

TECLADO MATRICIAL 4x4

Información: <http://www.prometec.net/teclados-matriciales/>

Un teclado es una matriz de teclas organizadas en filas y columnas. Suelen ser de 4x4, como el de la figura, pero si no pensamos utilizar las letras, podemos olvidarnos de la columna 4 y ahorramos el uso de un pin de Arduino.

Para utilizar este teclado con Arduino utilizamos la librería Keypad. La podemos descargar de:

<http://playground.arduino.cc/Code/Keypad#Download>

Antes de describir las principales funciones de esta librería veamos un ejemplo.

Ejemplo: este programa lee la tecla pulsada y la escribe en el monitor serie.

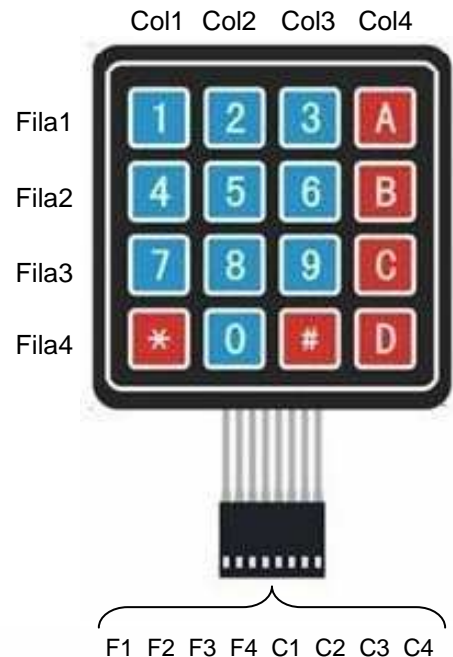
```
#include <Keypad.h> //añadimos la librería keypad
//creamos constantes para definir el número de filas y de columnas
const byte Filas = 4;
const byte Colum = 4;
//creamos una matriz con los caracteres del teclado
char teclas[Filas][Colum] = {
    {'1','2','3','A'},
    {'4','5','6','B'},
    {'7','8','9','C'},
    {'*','0','#','D'}
};

//indicamos a que pines de Arduino estan conectadas filas y columnas
byte pinFilas[Filas] = {9, 8, 7, 6}; //en orden F1, F2, F3 y F4
byte pinColum[Colum] = {5, 4, 3, 2}; //en orden C1, C2, C3 y C4

//creamos un objeto de la libreria keypad para usar sus funciones
Keypad Teclado = Keypad( makeKeymap(teclas), pinFilas, pinColum, Filas, Colum);

void setup(){
    Serial.begin(9600);
}

void loop(){
    //cuando se pulse una tecla, se guardará en una variable llamada tecla
    char tecla = Teclado.getKey();
    if (tecla != NO_KEY){ // NO_KEY es una constante definida en Keypad.h
                        // equivale a 0. Sería igual poner if(tecla != 0)
        Serial.println(tecla); //Si se ha pulsado una tecla, se envia por serial
    }
}
```



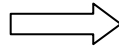
Hay que tener en cuenta que todos los signos leídos del teclado son caracteres, no números, por lo que no se pueden hacer operaciones con ellos, como sumas o comparaciones con > o <. Para ello los tendríamos que **convertir en números**. Una forma fácil para los números de 0 a 9 es definir una variable tipo *int* y realizar la operación de restarle 48 o bien el carácter '0' a la variable

que guarda el carácter (la variable `tecla` en este ejemplo), ya que con ello se pasa del código ASCII del carácter numérico al valor numérico. Es decir, las instrucciones serían:

<pre>int numero; numero = tecla - 48;</pre>	o bien:	<pre>int numero; numero = tecla - '0';</pre>
---	---------	--

Nota: El código ASCII del carácter '0' es 48.

