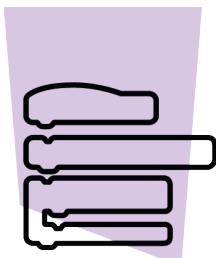
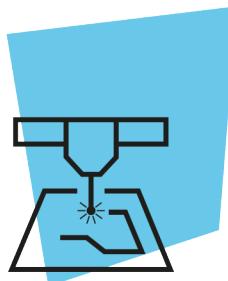
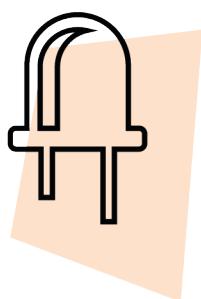
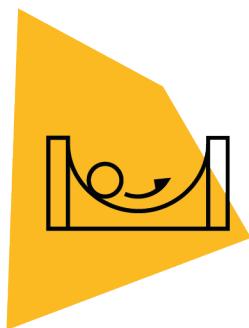
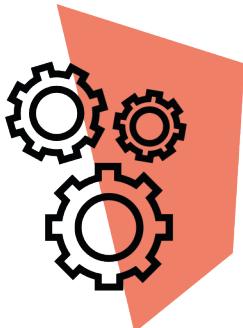


ROBÓTICA

Módulo 2



JoyStick Shield

AULA 19

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

Darice Alessandra Deckmann Zanardini

Validação de Conteúdo

Cleiton Rosa

Revisão Textual

Adilson Carlos Batista

Leitura Crítica e Normalização Bibliográfica

Ricardo Hasper

Projeto Gráfico e Diagramação

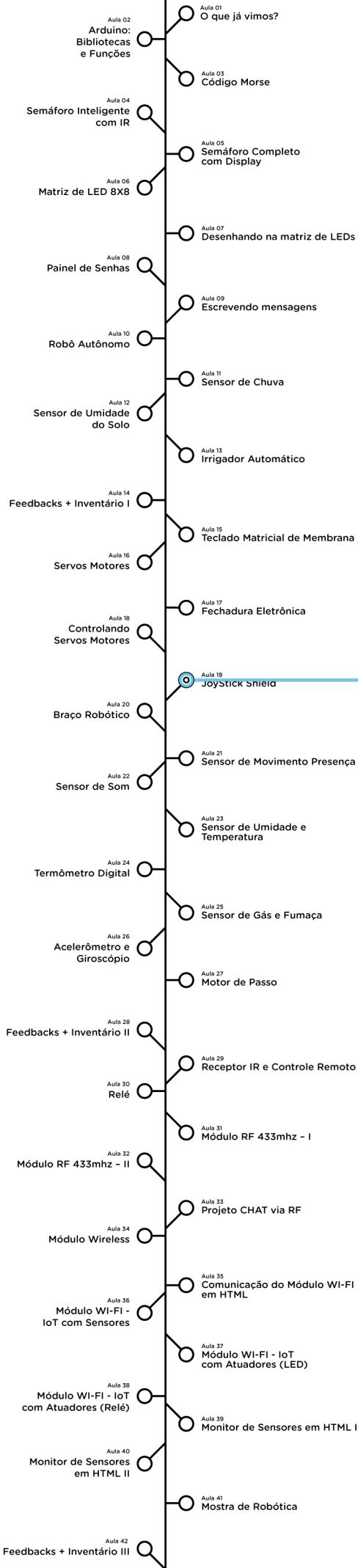
Edna do Rocio Becker

Ilustração

Jocelin Vianna (Educa Play)

2022

Aula 18 Controlando Servos Motores



Aula 19 JoyStick Shield

Aula 20 Braço Robótico [Montagem]

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	13
3. Feedback e finalização	25
Videotutorial	26

19 Joystick Shield



Introdução

O pessoal que curte jogos se anima quando vê um joystick – afinal, tantos movimentos e combinações que podem ser realizados nas sagas dos games, não é? Mas você sabia que os joysticks não são utilizados apenas em jogos ou consoles? Nesta aula, vamos saber mais sobre este componente, conhecer sua história e programar um, associado ao Arduino, para ser utilizado em projetos que precisam ser controlados.

Figura 1 – Joystick do console PS4



Fonte: WikiMedia Commons



Objetivos desta aula

- Conhecer o joystick;
- Programar o joystick shield DIY para Arduino;
- Testar as funcionalidades do joystick shield DIY para Arduino.



19 Joystick Shield



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.



19 Joystick Shield



Habilidades do século XXI a serem desenvolvidas

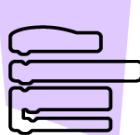
- Pensamento crítico;
- Afinidade digital;
- Resiliência;
- Resolução de problemas;
- Colaboração;
- Comunicação.



Lista de materiais

- 1 placa protoboard;
- 1 placa Arduino Uno R3;
- 1 cabo USB;
- 1 joystick shield;
- 18 jumpers;
- 2 servomotores;
- 1 buzzer;
- 6 LEDs;
- 6 resistores 220 ohms;
- 1 notebook;
- Software Arduino IDE.





Roteiro da aula

1. Contextualização

O joystick é um artefato controlador de movimentos com uma alavanca ou haste giratória, em sua base, que informa ao dispositivo que está sendo controlado os ângulos ou direção do movimento a ser feito. O joystick pode contar também com botões complementares para o controle do dispositivo associado.

Além da aplicação em jogos, o joystick é utilizado também para o controle de equipamentos variados, como drones, guindastes, câmeras e veículos diversos, como cadeiras de rodas, caminhões, veículos subaquáticos e ônibus espaciais.

Figura 2 - Astronauta Richard N. Richards na cabine do ônibus espacial Columbia durante o voo STS-28, em 1989.



Fonte: WikiMedia Commons

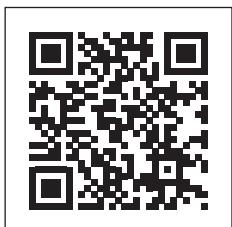
Originalmente, os joysticks foram criados por aviadores, no início do século XX, pela necessidade de controle de aeronaves/aviões e contava apenas com uma alavanca central ou lateral. Como em outras aplicações do joystick, atualmente os destinados às aeronaves contam também com botões para o controle de condições variadas do voo.

19 Joystick Shield



Para saber mais...

A utilização de joysticks no mundo dos games começou com o jogo Spacewar!, na década de 60. Antes de Spacewar!, o jogo Tennis for Two, por exemplo, era controlado por um osciloscópio adaptado, não por um joystick.



[Spacewar! by Steve Russell, 1962](#)

Confira, neste vídeo, o visual do jogo Spacewar!

Falando em games, você sabe quais são os benefícios dos games? Confira!

Figura 3: Benefícios de jogar videogame



Fonte: SEED/DTI/CTE



19 Joystick Shield

São variados os modelos de joystick, os quais sempre contam com uma haste, alavanca ou manopla fixada para controle de movimentos angulares. Além desta característica principal, o joystick pode contar também com botão (ou botões) de comando para atender às necessidades do equipamento a ser operado.

Quadro 1 – Modelos de joystick

		
Controle do console Nintendo 64	Controle de caminhão Mercedes-Benz Unimog	Controlador de câmeras Ventech PTZ
		
Controle do console Gemini	Controle do computador doméstico TheC64 Mini	Controle do console Atari 2600

Fonte: WikiMedia Commons (adaptado).

