

Libre Arduino

Documentación de Hardware Libre, por Abel Omar Serú

Teclado matricial analógico: 16 teclas, 1 pin

Descripción

En esta entrada se presentará cómo puede conectarse un teclado matricial 4x4 empleando un puñado de resistencias para distinguir qué botón se le ha pulsado mediante la simple lectura analógica de un solo pin.

La idea consiste en construir un divisor de tensión mediante el cual cada botón presionado entregue a la entrada analógica un único valor debido a la caída de tensión que ocasionan sus resistencias asociadas tanto en filas como columnas.

En sí será una actividad sencilla que podrá complejizarse cuanto uno quiera. Debo agradecer a RedPic de TodoPic.com.ar por la excelente información publicada en este [post](#). Gracias a su voluntad de compartirla hoy puedo estar presentando este conocimiento con aportes propios. El post también puede descargarse en PDF desde [aquí](#).

Si se desconoce qué es un teclado matricial y las formas de operarlo véase este [enlace](#). El modelo empleado para la siguiente actividad consiste en un ejemplar ya armado, flexible y muy económico, ver en [DealExtremme](#).

Usualmente un teclado matricial es consultado mediante un algoritmo en donde a muy altas velocidades se ponen en alto la columnas y se consulta el estado de las filas, por lo que un teclado matricial conectado a la placa en forma estándar consumirá 8 pines y ocupará unas cuantas líneas de programa para el algoritmo de detección ante la pulsación de teclas. Esta técnica en verdad no tiene comparación con algo tan sencillo como saber que el botón pulsado puede consultarse con una sola línea de programa y algunos If para asignar el dato definitivo al valor analógico leído. Realmente es una gran ventaja emplear esta técnica de lectura mediante un solo pin, se reduce el hardware, el código, se simplifica el *debouncing*, entre muchas ventajas más.

Actividad

La actividad que se presentará será muy simple ya que la prioridad será que el lector aprenda a conectar el teclado junto con las resistencias y a generar la tabla de valores para cada botón dado que las resistencias varían dentro de su tolerancia. Esta tabla de valores podrá ser elaborada mediante un simple sketch donde se comunicará a la pc el valor analógico leído ante la pulsación de cualquier tecla.

Una vez creada la tabla de valores, con una sencilla serie de condicionales If podrá asignarse a cada valor analógico (o rango de valores) un número del 0 al 9, de la A a la D o numeral / asterisco.

Se requerirá de una breadboard para el armado del circuito, 8 cables para filas y columnas del teclado, tres cables a la placa (5V, GND y señal) y 8 resistencias.

PÁGINA PRINCIPAL

[\[Clic para ir \]](#)

[!] PRÓXIMAMENTE

- Interfaz de usuario, parte II: botón y potenciómetro para entrada de datos.
- *Mejora en comentarios de programas.*
- *Imágenes ilustrativas.*

ÍNDICE DE ENTRADAS

A. Introducción

1. Descripción del blog, alcance y perspectivas - **Nuevo!**
2. ¡Bienvenidos! un breve mensaje de presentación - **Nuevo!**

B. Conceptos Generales

1. 123D Circuits: simulador de Arduino online
2. Arduino, videotutoriales - **Nuevo!**
3. Guía básica de programación
4. Primeros pasos con la placa Arduino - **Nuevo!**
5. Textos, manuales y tutoriales de utilidad. Sitio oficial Arduino - **Nuevo!**
6. ¿Qué es Arduino? - **Nuevo!**

C. Interacción con Periféricos

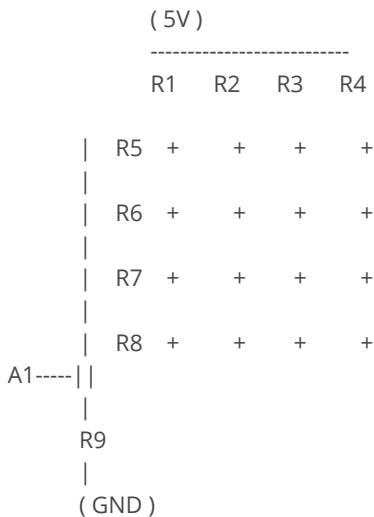
1. Display LCD 16x2, demostrativo simple - **Nuevo!**
2. Entradas analógicas y salidas PWM
3. Interfaz de usuario, parte I: LCD - **Nuevo!**
4. LCD Keypad Shield
5. Módulo RTC DS1302. Reloj de tiempo real - **Nuevo!**
6. Reloj de tiempo real por software con almacenamiento en EEPROM interna - **Nuevo!**
7. Sensor de distancia por ultrasonido Ping
8. Sensores de temperatura y humedad DHT11/DHT22, versión 3 pin.

