



Blog da Robótica





Como utilizar o teclado matricial 4×4 keypad com Arduino

📅 30 de junho de 2022 👤 Carol Correia Viana



projetos com microcontroladores, como o Arduino.

Este módulo possui 16 botões dispostos em 4 linhas e 4 colunas e um conector de 8 pinos para ligação, conforme pode ser observado na Figura 1.





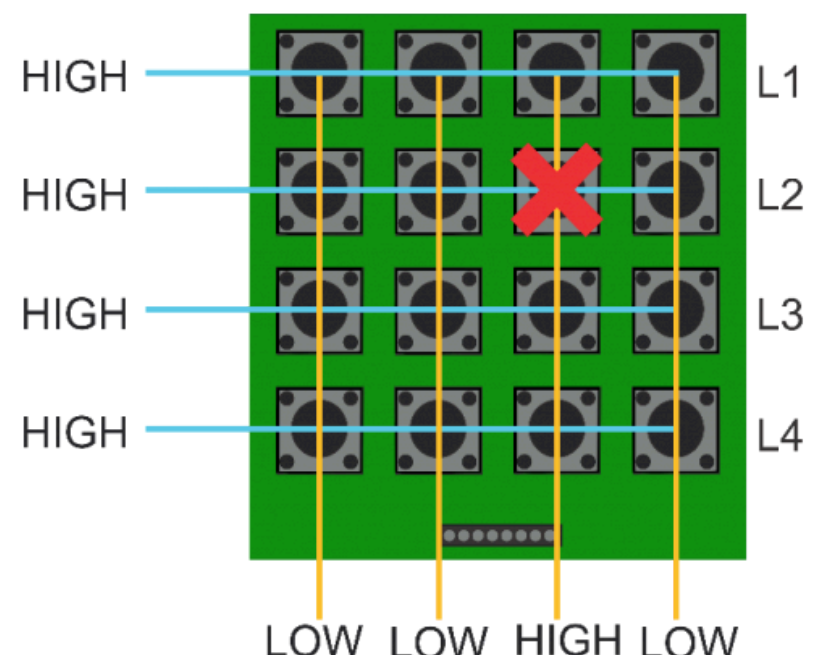
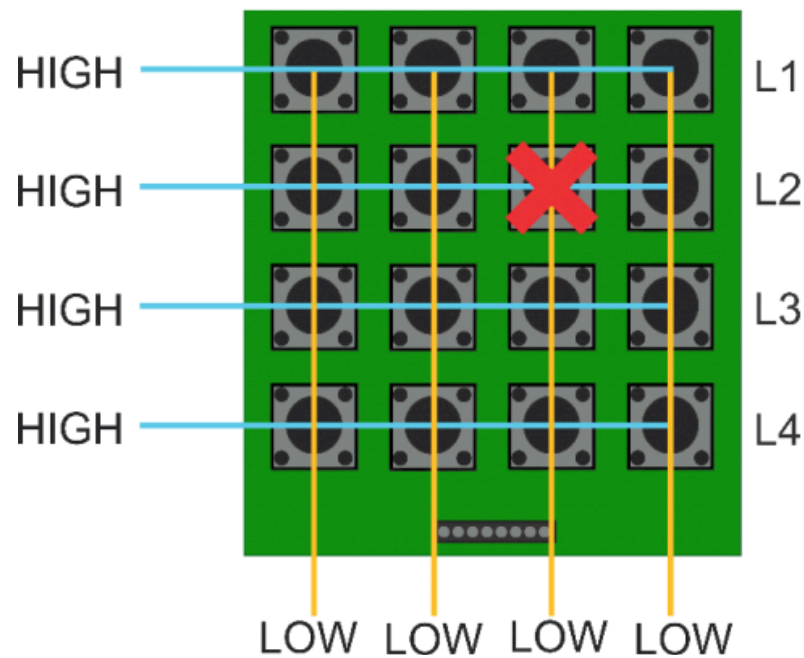
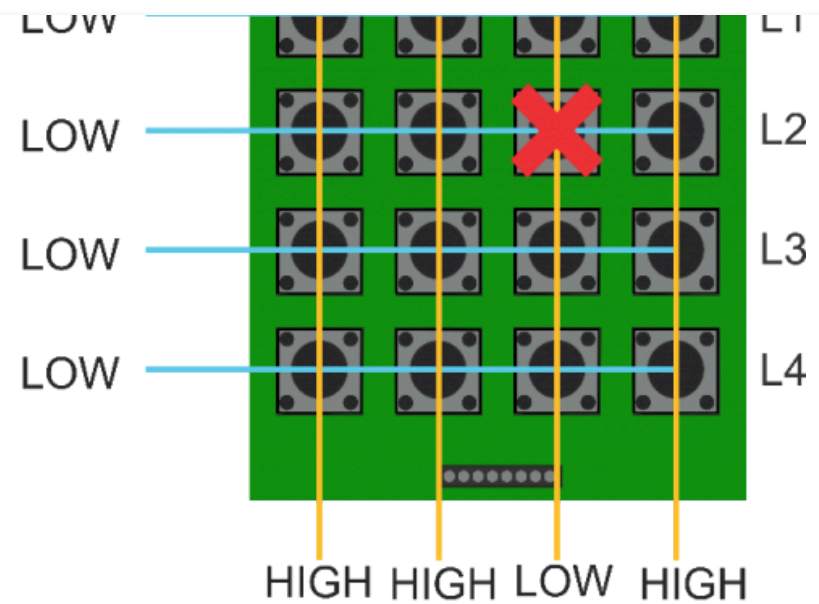
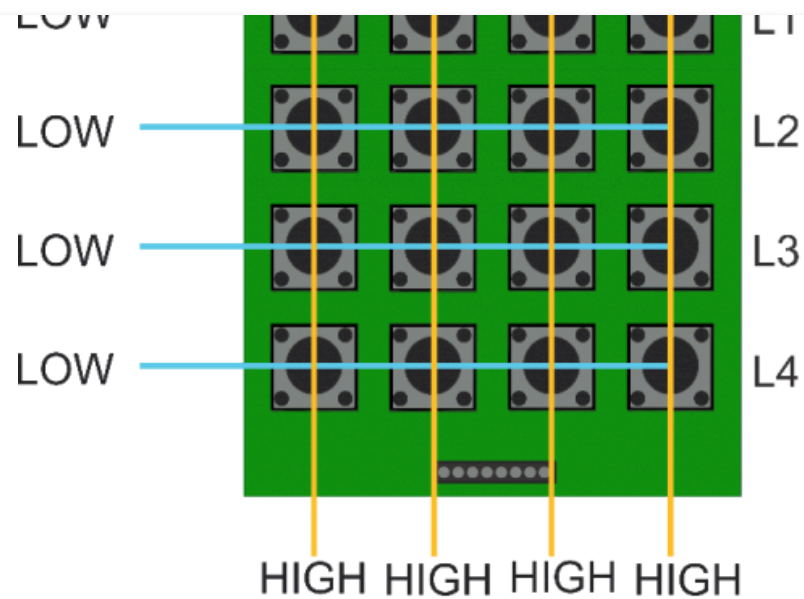
Figura 1 – Teclado matricial 4 x 4 Keypad.

