

TECLADOS MATRICIALES


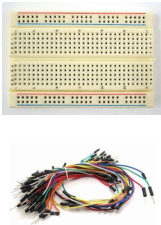

Arduino y los keypads de 4x4

Home • Teclados Matriciales

OBJETIVOS

- ★ Familiarizarnos con un teclado matricial de 4×4.
- ★ Usar otra librería externa: KeyPad.
- ★ Volver a revisar los arrays, esta vez de dos dimensiones.

MATERIAL REQUERIDO.

	Arduino Uno o similar. Esta sesión acepta cualquier otro modelo de Arduino.
	Una Protoboard más cables.
	Un teclado matricial 4×4

COMO FUNCIONA UN TECLADO MATRICIAL

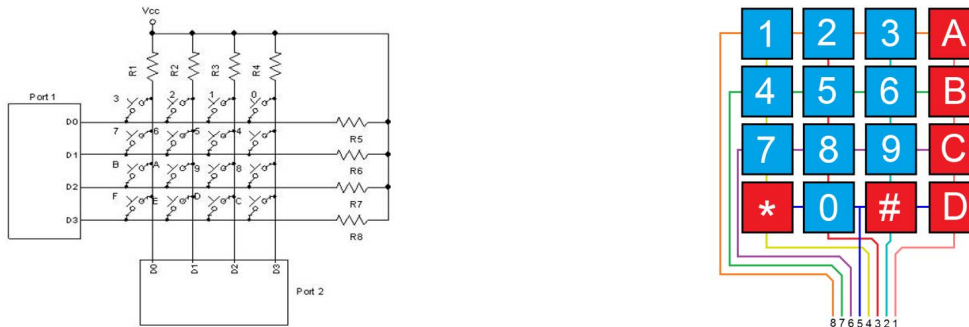
Un teclado no es más que una colección de botones, a cada uno de los cuales le asignamos un símbolo o una función determinada. Pero botones al fin y al cabo.

Leer botones es algo que ya no tiene secretos para nosotros, pero si conectáramos cada tecla a un pin digital de nuestro Arduino, pronto estaríamos en apuros.

El teclado de nuestro ordenador suele ser de alrededor de 106 teclas, así que el método de fuerza bruta va a entrar en apuros rápidamente. Necesitamos otra solución.

Y como el mundo está lleno de gente ingeniosa se les ocurrió una solución de lo más elegante, una **matriz de teclas**.

Vamos a ver un ejemplo con un pequeño **teclado numérico** de 16 teclas tipo los de los teléfonos móviles o los de los cajeros automáticos.



Para que nuestro Arduino pueda saber que tecla se pulsa, basta con poner tensión en las filas de forma secuencial y luego leer las columnas para ver cuál de ellas tiene HIGH. Los **teclados matriciales** usan una combinación de filas y columnas para conocer el estado de los botones. Cada tecla es un pulsador conectado a una fila y a una columna. Cuando se pulsa una de las teclas, se cierra una conexión **única** entre una fila y una columna.

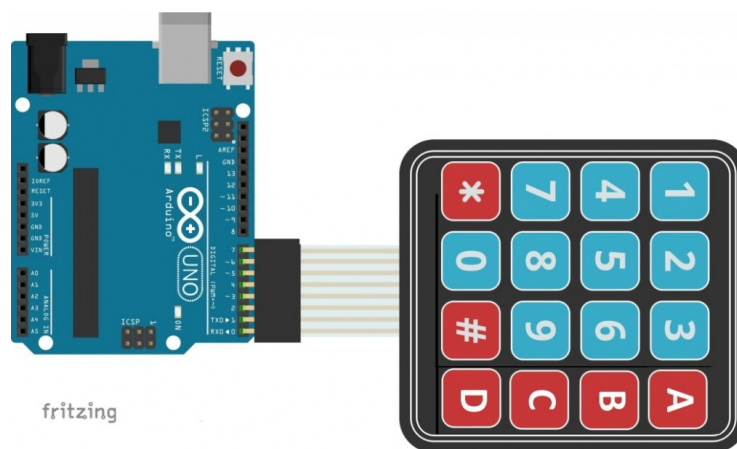
Por ejemplo, ponemos HIGH en la primera fila (hilo 8 en el diagrama de la derecha) y después leemos sucesivamente los hilos correspondientes a las columnas (hilos 4, 3, 2, 1). Si ninguno está en HIGH es que no se ha pulsado ninguna tecla de la primera fila.