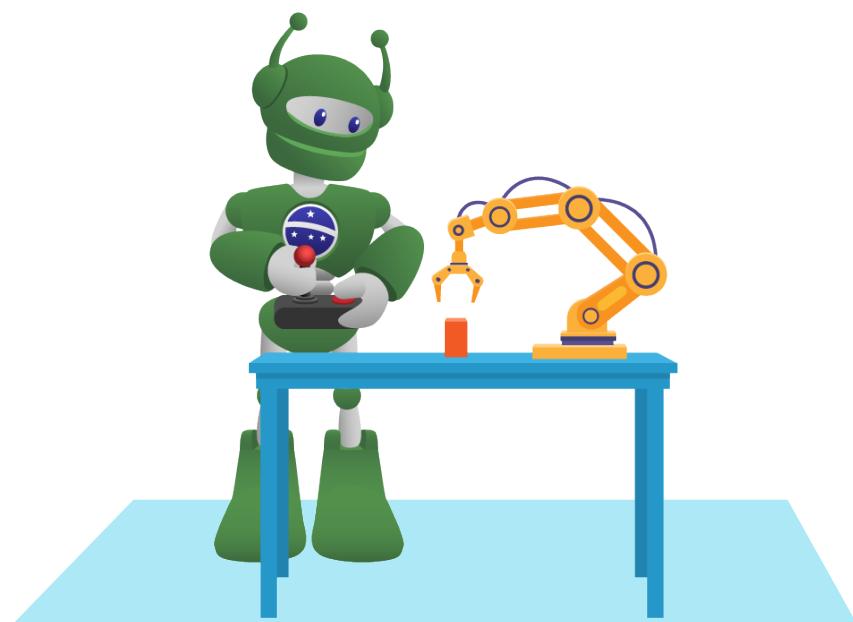


19 Joystick Shield

O joystick shield para Arduino possui, em sua haste, dois potenciômetros para os eixos X e Y, com valores em uma faixa de 0-1023, e um botão K, ativado quando a haste é pressionada para baixo. Em repouso, os potenciômetros do eixo X e do eixo Y ficam com o valor em torno de 512 (média da faixa de valor 0-1023). À medida que a haste do joystick shield for manipulada, os potenciômetros registrarão valores mais altos ou mais baixos relacionados ao movimento que estiver sendo realizado, os quais poderão ser acompanhados pelo Monitor Serial do Arduino IDE.

Ao lado da haste, há um total de seis botões para serem programados e permitir a montagem do controle com funções atribuídas a cada botão. Os botões A, B, C e D geralmente são utilizados para funções de localização e movimento (deslocar-se acima, abaixo, à esquerda ou à direita) e os botões E e F geralmente para funções de seleção ou início.

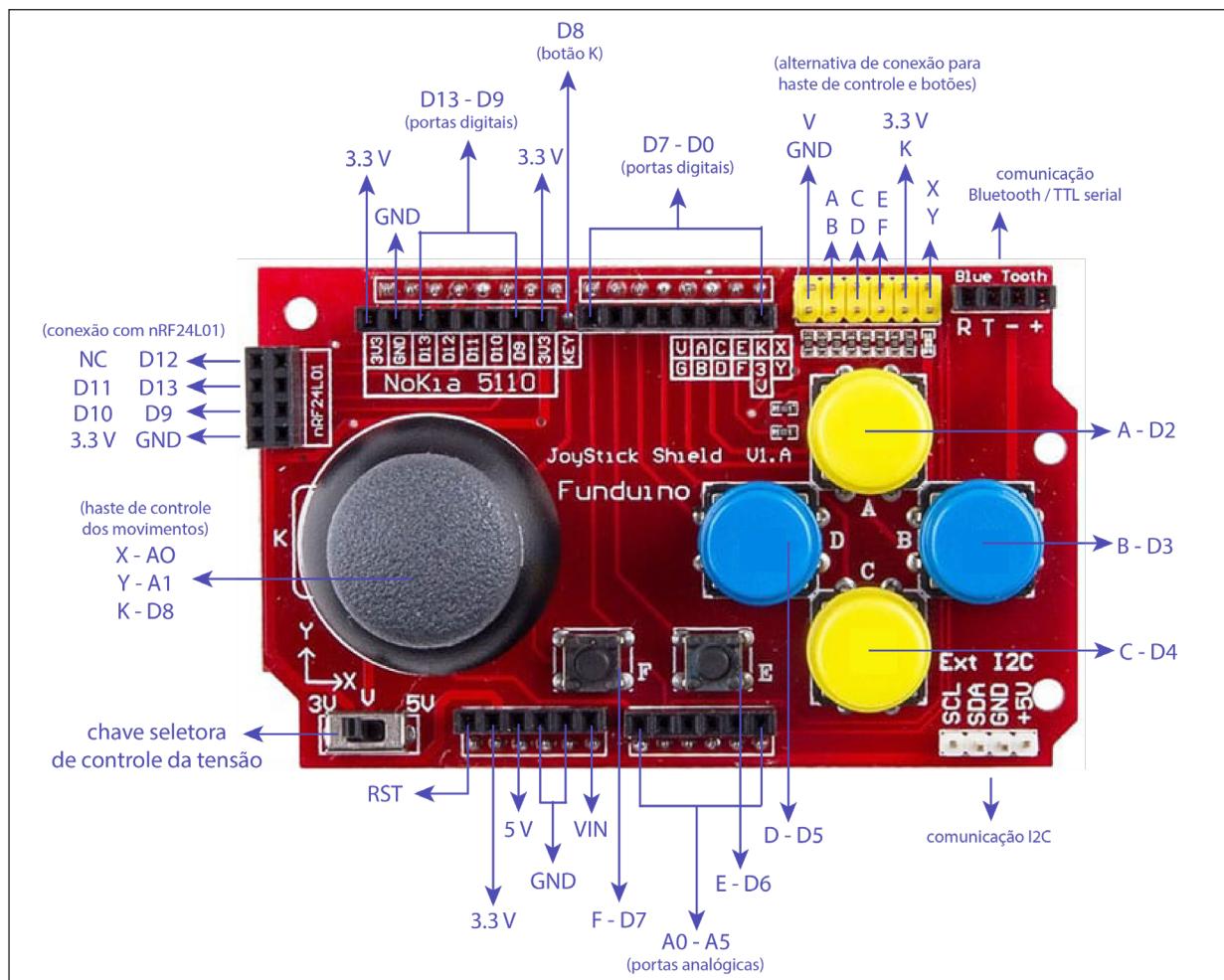
Além da haste e dos botões, o joystick shield possui chave deslizante de tensão que permite selecionar entre 3.3 V e 5 V - para evitar danos ao joystick shield, certifique-se de configurá-lo para a tensão correta da placa de prototipagem (Arduino) que você estiver usando.



AULA

19 Joystick Shield

Figura 4 - Joystick shield e suas conexões com o Arduino



Fonte: SEED/DTI/SEED

Joystick Shield DIY para Arduino - Especificações Técnicas

Dimensões (CxLxA): ~ 87 x 53 x 30mm

Peso: 34g

Joystick eixos: Eixo X A0, Eixo Y A1, Eixo Z

Botões coloridos: A D2, B D3, C D4, D D5, E D6, F D7, K D8.