No teclado matricial 4 x 4 keypad, cada botão em uma linha é conectado aos outros botões da mesma linha e coluna, formando uma matriz. Para identificar qual botão foi pressionado, o microcontrolador configura todas as colunas como nível ALTO e todas as linhas como nível BAIXO. Quando um botão é pressionado, o pino da coluna é colocado em nível BAIXO e o microcontrolador consegue identificar a coluna em que o botão pressionado está.

para identificar qual pinó da coluna recebeu a energia.

Na Figura 2 pode ser observada a combinação da linha 2 e a coluna 3, significando que o botão S7 foi pressionado.





O teclado matricial 4 x 4 keypad é ideal para conjuntos eletrônicos que necessitam de codificações ou para colocação de senhas em sistemas eletrônicos. Ele pode ser aplicado nos mais diversos projetos, apresentando excelente acabamento e perfurações para fixação nas extremidades.

Neste projeto, vamos realizar a leitura dos botões do teclado matricial 4 x 4 keypad e exibir o qual botão foi pressionado no monitor serial.

MATERIAIS NECESÁRIOS

- 1 x [Placa UNO SMD R3 Atmega328 compatível com Arduino UNO](#);
- 1 x [Cabo USB](#);
- 1 x [Protoboard](#);
- 1 x [Teclado matricial 4 x 4 keypad](#);
- [Fios de jumper macho-fêmea](#).

ESQUEMÁTICO DE LIGAÇÃO DOS COMPONENTES

Conecte os componentes na protoboard como ilustra a Figura 3. Lembre-se que a placa UNO deve estar desconectada enquanto você monta o circuito.