Pasamos a la segunda fila (hilo 7) y ponemos HIGH, si al leer los hilos 4, 3, 2, 1 encontramos que el hilo 1 está en HIGH, es que se ha pulsado la tecla correspondiente a la "B".

De este modo, para leer un teclado matricial de 4x4 necesitamos 8 hilos en lugar de 16, aunque nos dará un poco más de guerra a la hora de programar. Para un teclado de PC 106 teclas bastaría una matriz de 10x11 o sea 21 hilos en vez de 106.

DIAGRAMA DE CONEXIÓN

Lo más práctico es conectar el teclado a Arduino directamente, podemos usar cables de protoboard, o bien con un peine de pines para que no se suelte nada.



Esta es otra de esas raras ocasiones en que un esquema electrónico tendría poco sentido, así que pasaremos rápidamente a ver el tema del programa.

- ✓ *Un apunte rápido: Por comodidad en el diagrama, y porque en los ejemplos que vienen con Arduino utilizan estos mismos números de pines del 0 al 7, los hemos mantenido. Pero conviene saber que Arduino usa los pines 0 y 1 para comunicarse a través del USB, por lo que no es buena usarlos para otra cosa cuando vayamos a usar el puerto serie.*

EL PROGRAMA DE LECTURA

Como ya dijimos, el programa ha de consistir en ir dando tensión consecutivamente a los pines de las filas 8, 7, 6, 5 y después ir leyendo los valores de las columnas para detectar que teclas se han pulsado.

No es un programa complicado, pero antes de que el pánico se apodere de los que siguen la sesión con creciente temor (*y recuerden que tienen algo urgente e ineludible que hacer*) comentaremos que alguien ha tenido este problema antes que nosotros y amablemente ha puesto a nuestra disposición una librería llamada Keypad ideal para leer este tipo de matrices sin complicaciones..

- ✓ Lo que no quita para que podáis escribir un programa que lea directamente la matriz y reconozcáis las pulsaciones. Animo, no es difícil y el ejercicio siempre es bueno.

Lo primero es descargar la librería e instalarla. La tenéis aquí [Descargar](#) , o bien en la página de Arduino

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

Id a la sección de "Download, install and import" donde encontrareis keypad.zip. Descargadlo y para instalarla recordad que basta con hacer:



En la ventana que sale, buscad el fichero Keypad.zip que habéis descargado, seleccionadlo y listo.

Una vez instalada para usarla basta con que le indiquéis que deseáis usarla. Si hacéis Programa\Importar Librería \ Keypad veréis que incluye la siguiente línea en vuestro programa:

```
#include <Keypad.h>
```

Vamos a ver como usamos esta librería. Lo primero es definir un par de constantes

```
const byte Filas = 4;          //Keypad de 4 filas
const byte Cols = 4;          //y 4 columnas
```

Y ahora una par de arrays para indicarle a la librería que pines de Arduino corresponden a las filas y cuales a las columnas del keypad:

```
byte Pins_Filas[] = {7, 6, 5, 4}; //Pines Arduino para las filas.
byte Pins_Cols[] = { 3, 2, 1, 0};  // Pines Arduino para las columnas.
```

Recordad que un array es una colección de elementos que pasamos entre llaves y separados por comas. Es más cómodo que andar definiendo 8 variables.

Recordad lo que vimos en la sesión 9, que cuando definimos una array por enumeración de sus elementos, como en este caso, no necesitamos pasar entre corchetes el número de elementos (Los cuenta el mismo C++ solito).