Interface analógica: A2 - A5

Interface digital: D9 - D13

Acesso ao botão de reset do Arduino

Expansão para interface I2C e TTL

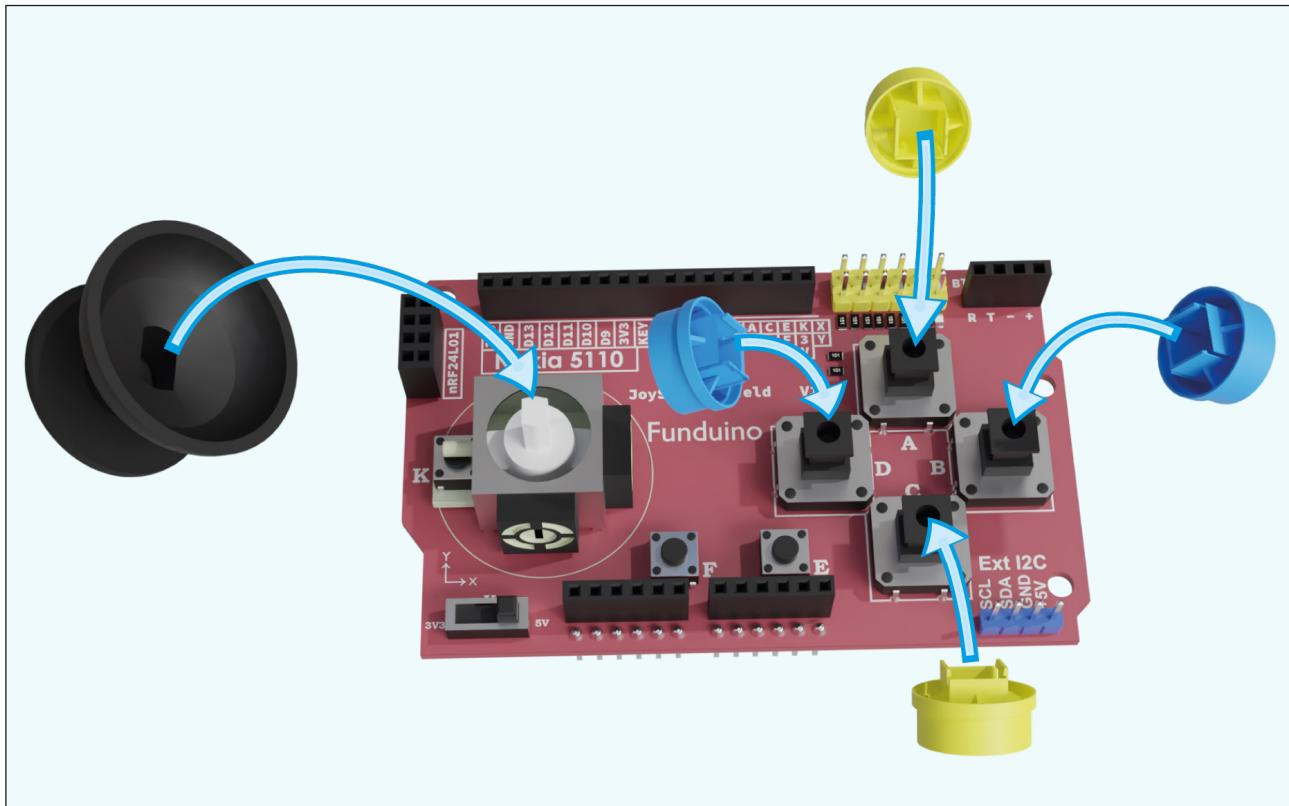
Interruptor de alimentação: 3.3 V e 5 V

19 Joystick Shield

O joystick shield também possui também pinos que possibilitam a conexão de módulo bluetooth ou dispositivo serial TTL, dispositivo I2C, módulo wireless nRF24L01 ou mesmo um display nokia 5110 (itens que NÃO acompanham nosso kit de robótica).

Para usar o joystick shield é preciso, na primeira vez que for utilizá-lo, fazer o encaixe dos botões (caso os mesmos não estejam encaixados).

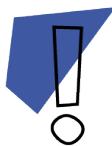
Figura 5 – Encaixe de botões no joystick shield



Fonte: SEED/DTI/CTE

Após, basta conectá-lo ao Arduino, realizar as demais conexões físicas e programar!

19 Joystick Shield



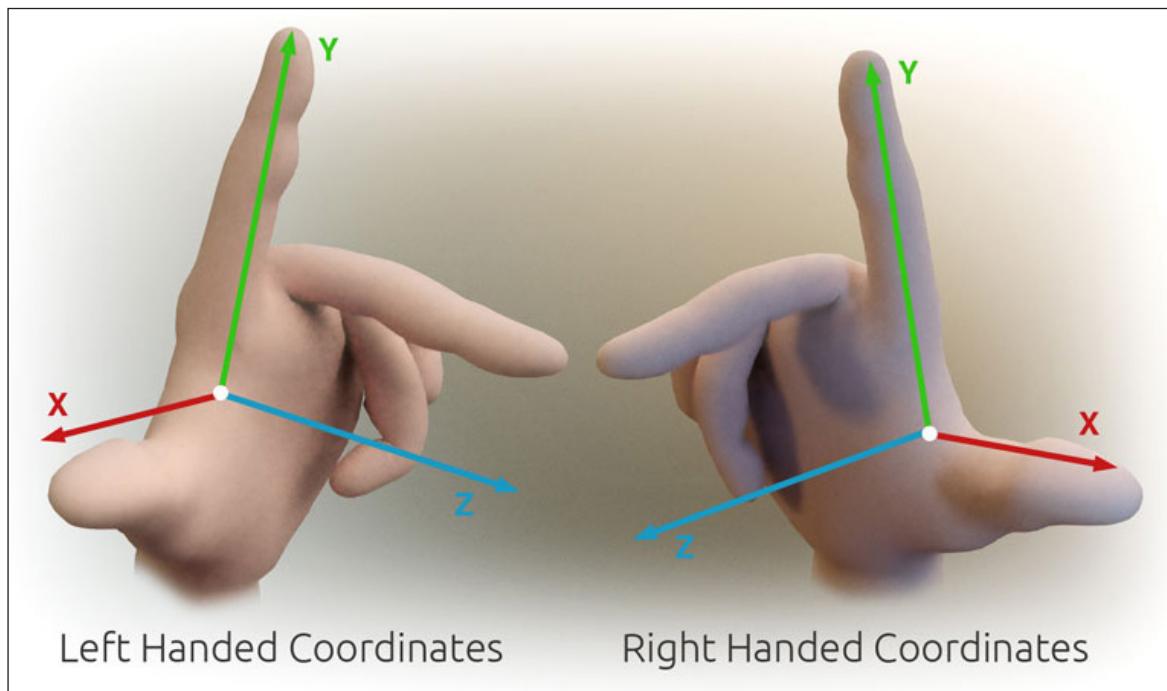
Para saber mais...

Eixos X, Y e Z

Em um sistema tridimensional de coordenadas cartesianas, os eixos perpendiculares X, Y e Z representam a abscissa (X), a ordenada (Y) e a cota (Z).

As coordenadas determinam a posição de um ponto em um plano ou no espaço. Em um plano, a posição de um ponto é geralmente determinada pelas distâncias a duas linhas retas (eixos coordenados) que se cruzam em um ponto (origem) em ângulos retos; uma das coordenadas é chamada de ordenada e a outra é chamada de abscissa. Quando temos a projeção espacial do terceiro eixo, a coordenada é chamada de cota.