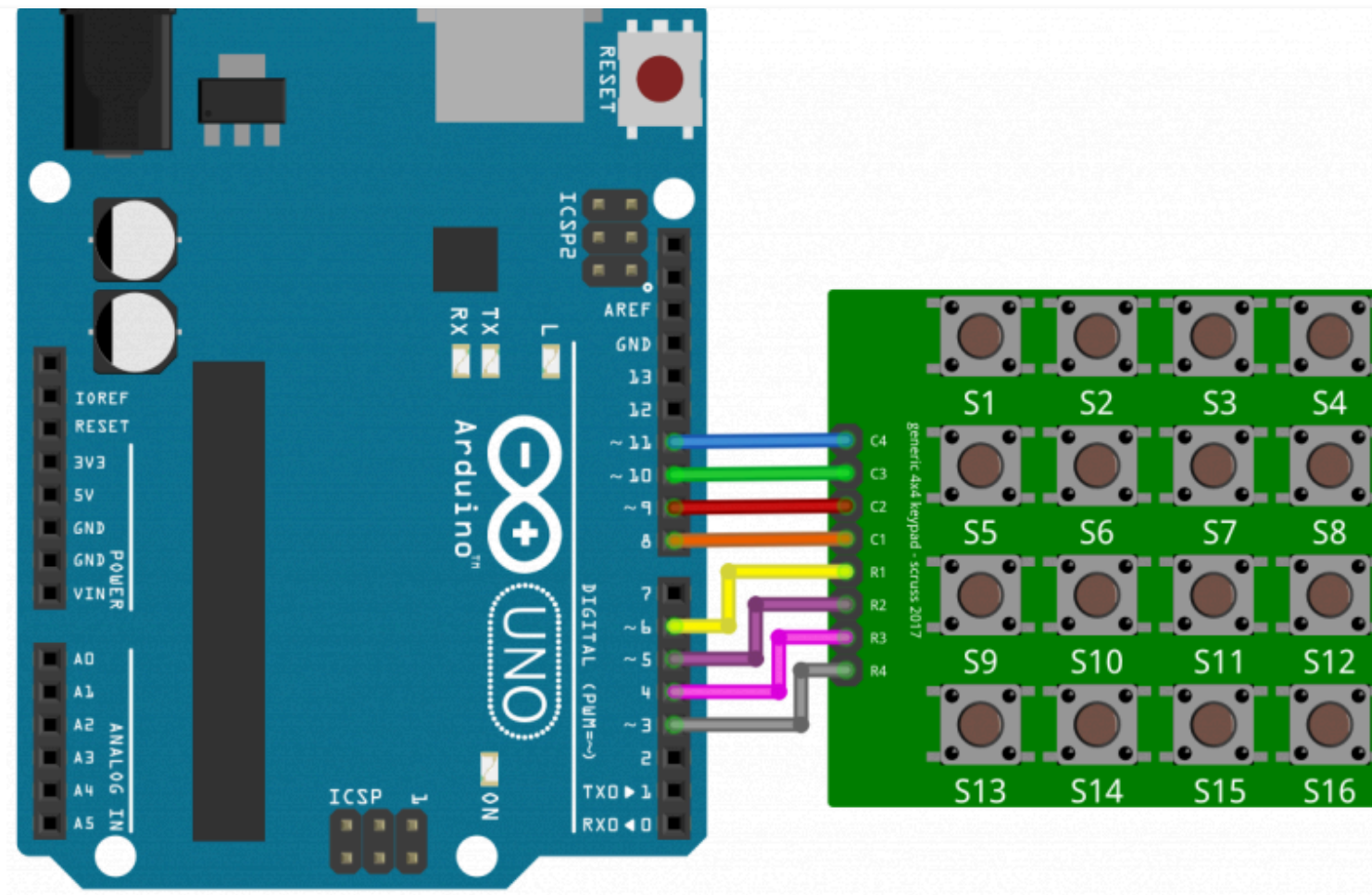


Figura 3 – Circuito para Projeto exibir botão pressionado no monitor serial

Ao montar seu circuito na protoboard observe os seguintes pontos:

- Os pinos correspondentes as colunas do teclado matricial são representados por C1, C2, C3 e C4. Estes pinos devem ser conectados, respectivamente, aos pinos 8, 9, 10 e 11 da placa UNO;

conectados as portas 0, 1, 2 e 3 da placa Uno.

ELABORANDO O CÓDIGO

Com o circuito montado, vamos a programação. A proposta deste projeto é realizar a leitura dos botões do teclado matricial 4 x 4 keypad e exibir qual botão foi pressionado no monitor serial. Para entender a lógica de programação acompanhe os seguintes passos:

1. Adicionar a biblioteca **Keypad.h**:

Para o desenvolvimento do nosso projeto, vamos incluir a biblioteca **Keypad.h**, desenvolvida para facilitar o uso de teclados do tipo matriz com o Arduino. Esta biblioteca encontra-se disponível no gerenciador de bibliotecas do Arduino IDE, que pode ser acessado por meio do caminho: Toolbar > Ferramentas > Gerenciar Bibliotecas...>

Adicionar biblioteca ZIP, conforme ilustra a Figura 4.



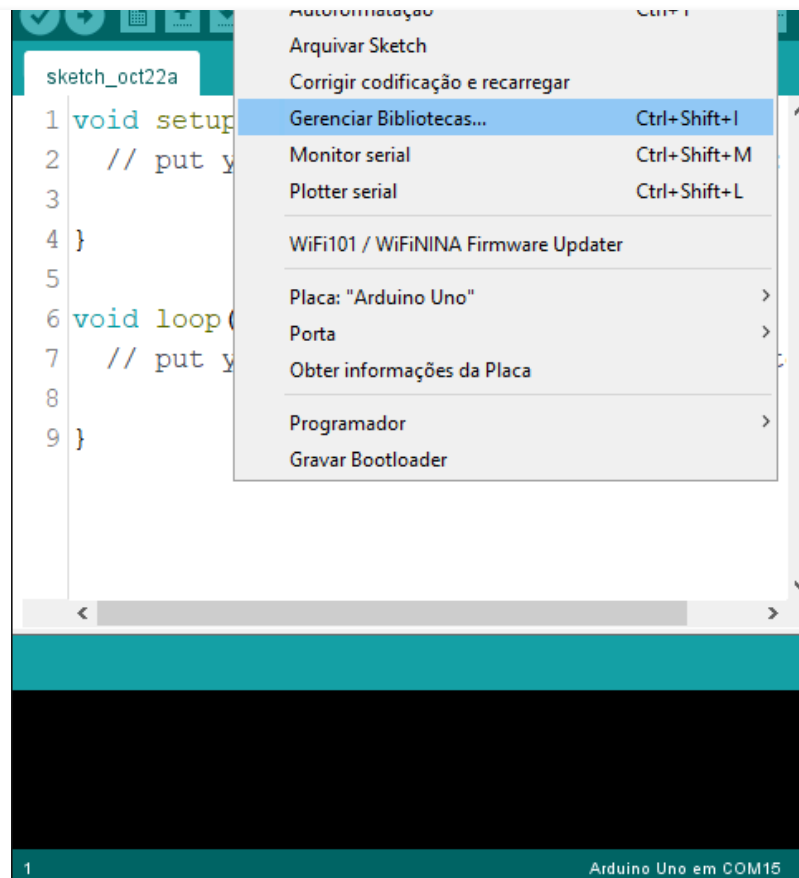


Figura 4 – Caminho para o gerenciador de bibliotecas do Arduino IDE.