Tendremos que definirle, además, que símbolos corresponden a cada posición de las teclas. Una nueva oportunidad para disfrutar de los arrays, pero esta vez de 2 dimensiones en vez de una.

```
char Teclas [ Filas ][ Cols ] =
{
    {'1','2','3','A'},
    {'4','5','6','B'},
    {'7','8','9','C'},
    {'*','0','#','D'}
};
```

- ✔ Definimos el array de 4×4 pasándole las dimensiones entre corchetes. Sus miembros van a ser 4 filas, cada una de las cuales es un array de 4 elementos, y después enumeramos los cuatro arrays de 4 miembros.
- ✔ Fijaros en que cada array va entre llaves. Los elementos de un array se separan por comas, pero al final de las llaves exteriores va un punto y coma porque toda el conjunto es una asignación.
- ✔ Los arrays siempre parecen complicados hasta que te acostumbras pero no es para tanto y este es un buen ejemplo para que comprendáis la idea de un array de 2 dimensiones.

Una de las curiosidades de leer una matriz de teclas y después asignar un carácter contenido en un array es que podemos redefinir el valor de las teclas sin más que cambiar el contenido del array.

- ✔ Si por lo que sea hubiéramos conectado mal los pines del keypad al Arduino, podríamos redefinir los valores del array para que coincidan con lo que pone en las teclas. Puede ser más fácil que mover de sitio los cables.
- ✔ De hecho es esa capacidad de mapear la representación de un carácter a una posición en una tabla, es lo que permite que podamos reconfigurar el mismo teclado a diferentes lenguas, sin más que cambiar los valores del array (aunque también ayudará mucho luego cambiar los símbolos rotulados en el teclado).

Por ultimo tendremos que crear una instancia de Keypad que llamaremos Teclado1

```
Keypad Teclado1 = Keypad(makeKeymap(Teclas), Pins_Filas, Pins_Cols, Fils, Cols);
```

Que leído así asusta, pero que en realidad es una tontería. Traduciendo sería: Crea una instancia del tipo Keypad que llamas Teclado1 y al que asignarás las teclas que tenemos en el array Teclas, y le decimos que hemos conectado las filas del keypad a los números de pines que te indico en el array Pins_Filas y las columnas al array Pins_Cols.

- ✔ MakeKeymap (Teclas) es una función disponible en la librería y que nos permite asignar un array de valores a una matriz. Se hace solo una vez al principio salvo que por lo que sea nos interese cambiarlo sobre la marcha.

Por ultimo para leer el keypad hacemos una llamada a otra función de la librería:

```
char pulsacion = Teclado1.getKey() ;
```

Por ultimo enviamos a la consola el carácter pulsado. Aquí tenéis el programa en versión completa: [Descargar](#)

```
#include <Keypad.h>           // Prog_19_1

const byte Filas = 4;         //Cuatro filas
const byte Cols = 4;          //Cuatro columnas

byte Pins_Filas[] = {7, 6, 5, 4}; //Pines Arduino para las filas
byte Pins_Cols[] = { 3, 2, 1, 0};  // Pines Arduinopara las columnas
char Teclas [ Filas ][ Cols ] =
{
    {'1','2','3','A'},
    {'4','5','6','B'},
    {'7','8','9','C'},
    {'*','0','#','D'}
};

Keypad Teclado1 = Keypad(makeKeymap(Teclas), Pins_Filas, Pins_Cols, Filas, Cols);

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

```
void loop()
{
    char pulsacion = Teclado1.getKey() ;
    if (pulsacion != 0)           // Si el valor es 0 es que no se
        Serial.println(pulsacion); // se ha pulsado ninguna tecla
}
```

Por cierto, en C++ y en otros lenguajes es muy frecuente escribir la instrucción:

```
if (pulsación != 0)
```

De esta otra manera (más elegante) *completamente equivalente*:

```
if (pulsación )
```

Porque la condición del if *debe evaluarse a* TRUE o FALSE y para los ordenadores un 0 es FALSE sin remisión, pero cualquier otro valor que no sea 0 es TRUE, incluyendo un carácter.

RESUMEN DE LA SESIÓN

- ★ Hemos usado nuestro primer keypad hexadecimal
 - ★ Son prácticos y muy sencillos de usar en cuanto le pilléis el tranquillo.
 - ★ Están disponibles en múltiples formatos y tipos, Aquí hemos usado uno típico de 4×4, pero también los hay 4×3 y en diferentes materiales y acabados
 - ★ Un antiguo teléfono móvil roto se puede desmontar y usar su teclado además de otras cosas, y se puede utilizar de la forma que hemos descrito.
- ★ Hemos usado la excusa del teclado para volver sobre los arrays. Vale la pena de que les dediquéis un poco de tiempo pues os ayudaran a resolver cómodamente más de un problemas de programación, y no se merecen la fama de malotes que tienen.
- ★ Hemos conocido otra librería externa, KeyPad. Os vendrá muy bien en vuestros proyectos. Más adelante confío en que podamos hacer algún ejemplo más sofisticado con un keypad como hacer que una cierta combinación abra una cerradura de servo por ejemplo.

[ANTERIOR](#)[SIGUIENTE](#)

(34) COMMENTS



Mario Roberto

08 Ene 2015

Excelente, muchas gracias. Me ha servido mucho.

[Reply](#)



Mazah

14 Mar 2015

Muy buena web, le he estado echando un ojo a varias de las sesiones y la verdad que esta todo muy bien explicado. Gracias por el trabajo.

[Reply](#)



Michel

09 May 2015

Oye disculpa pero el codigo solo agarra 1 y 2 , de ahi para abajo, no agarra las letras ni el 3 para abajo es normal?

Reply



admin

10 May 2015

Hola Michel,

Reply

Ese problema indica que algunas de las conexiones no estan bien y por eso hay letras que no fncionan. Revisa las conexiones para asegurarte y veras que pronto tienes tu teclado en marcha.



Alberto Palomero

15 May 2015

Yo tenia el mismo problema, Solo me entraban los numeros correspondientes a las columnas 1 y 2 (cables 4 y 3), las columnas 3 y 4 (cables 2 y 1) no me entraban.

Reply

Si te fijas los cables 2 y 1 estan en los pines 0 y 1 de Arduino, como bien dicen al principio del post, esto puede ocasionar problemas porque son los pines utilizados para la comunicacion con arduino. En mi caso era asi. Al cambiarlos a otros pines todo funciona perfectamente. Tu caso tiene pinta de ser igual que el mio porque te fallan justamente las mismas teclas, las correspondientes a cables 2 y 1.

Saludos



admin

15 May 2015

Reply

No me había dado cuenta en la pregunta previa, pero si conectas algunas de las filas o columnas de la matriz a los pines 0 o 1 de tu ARduino UNO (Parece un trabalenguas) que son los que se usan para la comunicacion USB, cuando leas el teclado e intentes enviar por el USB, para transmitir la tecla pulsada, interferiran entre si ambas funciones provocandote errores.

Por eso, evita en la medida de lo posible el uso de los pines 0 y 1 en tus programas y muchos problemas raros desapareceran



Jicamasconchile

10 Jun 2015

Hola!, genial que compartas el conocimiento!

Reply

Bueno, yo me he estado escribiendo una mini-librería para controlar mi teclado (es el mismo que se usa aquí), todo iba tan bien hasta que empecé con la parte de detectar pulsaciones múltiples, solo puedo detectar pulsaciones múltiples en las filas (1,2,3,4,A) tres al mismo tiempo, pero en las columnas(1,4,7,*) solo puedo detectar una pulsación, he estado pensando y aun nada ://

conocerás alguna forma de detectar las pulsaciones múltiples?, puedes pasar un ejemplo?, no se puede por como esta diseñado el teclado?.

PD: no he buscado en google, pero me gustaría conocer tu opinión.

Eres genial!



admin

10 Jun 2015

Reply

Hola Alex, Debe ser posible, porque aunque no le he dedicado una pensada, la version actual de la libreria keypad dice que reconoce multiples pulsaciones.

Aqui tienes la nota para que le eche suna ojeada si quieres: <http://playground.arduino.cc/Code/Keypad>