Figura 6 - Sistema tridimensional das coordenadas cartesianas representado pelas mãos esquerda e direita



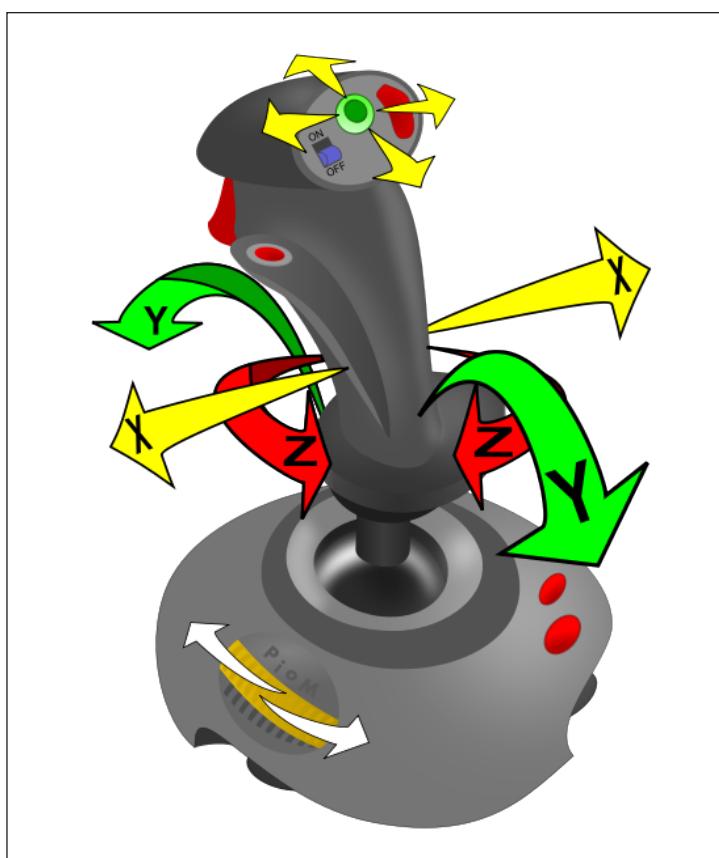
Fonte: WikiMedia Commons



19 Joystick Shield

As configurações dos joysticks, em geral, envolvem os movimentos da esquerda para direita e de cima para baixo, podendo haver configurações que proporcionem variações entre estes dois sentidos, operando em três dimensões.

Figura 7 - Direções dos movimentos do joystick



Fonte: Piotr Michał Jaworski; POLAND/Poznań, 2007. WikiMedia Commons.

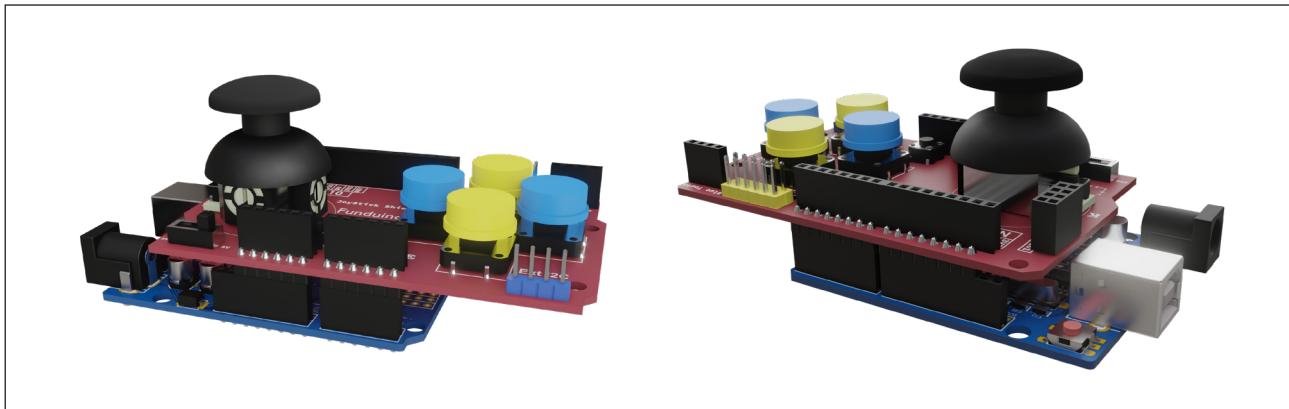
Vamos para a montagem e programação do nosso projeto? Com ele, você experimentará as funções de cada botão e haste, os quais poderão ser configurados em projetos futuros para outras aplicações em Robótica.

19 Joystick Shield

2. Montagem e programação

Para iniciarmos, é preciso primeiro acoplar o joystick shield, com os botões já encaixados, sobre o Arduino Uno R3 (figura 8).

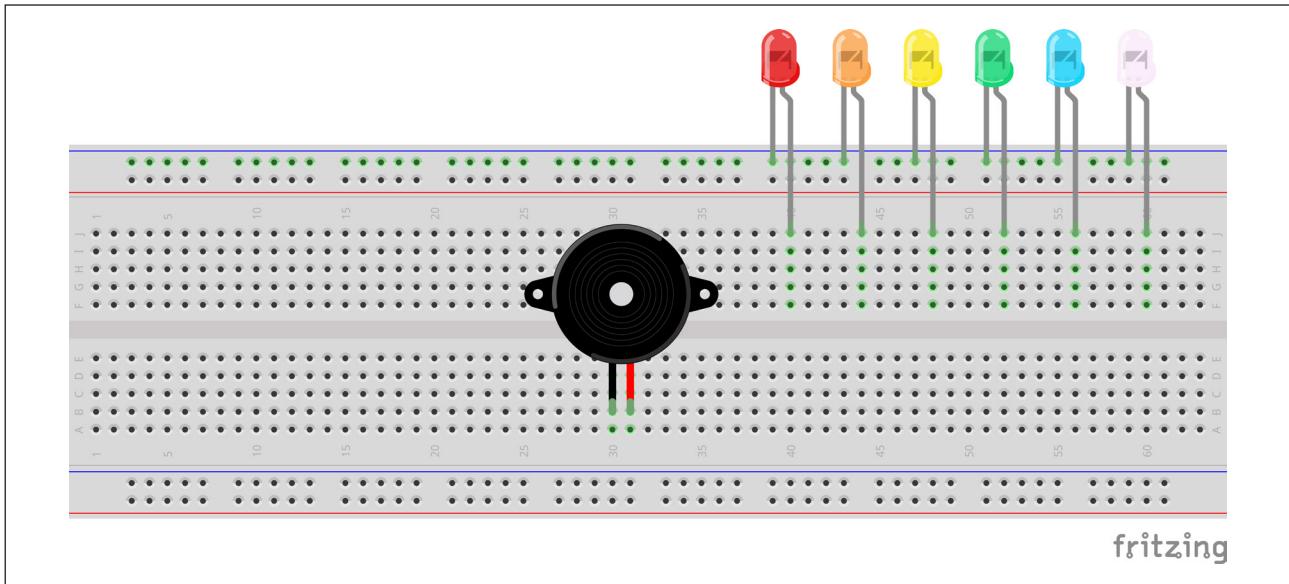
Figura 8 – Acoplando o joystick shield à placa Arduino



Fonte: Fritzing (adaptado)

Agora insira, sobre a placa protoboard, o buzzer passivo e seis LEDs (figura 9).