Em seguida, no campo de pesquisa busque por keypad e instale a biblioteca clicando em “Instalar”, conforme Figura 5.

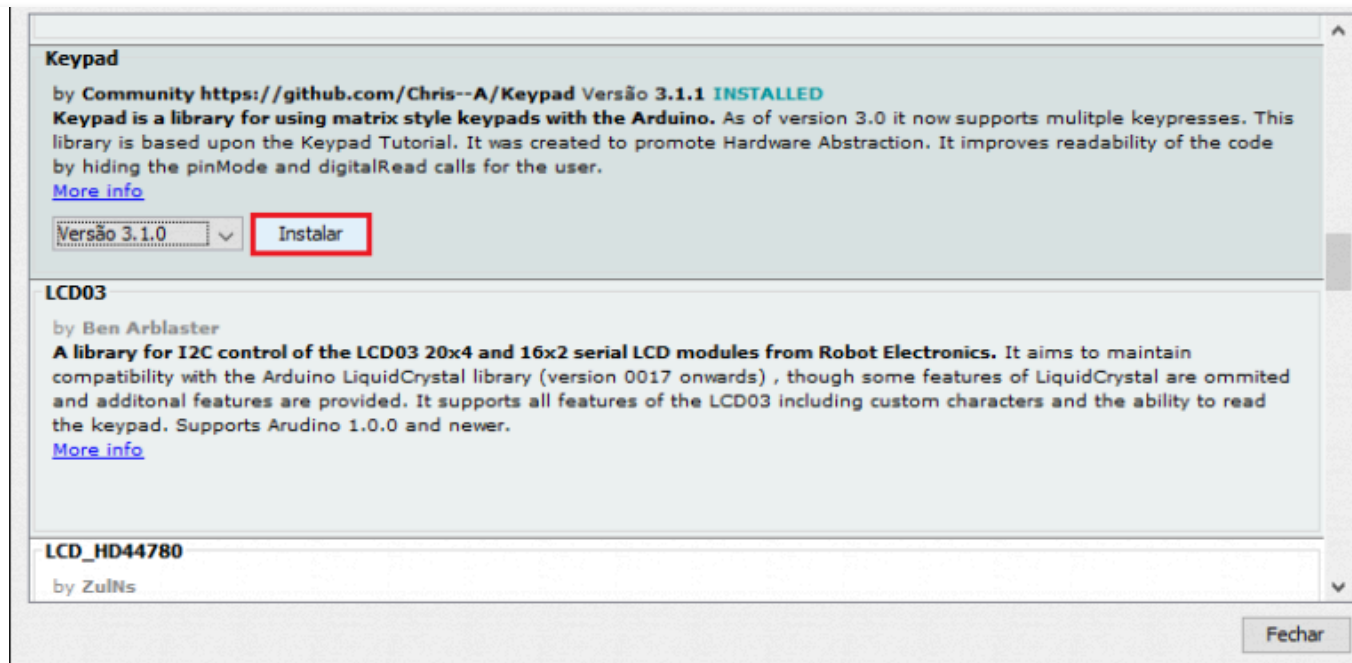


Figura 5 – Gerenciador de bibliotecas.

2. Incluir a biblioteca no código:

Iniciaremos incluindo a biblioteca através da instrução `#include <Keypad.h>`.

3. Declarar as variáveis:

Em nosso código, vamos declarar duas variáveis para armazenar o número de linhas e colunas do nosso teclado matricial, `numLinhas` e `numColunas`. Estas variáveis serão do tipo `const byte`, o que significa que elas são variáveis de leitura que podem armazenar um de 0 a 255.

pressionado no teclado. Você pode atribuir os caracteres que preferir.

Logo após, as portas 6, 5, 4 e 3 da placa UNO foram declaradas no array `pinosLinhas[numLinhas]` e as portas 8, 9, 10 e 11 no array `pinosColunas[numColunas]`.