Placa: Arduino Duemilanove, IDE V1.0.5

[Imagen del teclado matricial]

Conexión del circuito

Mapa del teclado matricial (frente)



R1 = 0 Ohm; R2 = 470 Ohm; R3 = 1K Ohm; R4 = 2,2K Ohm; R5 = 220 Ohm; R6 = 330 Ohm; R7 = 470 Ohm; R8 = 560 Ohm; R9 = 1,2K Ohm.

Nota: Se recomienda emplear resistencias de tolerancia 1%. En caso de que no se pueda acceder a estas, la tolerancia general del 5% funcionará correctamente.

Detalle: Todas las columnas deberán ser conectadas a 5V mediante sus respectivas resistencias y todas las filas deberán ser conectadas a A1 mediante sus respectivas resistencias. Desde A1 saldrá, en paralelo, una conexión a GND mediante R9.

"A1" - Pin "A1" del Arduino

PINES. Filas y columnas, vista frontal del conector



Adquiriendo datos del teclado por primera vez

En esta técnica es fundamental aclarar que dos teclados distintos no se comportarán de la misma forma debido a la tolerancia de las resistencias principalmente y a las distintas variaciones en caída de tensión que ocasionarán las resistencias de un teclado en comparación con las de otro. Para evitar esto, luego de haber ensamblado el teclado matricial con las resistencias a su alrededor debe realizarse una primera prueba con un sketch muy simple.

Esta actividad inicial se realizará una sola vez a menos que el teclado sea desmontado y reemplazadas sus resistencias, pero si la intención es armarlo en PCB definitivamente, esta prueba deberá ejecutarse sólo una vez.

El procedimiento es el siguiente:

- Teclado matricial analógico: 16 teclas, 1 pin

D. Modelos para Construcción

- Nanino, excelente propuesta para diseños finales
- Paperduino, por Iñigo Zuluaga Uribasterra de Txapuzas Electrónicas
- Placas Arduino de hardware libre para construcción

I. Recursos y aplicaciones

- 123D Circuits: simulador de Arduino online
- Recurso para docentes: planilla multiproósito en formato PDF para notas/asistencia

DESCARGA INSTANTÁNEA DE LA PÁGINA ACTUAL EN PDF

[Clic aquí]

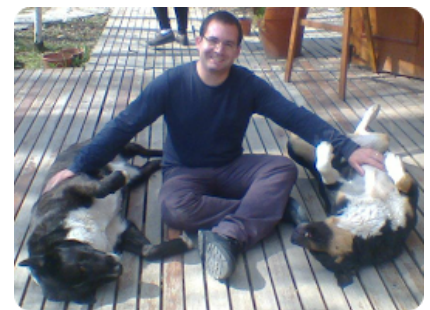
LICENCIA DE CULTURA LIBRE



Blog Libre Arduino por Abel Omar Serú se encuentra bajo una [Licencia Creative Commons Atribución-CompartirIgual 3.0 Unported](#).

Basada en una obra en www.arduino.cc. Permisos que vayan más allá de lo cubierto por esta licencia pueden encontrarse en aseru1988@gmail.com.