Figura 9 - Conexão de buzzer e LEDs à protoboard

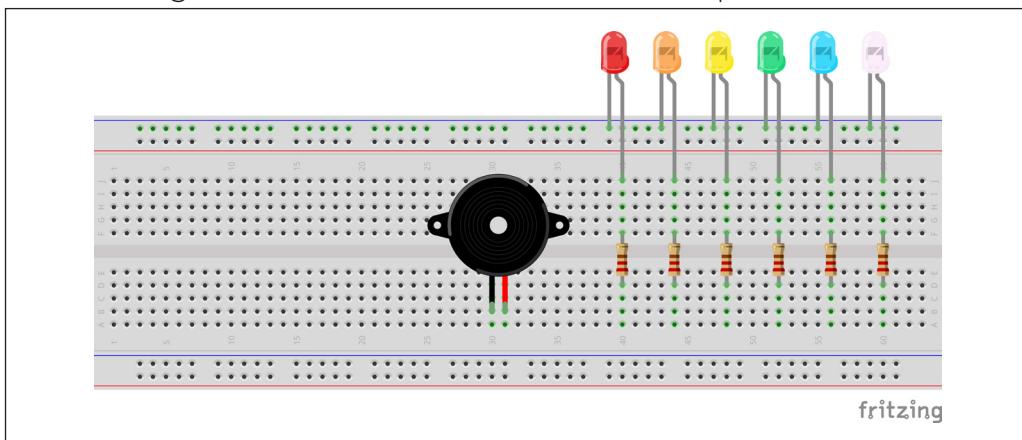


Fonte: Fritzing

19 Joystick Shield

Em seguida, insira os resistores na protoboard, atentando para que estejam conectados à mesma coluna na qual se encontram os pinos positivos dos LEDs (figura10).

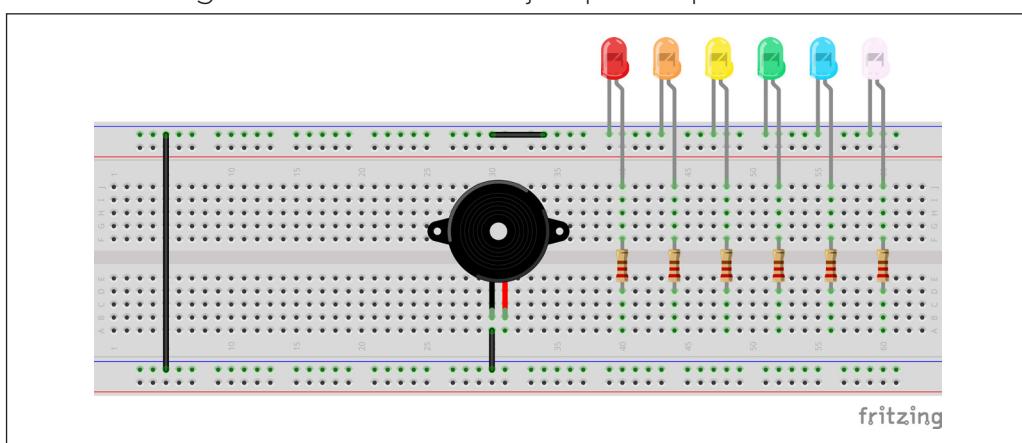
Figura 10 - Conexão dos resistores à protoboard



Fonte: Fritzing

Conekte, com três jumpers, as áreas da protoboard: com um jumper, interligue a linha de furos superior da protoboard à linha de furos inferior; com um jumper, o lado direito da linha de furos superior da protoboard, à qual estão ligados os polos negativos dos LEDs, com o lado esquerdo da linha de furos superior da protoboard. Por fim, insira um jumper na coluna à qual está conectado o polo negativo do buzzer para ligá-lo à linha inferior da protoboard (figura 11).

Figura 11 - Conexão de jumpers à protoboard

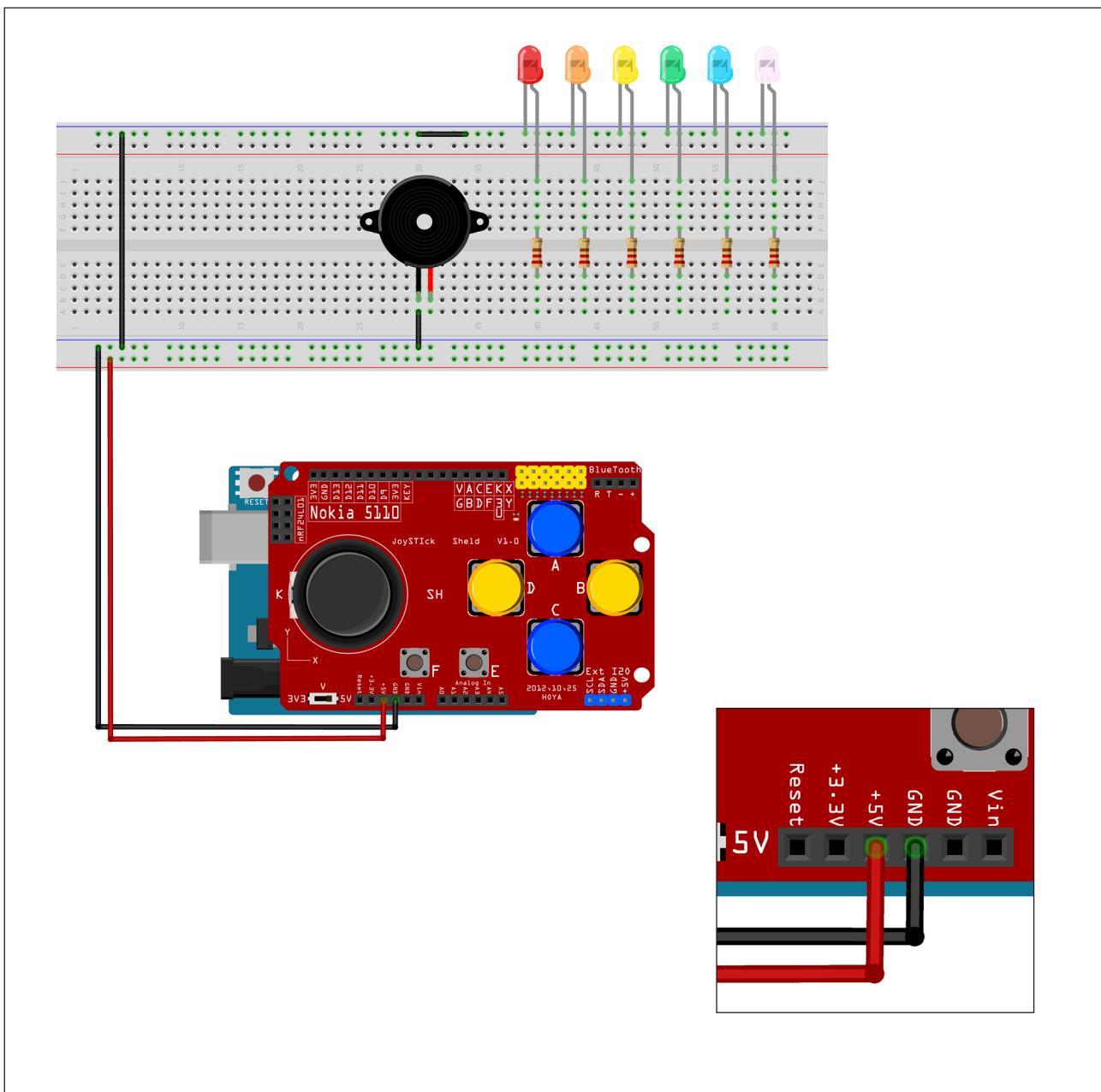


Fonte: Fritzing

19 Joystick Shield

Conecte um jumper entre a linha inferior azul da protoboard e a porta GND do joystick shield acoplado sobre o Arduino Uno. Em seguida, conecte um jumper entre a linha inferior vermelha da protoboard e a porta 5 V do joystick shield (figura 12).