Si vamos a utilizar sólo los números del teclado, podemos definir un teclado con 3 columnas. Así nos ahorramos un pin de Arduino.



```
const byte Filas = 4;
const byte Colum = 3;
char teclas[Filas][Colum] = {
    {'1', '2', '3'},
    {'4', '5', '6'},
    {'7', '8', '9'},
    {'*', '0', '#'} };
byte pinFilas[Filas] = {9, 8, 7, 6};
byte pinColum[Colum] = {5, 4, 3};
```

LIBRERÍA KEYPAD

La librería Keypad dispone de varias funciones. Las que más usaremos son:

- Keypad

Crea un objeto tipo Keypad para controlar nuestro teclado.

La sintaxis es: **Keypad** Teclado = **Keypad** (**makeKeymap** (teclas), pinF, pinC, nF, nC)

donde:

- *Teclado*: es el nombre que le damos al objeto tipo Keypad creado.
- *teclas*: es el nombre de una matriz bidimensional de caracteres previamente definida donde asociamos un carácter a cada una de las teclas del teclado.
- *pinF*: es una matriz unidimensional previamente definida donde hemos indicado los pines donde van conectadas las columnas del teclado en el orden F1, F2, F3 y F4.
- *pinC*: es una matriz unidimensional previamente definida donde hemos indicado los pines donde van conectadas las filas del teclado en el orden C1, C2, C3 y C4.
- *nF*: es un número o una variable que contiene el número de filas del teclado.
- *nC*: es un número o una variable que contiene el número de columnas del teclado.

- .getKey()

Lee y devuelve el carácter correspondiente a la tecla presionada si se ha presionado alguna. Si no se ha pulsado ninguna tecla devuelve un 0.

La sintaxis es: Teclado.**getKey**()

donde *Teclado* es el nombre del objeto tipo Keypad.

- .waitForKey()

Esta función esperará indefinidamente hasta que alguien pulsa una tecla. Esto significa que el resto del código no se ejecuta durante la espera, salvo que se trate de una rutina de interrupción; le ocurre lo mismo que a la función `delay()`. Una vez presionada la tecla, la función lee y devuelve el carácter correspondiente a la tecla presionada.

La sintaxis es: Teclado.**waitForKey**() donde *Teclado* es el nombre del objeto tipo Keypad.

CAPTURA DE VARIAS PULSACIONES

Habr  ocasiones en las que nos interese capturar varias pulsaciones para formar cadenas de caracteres. Por ejemplo, para teclear una clave de acceso para que se abra una puerta.

En este caso definiremos variables tipo cadena (String) a las cuales se puede acceder de forma independiente a cada car cter de la cadena e iremos introduciendo los caracteres pulsados en el teclado.

Ejemplo: Este programa lee el teclado de forma constante. Cuando se pulsa la tecla # captura los caracteres correspondientes a las tres siguientes teclas pulsadas y sustituye con ellos los caracteres de la cadena *clave*. A continuaci n, imprime la clave y, en caso de que  sta coincida con alguna de las claves v lidas contenidas en la matriz de cadenas *claves_validas*, muestra el mensaje "Esta clave es v lida" y as  sucesivamente.

```
#include <Keypad.h>
String clave = "000";
char inicia_clave = '#';
String claves_validas[]={ "426", "010", "766"};
const byte Filas = 4;
const byte Colum = 4;
char teclas[Filas][Colum] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
byte pinFilas[Filas] = {9, 8, 7, 6}; //pines de filas en orden F1, F2, F3 y F4
byte pinColum[Colum] = {5, 4, 3, 2}; //pines de columnas en orden C1, C2, C3 y C4
Keypad Teclado = Keypad( makeKeymap(teclas), pinFilas, pinColum, Filas, Colum);

void setup(){
  Serial.begin(9600);
  Serial.println("Empezamos");
}

void loop(){
  int n=0; // cada vez que iniciamos el loop reiniciamos el contador
  //cuando se pulse una tecla, se guardar  en una variable llamada tecla
  char tecla = Teclado.getKey();
  if(tecla != 0){ // Si se pulsa alguna tecla
    if(inicia_clave==tecla){ //Si la tecla es # se inicia la captura
      while(n<3){ //se capturan tres pulsaciones y se introducen en clave
        clave[n]=Teclado.waitForKey();
        n=n+1;
      }
      Serial.println(clave); //Se imprime la clave pulsada
      for (int i=0;i<=2;i++){ //se mira si coincide con alguna clave v lida
        if(clave==claves_validas[i])Serial.println("Esta clave es valida");
      }
    }
  }
}
```