4. Criar objeto para inicializar o teclado matricial:

As bibliotecas do Arduino normalmente se apresentam em classes que possuem funções para adicionar, configurar ou executar alguma tarefa de algum dispositivo. Desse modo, para usar o teclado matricial 4 x 4 keypad precisamos inicializá-lo.

Nesse caso, criamos o objeto **Keypad** `meuteclado = Keypad(makeKeymap(teclasMatriz), pinosLinhas, pinosColunas, numLinhas, numColunas);` em que:

- `makeKeymap`: inicializa a matriz de botões do teclado para ser igual a matriz `teclasMatriz` com os caracteres definidos pelo programador;
- `pinosLinhas` e `pinosColunas`: são os pinos da placa UNO conectados as linhas e as colunas do teclado matricial;
- `numLinhas` e `numColunas`: são o número de linhas e colunas do teclado.

5. Inicializar a comunicação serial:

Dentro da função `setup()`, inicializamos a comunicação serial através da instrução: **`Serial.begin(9600);`**



na função `loop()`, chamamos a variável `pressionado` para retornar o botão pressionado, se houver. Para verificar se o botão foi pressionado, usamos a estrutura condicional `if`. Se o botão for pressionado, o caractere correspondente será impresso no monitor serial.

Ao final, o código do projeto proposto ficará da seguinte maneira:

```
1.  /*
2.    =  TECLADO MATRICIAL 4x4 KEYPAD COM ARDUINO  =
3.    =====
4.    == BLOG DA ROBOTICA - www.blogdarobotica.com ==
5.    =====
6.    Autor: Carol Correia Viana
7.    E-mail: contato@blogdarobotica.com
8.    Facebook: facebook.com/blogdarobotica
9.    Instagram:@blogdarobotica
10.   YouTube: youtube.com/user/blogdarobotica
11.   =====
12.   == CASA DA ROBOTICA - www.casadarobotica.com ==
13.   =====
14.   Facebook: facebook.com/casadaroboticaoficial
15.   Instagram:@casadarobotica
16.   =====
17.  */
18.
19.  #include <Keypad.h> //Biblioteca para uso de teclados matriciais
20.
21.  const byte numLinhas = 4; //Quantidade de linhas do teclado
22.  const byte numColunas = 4; //Quantidade de colunas do teclado
23.
24.  //Matriz de caracteres referente aos botões do teclado
25.  char teclasMatriz[numLinhas][numColunas] = {
26.    {'1', '2', '3', '4'},
27.    {'5', '6', '7', '8'},
28.    {'9', '0', '*', '+'},
29.    {'#', '$', '%', '@'}
30.  };
```

```
34.  
35. //Cria um objeto do tipo Keypad  
36. Keypad meuteclado = Keypad(makeKeymap(teclasMatriz), pinosLinhas, pinosColunas, numLinhas, numColunas);  
37.  
38. void setup() {  
39.     Serial.begin(9600); //Inicia a comunicação serial  
40.     Serial.println("Aperte um botão..."); //Imprime mensagem no monitor serial  
41. }  
42.  
43. void loop() {  
44.     char pressionado = meuteclado.getKey(); ///Verifica se alguma tecla foi pressionada  
45.  
46.     if (pressionado) { //Se alguma tecla foi pressionada  
47.         Serial.print("Botão pressionado: "); //Imprime mensagem no monitor serial  
48.         Serial.println(pressionado); //Imprime tecla pressionada  
49.     }  
50. }
```

Espero que tenham gostado deste tutorial. Em caso de dúvidas, deixe seu comentário abaixo.