Figura 12 - Conexão da protoboard às portas 5 V e GND do joystick shield

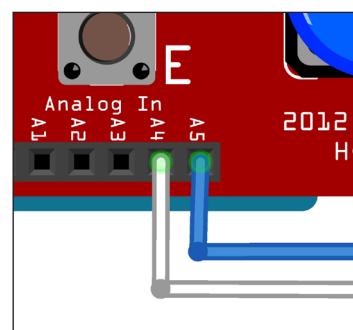
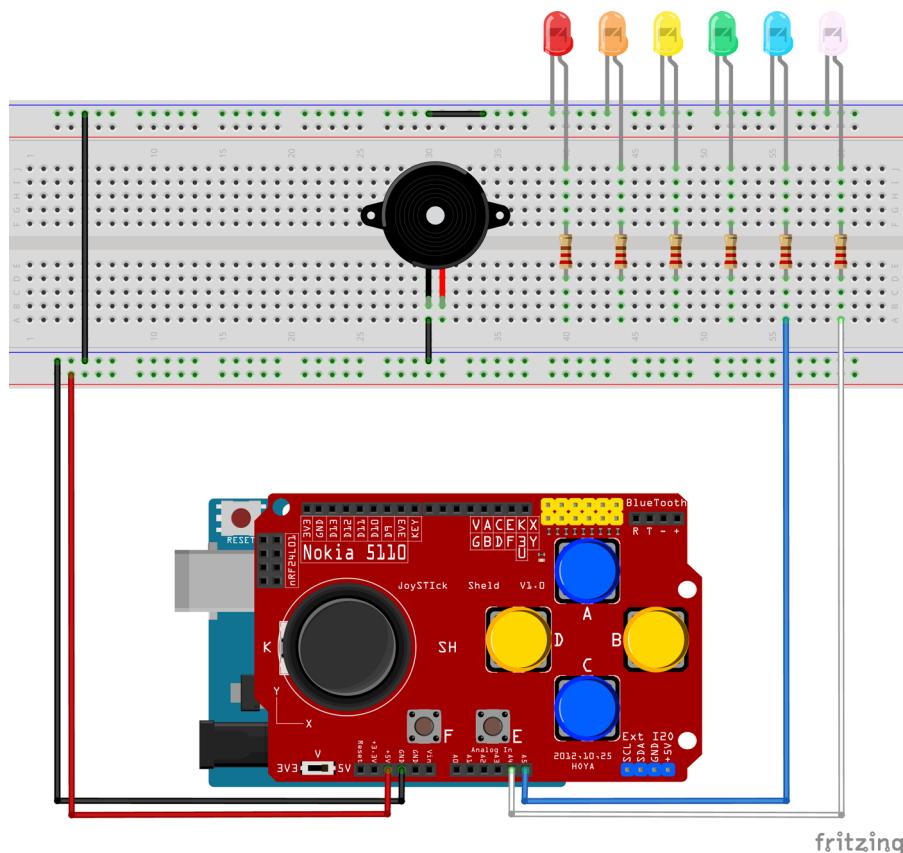


Fonte: Fritzing

19 Joystick Shield

Agora, utilizando dois jumpers, conecte dois LEDs às portas analógicas A4 e A5 do joystick shield, respectivamente (figura 13).

Figura 13 - Conexão dos LEDs às portas analógicas do joystick shield

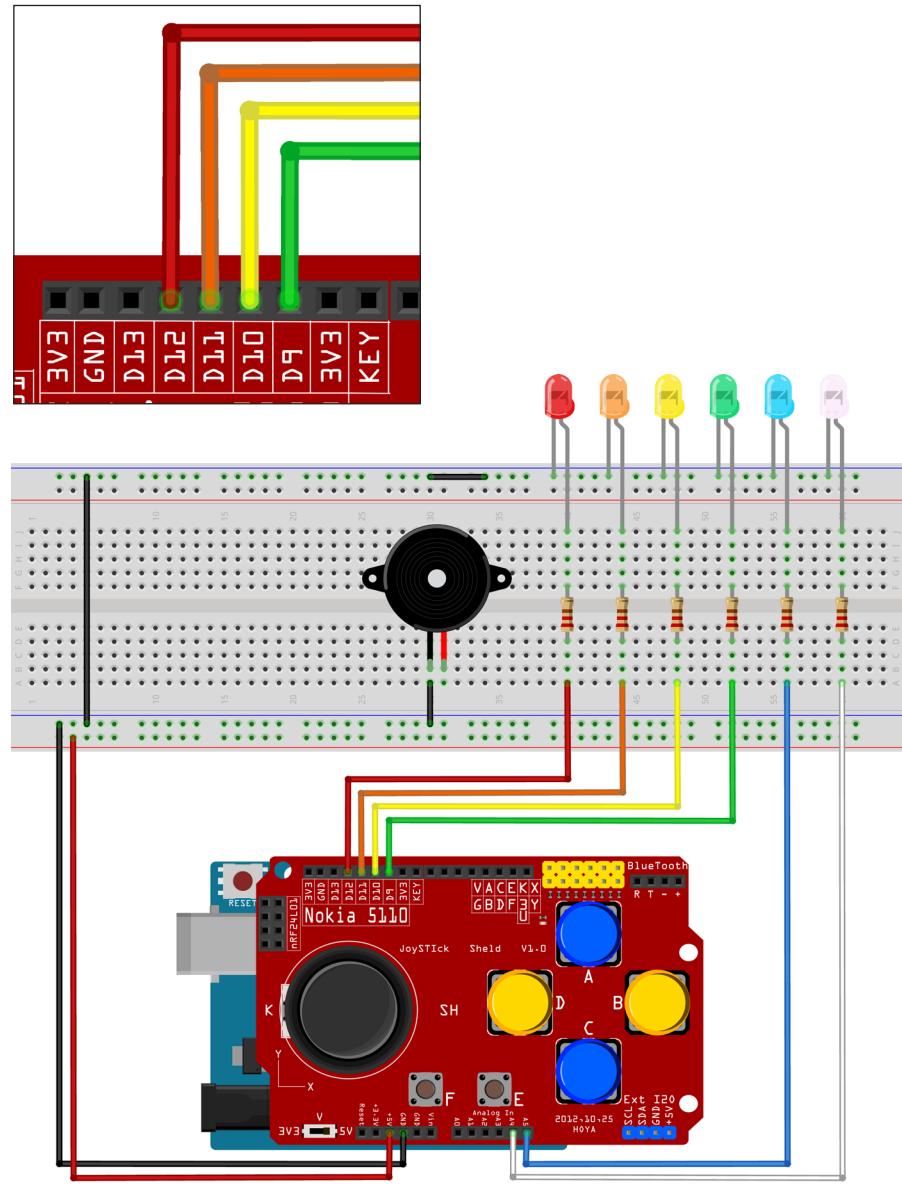


Fonte: Fritzing

19 Joystick Shield

Utilizando outros quatro jumpers, conecte os demais LEDs ao joystick shield, respectivamente às portas digitais 9, 10, 11 e 12 (figura 14).

