquerda e o */
    /* ângulo do Servo ainda é menor que 180°, faça... */
    if (recebido == 'x' && posicao_base < angulo(180)) {
      /* Mova o servo da base para a posição atual. */
      servo_base.write(posicao_base);
      /* Incremente o valor da posição da base. */
      posicao_base = posicao_base + incremento;
      /* Pequena pausa. */
      delay(5);
    }
    /* Recebe os comandos para atualizar o modo de */
    /* operação. */
    receber_comando();
    escolha_modos();
  }
}
```

```
/* Enquanto o modo de operação for D, faça... */
while (funcao_selecionada == 'D') {
  /* Se o joystick for deslocado para trás e o */
  /* ângulo do Servo ainda é maior que 50°, faça... */
  if (recebido == 'y' && posicao_direita > angulo(50)) {
    /* Mova o servo da direita para a posição atual. */
    servo_direita.write(posicao_direita);
  /* Decremente o valor da posição da direita. */
  posicao_direita = posicao_direita - incremento;
  /* Pequena pausa. */
  delay(5);
}
/* Se o joystick for deslocado para frente e o */
/* ângulo do Servo ainda é menor que 180°, faça... */
if (recebido == 'Y' && posicao_direita < angulo(180)) {
  /* Mova o servo da direita para a posição atual. */
  servo_direita.write(posicao_direita);
  /* Incremente o valor da posição da direita. */
  posicao_direita = posicao_direita + incremento;
  /* Pequena pausa. */
  delay(5);
}
/* Recebe os comandos para atualizar o modo de */
/* operação. */
receber_comando();
escolha_modos();
}
/* Enquanto o modo de operação for E, faça... */
while (funcao_selecionada == 'E') {
  /* Se o joystick for deslocado para trás e o */
  /* ângulo do Servo ainda é maior que 100°, faça... */
  if (recebido == 'y' && posicao_esquerda > angulo(100)) {
    /* Mova o servo da esquerda para a posição atual.*/
    servo_esquerda.write(posicao_esquerda);
    /* Decremente o valor da posição da esquerda. */
    posicao_esquerda = posicao_esquerda - incremento;
    /* Pequena pausa. */
    delay(5);
  }
}
```



```
/* Se o joystick for deslocado para frente e o */
/* ângulo do Servo ainda é menor que 180°, faça... */
if (recebido == 'Y' && posicao_esquerda < angulo(180)) {
    /* Mova o servo da esquerda para a posição atual.*/
    servo_esquerda.write(posicao_esquerda);
    /* Incremente o valor da posição da esquerda. */
    posicao_esquerda = posicao_esquerda + incremento;
    /* Pequena pausa. */
    delay(5);
}
/* Recebe os comandos para atualizar o modo de */
/* operação. */
receber_comando();
escolha_modos();
}
/* Enquanto o modo de operação for G, faça... */
while (funcao_selecionada == 'G') {
    /* Se o joystick for deslocado para a esquerda e o */
    /* ângulo do Servo ainda é maior que 60°, faça... */
    if (recebido == 'x' && posicao_garra > angulo(60)) {
        /* Mova o servo da garra para a posição atual. */
        servo_garra.write(posicao_garra);
        /* Decremente o valor da posição da garra. */
        posicao_garra = posicao_garra - incremento;
        /* Pequena pausa. */
        delay(5);
    }
    /* Se o joystick for deslocado para a direita e o */
    /* ângulo do Servo ainda é menor que 90°, faça... */
    if (recebido == 'X' && posicao_garra < angulo(90)) {
        /* Mova o servo da garra para a posição atual. */
        servo_garra.write(posicao_garra);
        /* Incremente o valor da posição da garra. */
        posicao_garra = posicao_garra + incremento;
        /* Pequena pausa. */
        delay(5);
    }
}
```

```
/* Recebe os comandos para atualizar o modo de */
/* operação. */
receber_comando();
escolha_modos();
}
}
/* Função auxiliar de captura dos comandos recebidos. */
void receber_comando() {
    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 que define o modo de operação. */
void escolha_modos() {
    /* Se receber B, então faça... */
    if (recebido == 'B') {
        /* Defina a função controle do Servo da Base. */
        funcao_selecionada = recebido;
    }
    /* Se receber D, então faça... */
    if (recebido == 'D') {
        /* Defina a função controle do Servo da Direita. */
        funcao_selecionada = recebido;
    }
    /* Se receber E, então faça... */
    if (recebido == 'E') {
        /* Defina a função controle do Servo da Esquerda. */
        funcao_selecionada = recebido;
    }
    /* Se receber G, então faça... */
    if (recebido == 'G') {
        /* Defina a função controle do Servo da Garra. */
        funcao_selecionada = recebido;
    }
}
}
```

```
/* Função auxiliar que converte o angulo de grau para */  
/* largura de pulsos em milissegundos. A biblioteca */  
/* utilizada (ServoTimer2) controla os Servos a */  
/* partir dessa informação ao invés de ângulos. */  
int angulo(int grau) {  
    posicao = map(grau, 0, 180, 600, 2400);  
    return posicao;  
}
```

A seguir, 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 subir o programa para o Arduino.

Após o carregamento do programa para o Arduino, o Braço Robótico poderá ser acionado pelo Joystick Shield da seguinte forma:

- Utilize os 4 botões coloridos para selecionar o Servo Motor a ser controlado;
- Utilize o Joystick movendo para os lados ou para frente e para trás, para controlar o Servo Motor selecionado.



Desafio:

Que tal utilizar outros braços robóticos e reorganizarem de tal forma que ao serem manipulados com Joystick Shield transfiram objetos uns com os outros.



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 o correto acoplamento do Joystick Shield ao Arduino;
- c. Verifique se a programação está adequada a cada porta digital.

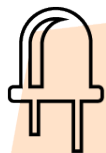
3. Feedback e Finalização:

- 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 funcionamento do monitor de sensores.
- 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

40

BRAÇO ROBÓTICO VIA RF



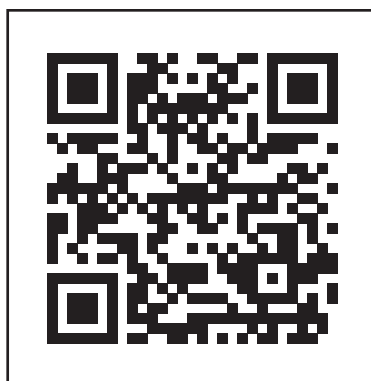
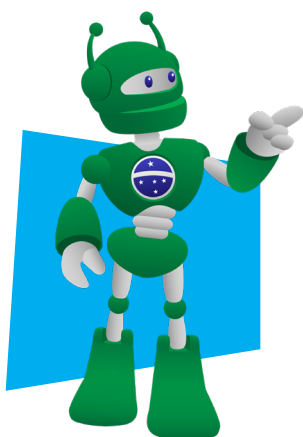
Videotutorial

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



<https://rebrand.ly/a40robotica2>

Acesse, também, pelo QRCode:





Referências

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.

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

MINARI, Gustavo. **Cientistas criam Braço Robótico capaz de achar até itens perdidos em uma bolsa**. Canaltech. Disponível em: <https://canaltech.com.br/robotica/cientistas-criam-braco-robotico-capaz-de-achar-ate-itens-perdidos-em-uma-bolsa-198106/>. Acesso em: 15 abr. 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: 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/)

