

# Teclado Matricial

Aprendemos a ligar chaves ao Arduino. Porém quando o número de chaves aumenta muito, é comum se usar uma configuração de matriz para as chaves de forma a diminuir o número de terminais necessários às ligações. Uma vez que simplificamos o hardware, a complexidade do software aumentará para poder trabalhar com este novo esquema. Por exemplo, um teclado 4x4 tem 16 chaves. Se ligássemos no esquema que aprendemos de 1 chave em 1 pino, teríamos que usar 16 pinos. No esquema matricial usamos apenas 8 pinos (4+4).

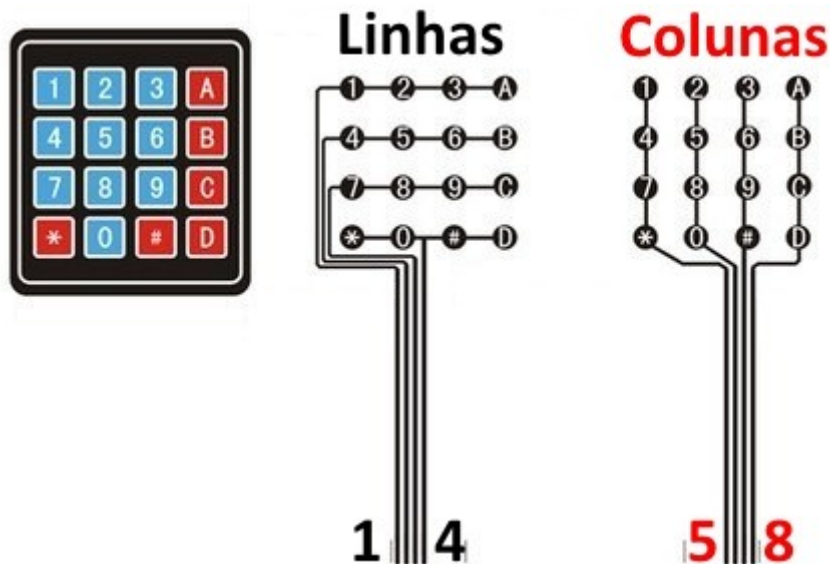
Como exemplo temos o teclado matricial de membrana, comumente usado nas aplicações com Arduino.



**Teclado matricial 4x4 de membrana**

## Pinagem Teclado Matricial 4X4:

Internamente são 16 teclas push-buttons tipo membrana dispostos na configuração abaixo em um formato keypad. Conforme a tecla é pressionada, é feita a conexão entre a linha e a coluna correspondentes. Se pressionarmos a tecla A no teclado matricial, será feita a conexão entre os pinos 1 (linha 1) e 8 (coluna 4), se pressionarmos a tecla 7, será feita uma conexão entre os pinos 3 (linha 3) e 5 (coluna 1), e assim por diante:



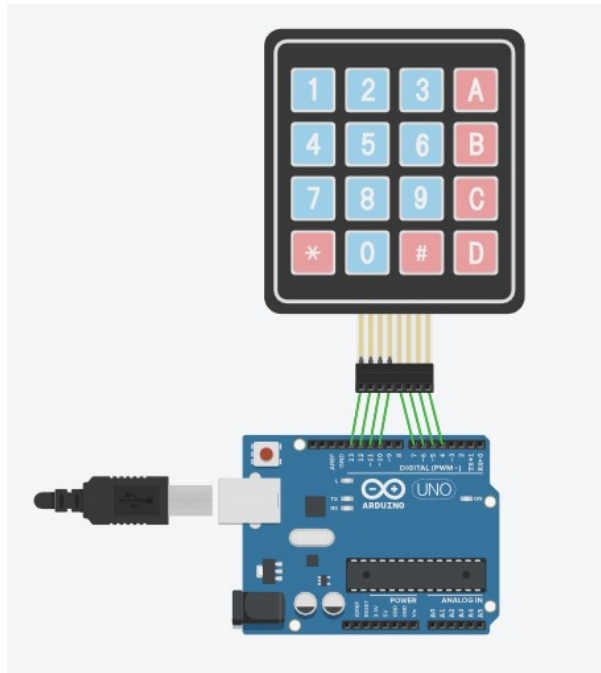
**Disposição das linhas e colunas de um teclado matricial**