Figura 14 - Conexão dos LEDs às portas digitais do joystick shield

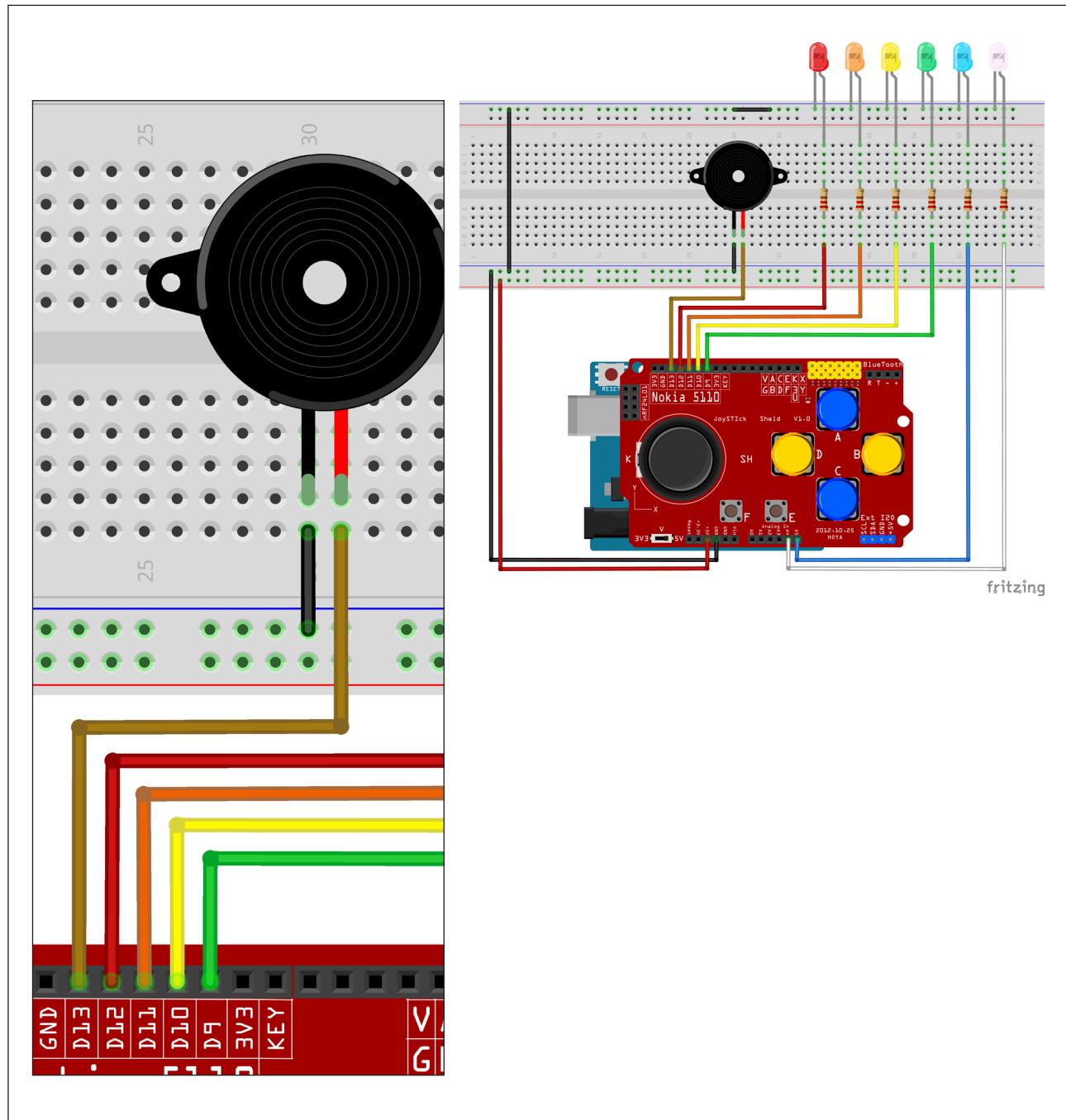


Fonte: Fritzing

19 Joystick Shield

Com 1 Jumper, conecte o terminal positivo do buzzer à porta digital 13 do joystick shield (figura 15).

Figura 15 - Conexão do buzzer ao joystick shield

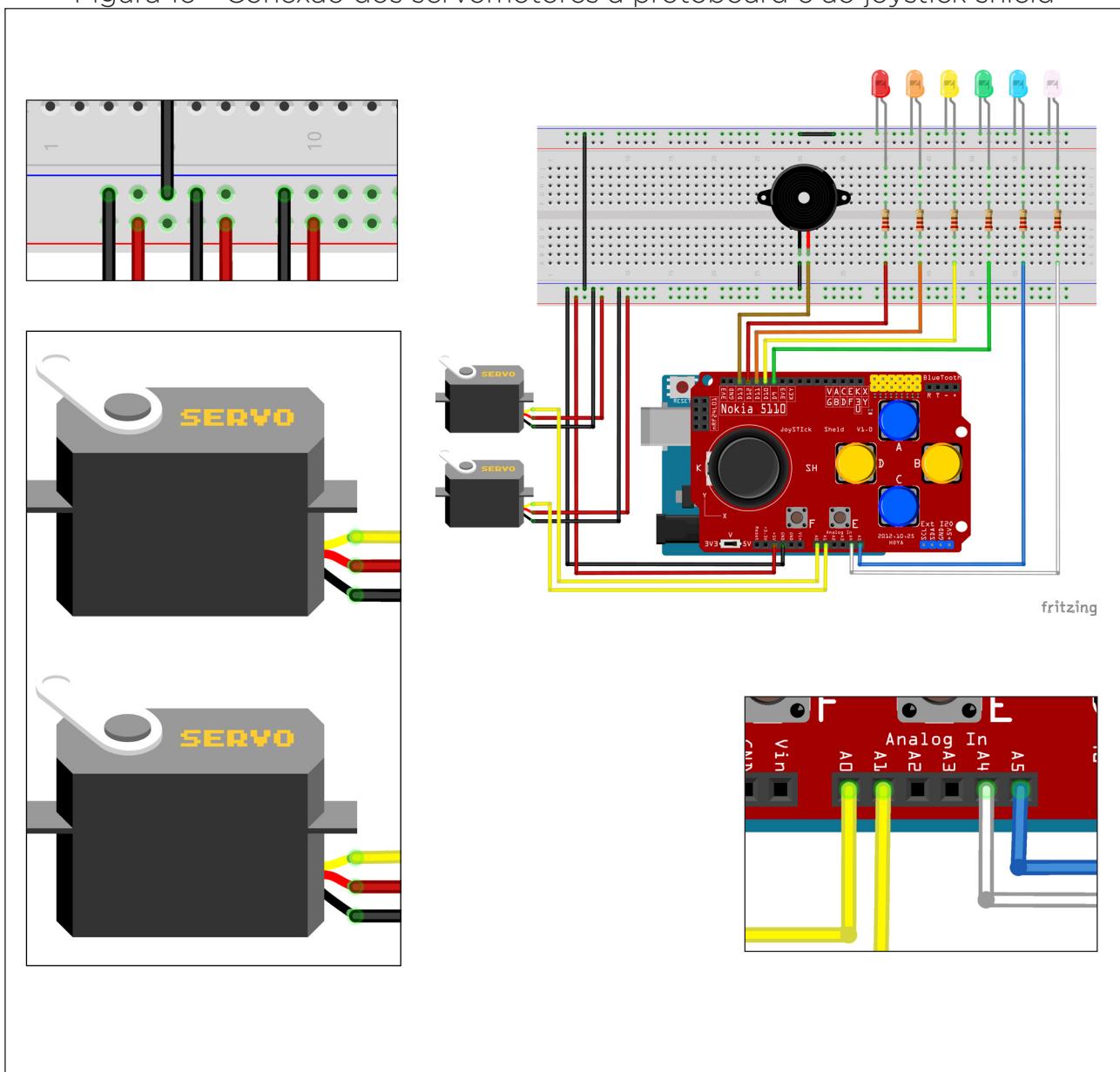


Fonte: Fritzing

19 Joystick Shield

Por fim, conecte os servomotores: com seis jumpers, interligue os terminais marrom e vermelho de cada servomotor aos furos das linhas inferiores azul e vermelha da protoboard, respectivamente. Depois, os terminais amarelos de cada servomotor aos pinos A2 e A3 do joystick shield (figura 16).