USO SIMULTÁNEO DEL TECLADO PARA INTRODUCIR CLAVES Y PARA ACTIVAR Y DESACTIVAR SALIDAS POR PULSACIÓN DIRECTA DE TECLAS.

En algunos proyectos donde disponemos de teclado, puede interesarnos que determinadas salidas se activen y/o desactiven mediante claves de varios dígitos (por ejemplo desactivación de sistema de alarma, apertura de puertas de acceso, etc.) mientras que otras salidas queremos que se activen mediante la simple pulsación directa de una tecla (por ejemplo luces, un ventilador, activación de sistema de alarma, etc.). Para ello, una buena opción es reservar las teclas alfabéticas del teclado (A, B, C y D) para órdenes directas, las teclas # y * para indicar que iniciamos la introducción de una clave numérica y las teclas numéricas para las claves.

Con este sistema sólo podríamos ordenar cuatro acciones de forma directa con una sola tecla. Si quisiéramos más tendríamos que recurrir a la pulsación de dos teclas, por ejemplo A1, A2,...

Ejemplo

En este ejemplo el programa lee el teclado continuamente; las teclas A, B, C y D se reservan para funciones directas. La tecla # se utiliza para indicar que se va a introducir una clave numérica de tres dígitos. La tecla * para indicar que se va a introducir una clave de un dígito.

Para este caso concreto sólo se le ha dado función a la tecla alfabética A, que enciende y apaga un LED conectado al pin 10. Las otras teclas alfabéticas se han dejado sin función.

La tecla # inicia la introducción de una clave de 3 dígitos. Si es correcta (hay tres claves correctas) encenderá o apagará un LED situado en el pin 11. Si no es correcta no hará nada.

La tecla * inicia la introducción de una clave de 1 dígito. Si el dígito es 1 encenderá o apagará un LED conectado en el pin 12, si es un 2 encenderá o apagará un LED conectado al pin 13 y si es otro número no hará nada.

Nota importante: en este programa no podemos utilizar la función `.waitForKey()` para leer el teclado, como hicimos en el anterior ejemplo, pues una vez que empezáramos a introducir la clave numérica hasta que termináramos, el programa no respondería a las órdenes dadas por las teclas directas.

```

/*
Programa que controla las pulsaciones en un teclado y permite encender
o apagar el LEDa a pulsar la tecla A; verificar si una clave numérica
de tres dígitos, precedida de #, es correcta o incorrecta y, en caso de
ser correcta, encender o apagar el LEDb; encender o apagar el LEDc si se
pulsar la tecla * seguida de 1 y encender o apagar el LEDd si se pulsa la
tecla * seguida de 2. Las teclas alfabéticas de orden directa pueden
pulsarse incluso durante la introducción de las claves numéricas y después
seguir introduciendo éstas.
*/
#include <Keypad.h>
const byte Filas = 4;  const byte Colum = 4;
char teclas[Filas][Colum] = {
    {'1', '2', '3', 'A'},
    {'4', '5', '6', 'B'},
    {'7', '8', '9', 'C'},
    {'*', '0', '#', 'D'}
};