Carol Correia Viana



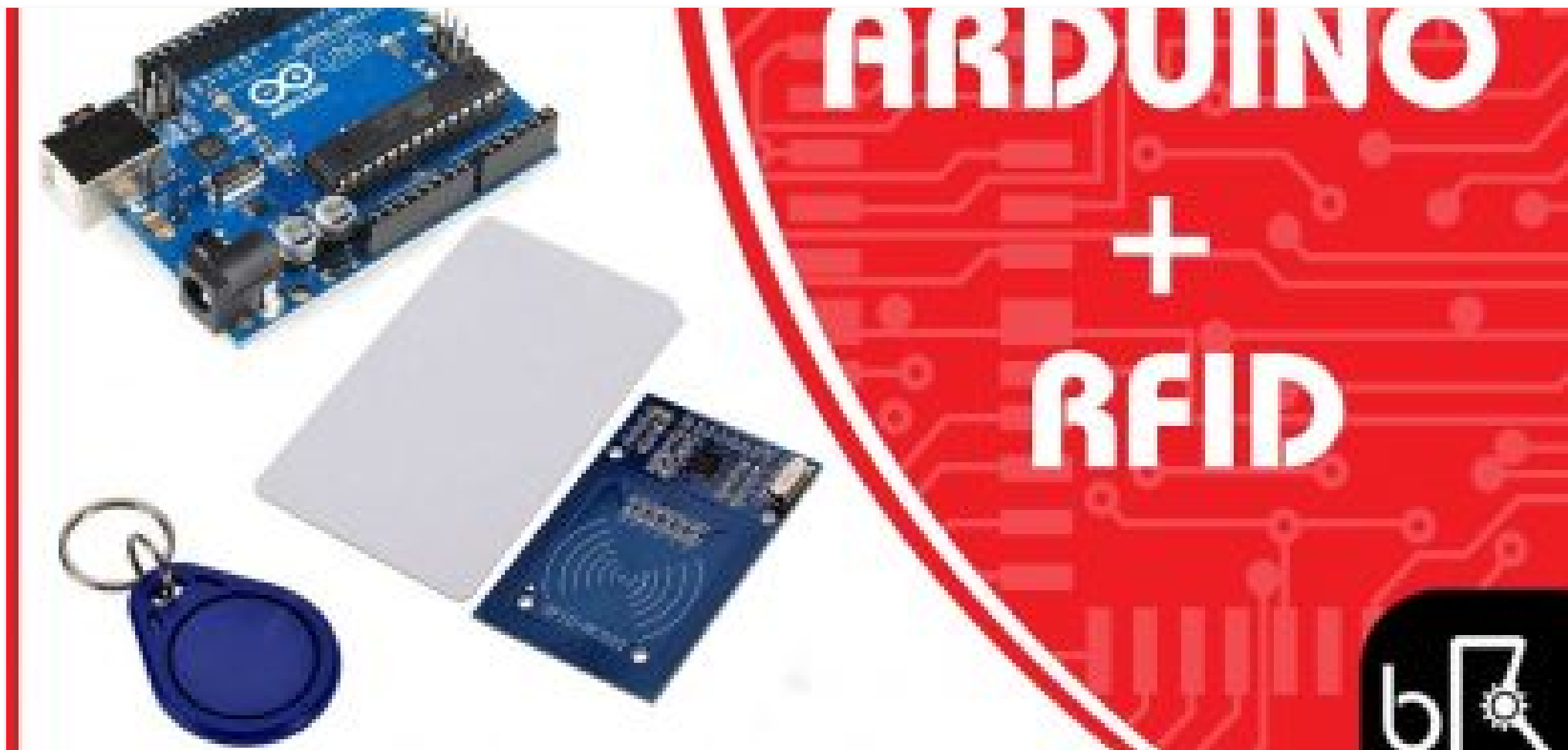
Carol Correia Viana

Bacharel em Engenharia Elétrica com ênfase em Eletrônica, mestra em Engenharia Industrial e especialista em Docência com ênfase em Educação Inclusiva. Atua no setor de Desenvolvimento de Produtos na Casa da Robótica. Editora chefe e articulista no Blog da Robótica. Fanática por livros, Star Wars e projetos Maker.