Figura 16 - Conexão dos servomotores à protoboard e ao joystick shield



Fonte: Fritzing

19 Joystick Shield



Agora, vamos programar!

Com o joystick acoplado ao Arduino Uno R3 e os demais componentes montados, vamos programar!

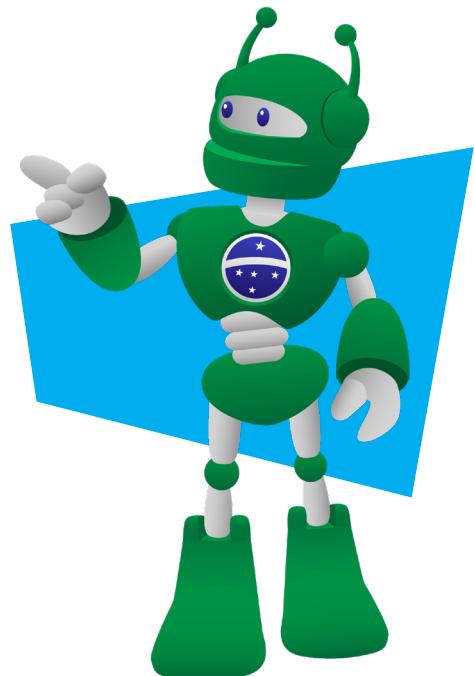
i. Linguagem de programação por código

Para iniciar a programação, conecte a placa Arduino Uno R3 ao computador, através do cabo USB, para que ocorra a comunicação entre a placa microcontroladora e o software Arduino IDE.

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

Atenção!

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



19 Joystick Shield

Quadro 01 - Programação por codificação do joystick shield

```
/* Programa: Utilizando Joystick Shield */

/* Inclui a biblioteca de controle do servo */
#include <Servo.h>
/* Cria os objetos para controlar os servos */
Servo servo1;
Servo servo2;

/* Define os pinos de dados dos servos */
int Servo1 = A2;
int Servo2 = A3;
/* Define os pinos dos botões */
int Botao_A = 2;
int Botao_B = 3;
int Botao_C = 4;
int Botao_D = 5;
int Botao_E = 6;
int Botao_F = 7;
int Botao_K = 8;
/* Define os pinos dos LEDs C ao F */
int LED_C = 9;
int LED_D = 10;
int LED_E = 11;
int LED_F = 12;
/* Define o pino do buzzer */
int Buzzer = 13;
/* Define os pinos dos potenciômetros */
int Pot_X = A0;
int Pot_Y = A1;
/* Define os pinos dos LEDs A e B. Usaremos essas portas analógicas para
simular portas digitais devido à limitação de pinos deste projeto. */
int LED_A = A4;
int LED_B = A5;
```



19 Joystick Shield

```
void setup() {  
    /* Endereça os objetos servos aos pinos de dados definidos */  
    servol.attach(Servo1);  
    servo2.attach(Servo2);  
    /* Loop para definir os pinos 2 ao 8 como entradas e ativa os resistores  
    internos do Arduino */  
    for (int i = 2; i <= 8; i++) {  
        pinMode(i, INPUT_PULLUP);  
    }  
    /* Loop para definir os pinos 9 ao 13 como saídas */  
    for (int i = 9; i <= 13; i++) {  
        pinMode(i, OUTPUT);  
    }  
    /* Define os pinos analógicos do LED_A e LED_B como saída digital */  
    pinMode(LED_A, OUTPUT);  
    pinMode(LED_B, OUTPUT);  
}  
  
void loop() {  
    /* Comandos para posicionar os servos de acordo com as leituras dos dados  
    dos potenciômetros */  
    servol.write(map(analogRead(Pot_X), 0, 1023, 0, 180));  
    servo2.write(map(analogRead(Pot_Y), 0, 1023, 0, 180));  
  
    /* Enquanto o Botão_A estiver pressionado, mantenha o LED_A ligado. */  
    while (!digitalRead(Botao_A)) {  
        digitalWrite(LED_A, HIGH);  
    }  
    /* Enquanto o Botão_B estiver pressionado, mantenha o LED_B ligado. */  
    while (!digitalRead(Botao_B)) {  
        digitalWrite(LED_B, HIGH);  
    }  
    /* Enquanto o Botão_C estiver pressionado, mantenha o LED_C ligado. */  
    while (!digitalRead(Botao_C)) {  
        digitalWrite(LED_C, HIGH);  
    }  
}
```



19 Joystick Shield

```
/* Enquanto o Botão_D estiver pressionado, mantenha o LED_D ligado. */
while (!digitalRead(Botao_D)) {
    digitalWrite(LED_D, HIGH);
}

/* Enquanto o Botão_E estiver pressionado, mantenha o LED_E ligado. */
while (!digitalRead(Botao_E)) {
    digitalWrite(LED_E, HIGH);
}

/* Enquanto o Botão_F estiver pressionado, mantenha o LED_F ligado. */
while (!digitalRead(Botao_F)) {
    digitalWrite(LED_F, HIGH);
}

/* Enquanto o Botão_K estiver pressionado, emita som através do Buzzer. */
while (!digitalRead(Botao_K)) {
    tone(Buzzer, 800);
}

/* Se nenhum botão estiver sendo pressionado, chame a função para desligar
todos os LEDs e desligue o Buzzer. */
Off_LEDs();
noTone(Buzzer);
}

/* Função responsável por desligar todos os LEDs. */
void Off_LEDs() {
    for (int i = 9; i <= 12; i++) {
        digitalWrite(i, LOW);
    }
    digitalWrite(LED_A, LOW);
    digitalWrite(LED_B, LOW);
}
```



19 Joystick Shield

Com o código-fonte inserido no Arduino IDE, compile o programa pressionando o botão **Verificar** para verificar se não há erros de sintaxe. Estando o código-fonte correto, pressione o botão **Carregar** para realizar a transferência do programa para o Arduino.

Após a transferência do programa, você poderá controlar os atuadores através dos botões e haste do joystick. Os botões acionarão os LEDs e a haste do joystick controlará os movimentos dos servomotores e buzzer.



Desafio

Que tal inserir uma matriz de LEDs 8x8 ao joystick shield? Crie setas para indicar a posição do joystick: cima, baixo, esquerda, direita e botões A, B, C, D, E, F e K.



E se...?

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

- a. Verifique se o joystick shield está posicionado corretamente sobre o Arduino Uno R3;
- b. Verifique se os componentes estão ligados nas linhas e colunas corretas da protoboard;
- c. Verifique se os jumpers estão ligados nos pinos corretos do joystick shield;
- d. Verifique se a programação está adequada a cada porta digital e analógica.

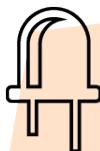
19 Joystick Shield

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 joystick shield;
- 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.



19 Joystick Shield



Videotutorial

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



<https://rebrand.ly/a19robotica2>

Acesse, também, pelo QRCode:



DIRETORIA DE TECNOLOGIAS E INOVAÇÃO (DTI)

COORDENAÇÃO DE TECNOLOGIAS EDUCACIONAIS (CTE)

EQUIPE ROBÓTICA PARANÁ

Andrea da Silva Castagini Padilha
Cleiton Rosa

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

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



Este trabalho está licenciado com uma Licença
Creative Commons – CC BY-NC-SA
[Atribuição - NãoComercial - Compartilhualgual 4.0](#)