```

```

byte pinFilas[Filas] = {9, 8, 7, 6}; //pines de filas en orden F1, F2, F3 y F4
byte pinColumn[Column] = {5, 4, 3, 2}; //pines de columnas en orden C1, C2, C3 y C4
Keypad Teclado = Keypad( makeKeymap(teclas), pinFilas, pinColumn, Filas, Column);
//definimos que la clave tiene 3 dígitos dándole un valor inicial
String clave = "000";
String claves_validas[]={ "426", "010", "766"};
int estadoteclado=0;
/*
La variable estadoteclado sirve para guardar el estado del teclado.
Vale 0 si no se ha introducido ningún carácter de inicio de clave (# o *),
10 cuando se ha introducido el carácter # de inicio de clave de tres
dígitos, 11 cuando se ha introducido el primer dígito de la clave numérica
de tres dígitos y 12 cuando se ha introducido el 2º dígito (en este estado,
cuando ya se introduce el tercer dígito comprueba la clave) y 20 cuando
se ha introducido el carácter * de inicio de clave de un dígito.
*/
int clavecorrecta=0; //Se pone a 1 cuando la clave introducida es correcta
//Definimos los pines donde estarán conectados los LEDs
const int LEDa=10; const int LEDb=11; const int LEDc=12; const int LEDd=13;
//indicamos que los LEDs empiezan apagados
int estadoLEDa=0; int estadoLEDb=0; int estadoLEDc=0; int estadoLEEd=0;
//Definimos el teclado

void setup(){
  Serial.begin(9600);
  Serial.println("Empezamos");
  pinMode(LEDa, OUTPUT); digitalWrite(LEDa, LOW);
  pinMode(LEDb, OUTPUT); digitalWrite(LEDb, LOW);
  pinMode(LEDc, OUTPUT); digitalWrite(LEDc, LOW);
  pinMode(LEDd, OUTPUT); digitalWrite(LEDd, LOW);
}

void loop(){
  char tecla = Teclado.getKey();
  if(tecla != 0){ // Si se pulsa alguna tecla
    switch(tecla){ //dependiendo de la tecla hago...
      case 'A':
        // instrucciones si se pulsa A
        if(estadoLEDa==0){digitalWrite(LEDa, HIGH); estadoLEDa=1;}
        else{digitalWrite(LEDa, LOW); estadoLEDa=0;}
        break;
      case 'B':
        // instrucciones si se pulsa B
        break;
      case 'C':
        // instrucciones si se pulsa C
        break;
      case 'D':
        // instrucciones si se pulsa D
        break;
    }
  }
}

```

```

    case '#':
        estadoteclado=10;
        break;
    case '**':
        estadoteclado=20;
        break;
    default:
        switch(estadoteclado){//dependiendo del estado del teclado hago...
            case 10:
                clave[0]=tecla;
                estadoteclado=11;
                break;
            case 11:
                clave[1]=tecla;
                estadoteclado=12;
                break;
            case 12:
                clave[2]=tecla;
                estadoteclado=0;
                Serial.print(clave); //Se imprime la clave pulsada
                for (int i=0;i<=2;i++){//se mira si coincide con alguna clave válida
                    if(clave==claves_validas[i]) clavecorrecta=1;
                }
                if(clavecorrecta==1){
                    Serial.println(" Esta clave es valida");
                    if(estadoLEDb==0){digitalWrite(LEDb,HIGH); estadoLEDb=1;}
                    else{digitalWrite(LEDb,LOW); estadoLEDb=0;}
                }
                else Serial.println(" Clave incorrecta");
                clavecorrecta=0;
                break;
            case 20:
                switch (tecla){
                    case '1':
                        if(estadoLEDb==0){digitalWrite(LEDb,HIGH); estadoLEDb=1;}
                        else{digitalWrite(LEDb,LOW); estadoLEDb=0;}
                        break;
                    case '2':
                        if(estadoLEDb==0){digitalWrite(LEDb,HIGH); estadoLEDb=1;}
                        else{digitalWrite(LEDb,LOW); estadoLEDb=0;}
                        break;
                }
                estadoteclado=0;
            }
        }
    }
}

```