Quando trabalhamos com 1 chave em 1 pino, sabemos que temos um circuito fechado quando a chave não está pressionada (através dos resistores de pull-up ou pull-down) e um outro circuito fechado quando a chave está pressionada. Para trabalhar com um teclado matricial, devemos usar as linhas como saída e as colunas como entrada (de forma invertida também funciona). De forma que se quisermos saber se uma chave foi pressionada, devemos fazer a varredura do teclado escrevendo nas linhas e lendo as colunas.

Se ligarmos as colunas como entrada e configurarmos os resistores de entrada como pull-up, se nenhuma chave estiver pressionada, sabemos que entrará o valor lógico 1 em todas as entradas. Portanto para saber se uma chave foi pressionada, devemos escrever 0 somente na linha à qual a chave será testada e verificar se a coluna correspondente foi zero também. Se escrevermos 0 em outra linha também, não saberemos se a chave que estamos testando foi pressionada ou se a chave da mesma coluna cuja linha foi posta em 0 foi pressionada. (Veja a figura abaixo com a ligação no Arduino). Portanto o procedimento para a varredura do teclado deve ser:

- Colocar 0 somente na linha da chave que se quer testar
- Ler a coluna correspondente à chave. Se for 0 é porque a chave foi pressionada (curto entre linha e coluna)
- Fazer isso para todas as outras chaves.

Por exemplo: se quisermos saber se a chave 6 do teclado foi pressionada, temos que escrever 0 somente na linha 2 e ler a coluna 3. Se for 0 é porque a chave foi pressionada (curto entre linha e coluna, portanto é lido na coluna o 0 que foi escrito na linha).



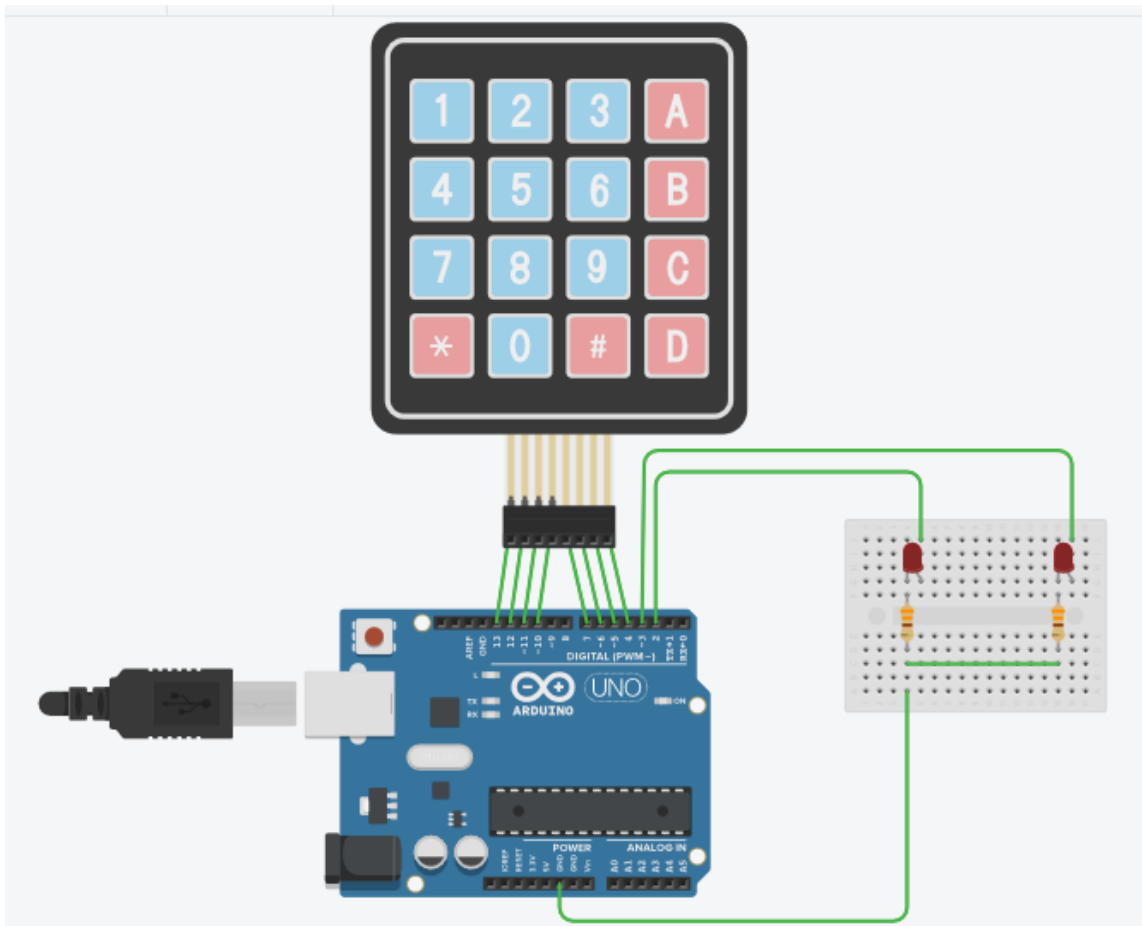
**Ligação de linhas e colunas no Arduino. As colunas devem ser configuradas como entradas com pull-up interno.**