Detectando campo magnético com reed switch e Arduino →

 Você pode gostar também





Controle de Acesso via RFID e Arduino

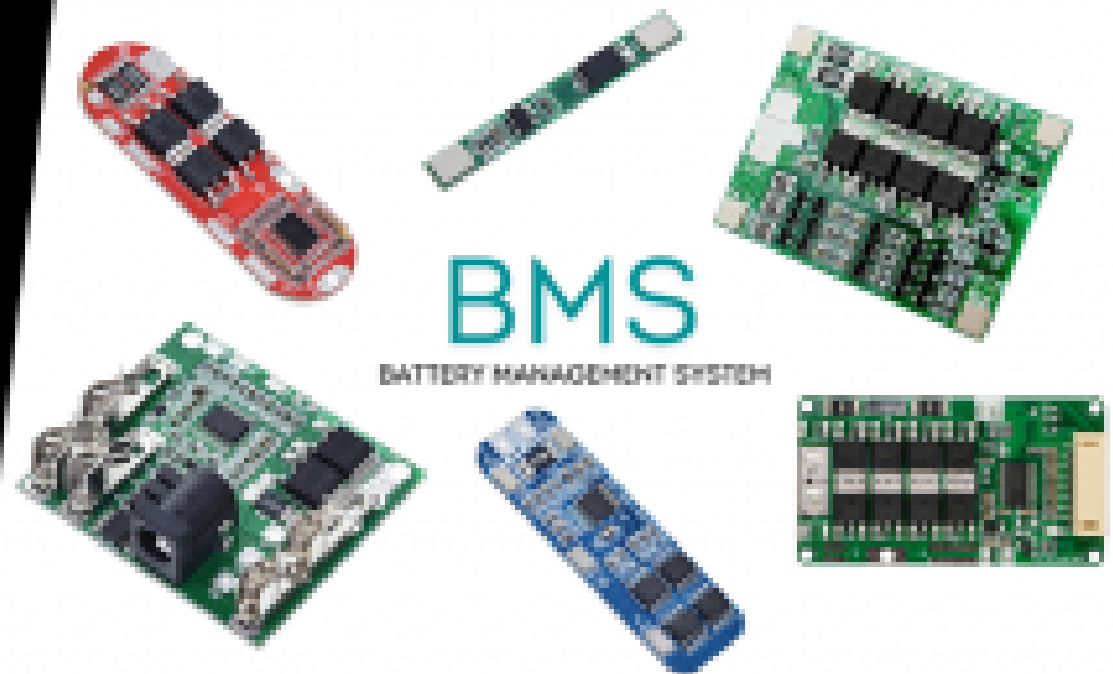
📅 20 de março de 2020



Expandindo as portas digitais do Arduino utilizando o registrador de deslocamento CI74HC595

📅 30 de junho de 2022

O QUE É E PARA QUE SERVE UMA **PLACA BMS?**



O que é e para que serve uma placa BMS?

📅 7 de março de 2025

Você precisa fazer o [login](#) para publicar um comentário.

Este site utiliza o Akismet para reduzir spam. [Saiba como seus dados em comentários são processados.](#)

Posts recentes

Jogo de Páscoa com Scratch e Makey Makey: Ajude o Coelho da Páscoa

O que é e para que serve uma placa BMS?

CONVERSORES / REGULADORES STEP UP: O QUE SÃO?

Circuito com massinha de modelar

Sobre



Páginas



Política de privacidade

Sobre

Nuvem de Tags

7segmentos **arduino** Arduino Uno attiny85 bc548 bluetooth hc-05 **BNCC** buzzer carregamento rápido ch340 DC-DC Dia Internacional da Mulher digispark dispositivos eletrônicos **Educação e Tecnologia** eficiência de conversão eficiência energética energia solar ESP8266 fontes de alimentação Inteligência Artificial LED Makey Makey **Micro:bit** NODEMCU **pensamento computacional** Pictoblox Planos de aula programação Arduino Programação em blocos projeto Arduino Projeto de robótica projeto eletrônico proteção contra curto-circuito raspberry pi pico **relé** RFID Robótica desplugada **Robótica Educacional** Scratch SemanadasCrianças Sensor de distância para Arduino sim900 sistemas de energia solar veículos elétricos

Apoio





Copyright © 2025 [Blog da Robótica](#). Todos os direitos reservados.

Tema: [ColorMag](#) por ThemeGrill. Powered by [WordPress](#).