Un saludo



Jicamasconchile

12 Jun 2015

Reply

Me he visto la librería y al parecer solo detecta múltiples pulsaciones de las filas o las columnas pero no ambas, depende del modo en que pones los pines de las filas y las columnas; si configuras las columnas como pines de entrada solo podrás detectar múltiples pulsaciones en las columnas y si pones las filas como pines de entrada solo podrás detectar múltiples pulsaciones en las filas. Y eso es por la estructura del teclado, todas mis pruebas me dan como resultado un corto cuando presionas mas de una tecla de una misma fila/columna que este en modo de salida.

También he buscado en google y al parecer si usas unos diodos puedes evitar el corto que se produce en los pines de salida cuando presionas mas de una tecla. aquí puedes verlo -> forum.arduino.cc/index.php?topic=107430.0

Me gustaría que le dieras una pensada y si estoy mal me lo hicieras saber.

Gracias por contestar!

Que gran página!



Manuelez

19 Sep 2015

Reply

Buenas noches, muy bien explicado tu foro! Tengo una consulta, trato de hacer que arduino detecte mientras no se haya pulsado ninguna tecla en 10seg se encienda un led. Mientras se pulse cualquier tecla en ese tiempo el led esté en modo OFF.

Por otra parte primero lo intenté con un pulsador y me sale genial pero con el teclado no. No sé si podrías orientarme o tener una idea más amplia de como podría realizar aquello.

De antemano muchas gracias



admin

20 Sep 2015

Reply

Manuel, si con un pulsador ya lo tienes, con el teclado es igual de facil. Basta con que lo leas y si el resultado es una tecla mantienes el mismo codigo. Si quieres pasa el programa al foro y lo miramos alli.

Un saludo



Alex

20 Sep 2015

Reply

Cómo puedo hacer esto en AtmelStudio? Pero en C.



admin

20 Sep 2015

Reply

Pues me temos que ni idea porque no he usado nunca el aatmel studio alex



Alfonso Jódar

13 Oct 2015

Reply

Hola, gracias por tu trabajo es increíble, de los mejores tutoriales que hay para aprender Arduino.

Mi pregunta es, estoy intentando un programador de riego y me gustaría, en un apartado leer una entrada Analógica (Sensor de Humedad) y en otro apartado escribir un valor por teclado para que cuando el sensor de ese valor se encienda una electroválvula de riego. El caso es que no encuentro nada para escribir desde teclado en una variable y que se guarde para su posterior comparación con el sensor. Me puedes echar una mano? Muchas gracias.



admin

14 Oct 2015

Reply

Hola Alonso, estoy seguro de que te podemos ayudar a resolver ese problema. Te propongo que lo pantees en la sección de proyectos del foro y allí entre todos seguro que podemos ayudarte. UN saludo



Timo

17 Oct 2015

Reply

Hola admin, para ser sincero desde un principio empeze hace menos de 2 dias a usar arduino (Tengo un nivel de electronica bajo y de programacion solo se usar c#) y me entretengo mucho investigando codigos de ideas que se me vienen a la mente para hacer con esta placa.

Ahora, sabiendo como conectar un LCD 16x2 a la arduino, quisiera saber, si usted me puede indicar que codigo debo usar para poder hacer que este teclado 4x4 me permita, mediante yo presionando por ejemplo la tecla 1, me la imprima en pantalla. Seria de muy gran ayuda porque para empezar no quiero usar la arduino para escribir en el codigo "lcd.print" con el teclado de la PC para que aparesca en el LCD, sino de usar este teclado para escribir algo, y de verdad no lo encuenrro en ninguna parte de internet. Gracias de antemano y espero tu respuesta



admin

18 Oct 2015

Reply

Hola Timeo,

Animo con tu estudio que poco a poco las cosas van encajando. Para leer el teclado no vas a tener dificultad, y aqui tienes un tutorial decicado a esos pequeños de 4x4 que te muestra como leerlos: <http://www.prometec.net/teclados-matriciales/> Y una vez que lo tengas imprimirlo en el display te llevara poco. UN saludo y hasta pronto



Rafael Figueroa

11 Nov 2015