Exemplo:

Usaremos um teclado 4x4 e dois LEDs. Quando a chave 4 for pressionada o LED 1 deverá acender e quando a chave 9 for pressionada, o LED 2 deverá acender. Caso contrário, os LEDs deverão ser apagados.

Note que neste exemplo não fazemos a varredura completa do teclado, vamos testar somente as chaves 4 e 9. Para este caso, nem precisaríamos ligar todas as linhas e colunas, bastando as linhas 2 e 3 e as colunas 5 e 7

Circuito:



Programa:

```
#define LIN1 13
#define LIN2 12
#define LIN3 11
#define LIN4 10

#define COL1 7
#define COL2 6
#define COL3 5
#define COL4 4

#define LED1 2
#define LED2 3
```

```

void setup() {
    pinMode (LED1, OUTPUT);
    pinMode (LED2, OUTPUT);

    pinMode (LIN1, OUTPUT);
    pinMode (LIN2, OUTPUT);
    pinMode (LIN3, OUTPUT);
    pinMode (LIN4, OUTPUT);

    pinMode (COL1, INPUT_PULLUP);
    pinMode (COL2, INPUT_PULLUP);
    pinMode (COL3, INPUT_PULLUP);
    pinMode (COL4, INPUT_PULLUP);

}

void loop() {
    // Teste da chave 4
    digitalWrite(LIN1, HIGH);
    digitalWrite(LIN2, LOW);
    digitalWrite(LIN3, HIGH);
    digitalWrite(LIN4, HIGH);
    // Para testar a chave 4, coloco apenas a linha 2 em 0

    if (digitalRead(COL1)==LOW) // chave 4 pressionada
        digitalWrite (LED1, HIGH);
    else
        digitalWrite (LED1, LOW);

    // Teste da chave 9

```

```
digitalWrite(LIN1, HIGH);  
digitalWrite(LIN2, HIGH);  
digitalWrite(LIN3, LOW);  
digitalWrite(LIN4, HIGH);  
// Para testar a chave 9, coloco apenas a linha 3 em 0  
  
if (digitalRead(COL3)==LOW) // chave 9 pressionada  
    digitalWrite (LED2, HIGH);  
else  
    digitalWrite (LED2, LOW);  
}
```

Faça o circuito, teste o programa e a seguir altere para que ele acenda o LED1 nos números de chaves ímpares e o LED2 acenda nos números de chaves pares.