Reply

Muchas gracias amigo me sirvio de mucho esta explicación que has presentado



admin

11 Nov 2015

Reply

Gracias por tu amabilidad Rafa, para eso estamos

17 Nov 2015



Antonio

Bajé tu ejemplo y me da este error al compilar.
Gracias por todo

Reply

Arduino: 1.5.7 (Windows 7), Placa:"Arduino Uno"

Prog_19_1.ino: In function 'void setup()':
Prog_19_1.ino:26:6: error: redefinition of 'void setup()'
sketch_nov17b.ino:3:6: error: 'void setup()' previously defined here
Prog_19_1.ino:27:25: error: expected ';' before '}' token
Prog_19_1.ino: In function 'void loop()':
Prog_19_1.ino:29:6: error: redefinition of 'void loop()'
sketch_nov17b.ino:8:6: error: 'void loop()' previously defined here

This report would have more information with
"Show verbose output during compilation"
enabled in File > Preferences.



admin

17 Nov 2015

Hola Antonio,
Aparentemente el error que reportas se debe a una doble definición de la función setup. Parece que la has definido dos veces. asegurate de hacerlo una única vez

Reply



Antonio

18 Nov 2015

Efectivamente, corregido y funcionando.
Muchas gracias.

Reply



Jaume

11 Dic 2015

Gracias, admin, poco a poco voy entendiendo las endiabladas combinaciones de fórmulas que en sesiones anteriores no conseguía comprender. No me veo capaz de programar nada, pero lentamente voy entendiendo cada una de ellas y sus combinaciones, y ya no me espantan (tanto) los sketch.
Muchas gracias de nuevo.

Reply



admin

12 Dic 2015

Me alegra oírlo Jaume.

Reply

Irás viendo que si le das una oportunidad, la programación no es tampoco difícil. Es más bien como nadar o andar en bici, cuesta mucho al principio pero pronto se convierte en algo natural y casi te preguntas que es lo que veías de raro al principio



Carlos

04 Feb 2016

Hola, para poder escribir en word o en notepad con mi matricial desde arduino como le puedo hacer?

Reply



admin

04 Feb 2016

Hola Carlos, ¿Quieres decir usando un teclado hexadecimal?

Reply



Ismael

05 Feb 2016

Buenas tardes, tenía una serie de preguntas sobre el uso de este teclado matricial.

Reply

¿un teclado matricial 4x4 como el expuesto en esta página puede ser usado como un teclado 4x3? es decir, sin usar la última columna. En el caso de que así sea, tengo el código de mi proyecto diseñado para el uso de un teclado 4x3 pero a la hora de incorporar mi teclado no me reconoce los datos que estoy pulsando en mi teclado. Pulso la contraseña que en principio provocaría el movimiento de mi servomotor, pero este no hace nada y creo que es porque o bien mi teclado no reconoce las pulsaciones o porque mi código no está correctamente adaptado a este tipo de teclado.

Muchas gracias de antemano por la ayuda. Un saludo



admin

05 Feb 2016

Hola Ismael, estos teclados se forman por la intersección de los 4 hilos de la columna con los 4 hilos de las filas. Si no quieres usar la última columna basta con que no la conectes y ya tienes un teclado de 4x3, pero es un desperdicio

Reply Usalos y lee lo que te escriben en el teclado que siempre puedes usar algunas teclas como comandos concretos



Ismael

05 Feb 2016

He intentado configurarlo de nuevo pero no me funciona el apartado del código del teclado. ¿me podría ayudar usted?

Reply

Muchisimas gracias por la respuesta.



admin

05 Feb 2016

claro Ismael, postea a qui tu codigo y seguro que entre todos podemos darte alguna idea

Reply



Ismael

05 Feb 2016

El funcionamiento que quería configurar con mi código era que al cargarlo el servomotor se pusiera en 0° y el diodo led rojo se encendiera. Si meto con el teclado la contraseña predefinida:1234 este gira hasta los 90° y el led verde se enciende, una vez ahí si pulso # este vuelve a la posición original; mientras que si estando el servo en la posición de 90° pulso * puedo implementar una nueva contraseña. Por último, pretendía resetear la memoria EEPROM del Arduino para así poder recuperar mi contraseña. El código es el siguiente:

```
#include
#include
#include
Servo miservo;

int cont=0;
int pos=0;
char* secretCode = "1234";
char codigo[4];
int position = 0;
int fail = 0;
int intentos = 0;
boolean failure = false;
boolean locked = true;
int d = 100;

const byte rows = 4;
const byte cols = 4;
byte rowPins[rows] = {9,8,7,6};
byte colPins[cols] = {5,4,3,2};

char keys[rows][cols] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, rows, cols);

int redPin = 11;
int greenPin = 10;

void setup()
{
  Serial.begin(9600);
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  miservo.attach(A1);
  loadCode();
  flash();
  updateOutputs();
  updateOutputs2();
  updateOutputs3();
}

void loop()
{
  char key = keypad.getKey();
  Serial.println(key);

  if (key == '*' && locked==false)
```

```
{
// desbloqueo y pulsado *, así que cambia el código
position = 0;
getNewCode();
updateOutputs();
}

if (key == '#' && locked==false)
{
locked = true;
position = 0;
updateOutputs();
}

if (key != secretCode[position] && key != '#' && locked==true)
{
failure=true;
position ++;
}
if (key == secretCode[position])
{
position ++;
}
if (position == 4 && failure==false)
{
locked = false;
updateOutputs();
position=0;
}
if (position == 4 && failure==true)
{
position=0;
failure=false;
intentos ++;
updateOutputs2();
updateOutputs3();
}
delay(d);
}

void updateOutputs()
{
if (locked)
{
digitalWrite(redPin, HIGH);
digitalWrite(greenPin, LOW);
miservo.write(90);
delay(800);
}
else
{
digitalWrite(redPin, LOW);
digitalWrite(greenPin, HIGH);
miservo.write(0);
delay(800);
}
}

void updateOutputs2()
{
digitalWrite(redPin, HIGH);
digitalWrite(greenPin, HIGH);
delay(3000);
digitalWrite(redPin, HIGH);
digitalWrite(greenPin, LOW);
delay(200);
}

void updateOutputs3()
{
if (intentos == 3)
{
digitalWrite(redPin, HIGH);
digitalWrite(greenPin, HIGH);
}
```

```
delay(10000);
intentos=0;
}
else
{
digitalWrite(redPin, HIGH);
digitalWrite(greenPin, LOW);
delay(200);
}
}

void getNewCode()
{
flash();
for (int i = 0; i < 4; i++ )
{
char key;
key = keypad.getKey();
while (key == 0)
{
key = keypad.getKey();
}
flash();
secretCode[i] = key;
}
saveCode();
flash();flash();
}

void loadCode()
{
if (EEPROM.read(0) == 1)
{
secretCode[0] = EEPROM.read(1);
secretCode[1] = EEPROM.read(2);
secretCode [2] = EEPROM.read(3);
secretCode[3] = EEPROM.read(4);
}
}
void saveCode()
{
EEPROM.write(1, secretCode[0]);
EEPROM.write(2, secretCode[1]);
EEPROM.write(3, secretCode[2]);
EEPROM.write(4, secretCode[3]);
EEPROM.write(0, 1);
}
void flash()
{
digitalWrite(redPin, HIGH);
digitalWrite(greenPin, HIGH);
delay(500);
digitalWrite(redPin, LOW);
digitalWrite(greenPin, LOW);
}
```

Aunque logréis o no ayudarme, gracias por el interés mostrado.



FELIPE

20 Feb 2016

Gracias por tu valiosa información, muy bueno

Reply



Jose Ramirez

04 Mar 2016

hola quisiera que el teclado matricial de esta practica escribiera en word o excel o notepad como un teclado normal pero usar solo esas 16 teclas como se haria

Reply



admin

05 Mar 2016

Hola Jose, creo que necesitas que windows vea al arduino como un teclado y para eso necesitas creo recordar usar un modelo Leonardo por ejemplo que tiene esa capacidad pero que en cambio esta ausente en un UNO.

GIVE A REPLY

Conectado como Javier. ¿Quieres salir?

Message

Post comment

Comments Protected by WP-SpamShield for WordPress



Sí, agrégame a tu lista de correos.

CATEGORIAS DE LOS
PRODUCTOS

Selecciona una categoría ▼