AUTOR



Abel Omar Serú. aseru1988@gmail.com

Recomendar esto en Google

ENLACES DE UTILIDAD

[Sitio Oficial Arduino](#)

[Txapuzas Electrónicas](#)

[Arduiteka](#)

[Ucontrol](#)

[Pablín Circuitos Electrónicos](#)

[Proyectos Make](#)

1. Conectar el teclado matricial junto con las resistencias como se detalló anteriormente.
2. Conectar la placa a la PC, próximamente se establecerá una comunicación.
3. Cargar a la placa el siguiente [sketch](#).
4. Abrir el monitor serial del IDE, especificar 9.600 Baudios.
5. Presionar de a uno por vez cada botón del teclado, recordar el orden (se aconseja seguir un orden "tipo lectura" Izq.-Der. / Arriba-Abajo)
6. Copiar a un block de notas la columna generada por el monitor serial donde se encontrarán todos los valores analógicos uno debajo del otro. [Ver resultado](#).
7. Recordando el orden con que se presionó cada botón, asignar, dentro del block de notas creado, a cada valor analógico su respectivo botón. [Ver resultado](#).
8. Habiendo hecho esto, aconsejo asignar a cada botón (es decir, a cada valor analógico) un número con el cual podrá cargarse una variable de programa cada vez que se pulsa un botón. Esto se hace porque si bien es sencillo hacer "boton = 0" ó "boton = 8" no será posible hacer "boton = A" ó "boton = #". Para esto se asigna un número con que muy fácilmente podrá saberse dentro del programa qué botón se presionó.
9. Con esta idea en mente deberá escribirse (a modo de guía, esto no es rigurosamente necesario para programar) una tabla como la siguiente. Obsérvese que las columnas está ordenadas de la siguiente forma:
 1. El valor analógico leído para cada botón.
 2. El botón del teclado que ocasionó la lectura de ese valor.
 3. El número que será cargado a la variable de programa "boton" para proceder a la toma de decisiones dentro del programa.

Tabla a continuación: *Valores analógicos detectados en A1 según tecla y número (#) de su variable de programa. Este procedimiento es muy similar al realizado con el control remoto sólo que esta vez es necesario "calibrar" nuestro teclado y acondicionarlo al sketch para que opere correctamente.*

| 1 | A1 (valor) | TECLA | # |
|----|------------|-------|----|
| 2 | ----- | | |
| 3 | 508 | 0 | 30 |
| 4 | 853 | 1 | 31 |
| 5 | 591 | 2 | 32 |
| 6 | 502 | 3 | 33 |
| 7 | 794 | 4 | 34 |
| 8 | 562 | 5 | 35 |
| 9 | 481 | 6 | 36 |
| 10 | 729 | 7 | 37 |
| 11 | 528 | 8 | 38 |
| 12 | 456 | 9 | 39 |
| 13 | 691 | * | 41 |
| 14 | 441 | # | 42 |
| 15 | 331 | A | 51 |
| 16 | 322 | B | 52 |
| 17 | 310 | C | 53 |
| 18 | 303 | D | 54 |

gistfile1.txt hosted with ♥ by GitHub

view raw

Nota: *El siguiente [Sketch](#) muestra cómo esta tabla es implementada a la práctica y de qué forma se establece una tolerancia en la detección de los botones.*

IMPORTANTE (1): Cada vez que el circuito sea montado/desmontado, cambio de cables, breadboard, etc. es posible que los valores volcados en la tabla cambien levemente, en caso de que el circuito sea armado en forma definitiva esto no ocurrirá.

Electrónica Fácil

Creative Commons Argentina

Ebooks Académicos de Electrónica

Biblioteca "Free Computer Books"

Buscar PDFs en la web

LIBRE ARDUINO



ARGENTINA



VISITAS


13,377

COMPARTIR

 Share

TRANSLATE

Seleccionar idioma ▼

Con la tecnología de  **Traductor**

IMPORTANTE (2): El valor de tolerancia aplicado a la detección de los botones pulsados no deberá ser muy grande (no mayor a 2) dado que existirá un solapamiento entre los valores tolerados y se producirán falsas detecciones como, por ejemplo, el botón 3 en lugar del 9.

Habiendo asignado a cada valor analógico su respectivo número dentro del programa, y habiendo respetado cada número para cada botón como lo indica la tabla anterior, podrá darse ejecución al sketch de prueba. En donde se realizará lo mismo que se vió en la entrada "Mando por control remoto infrarrojo" sólo que esta vez será mediante el teclado matricial. En entradas más avanzadas se verá cómo emplear el teclado para componer una clave, almacenarla en la EEPROM y verificarla cada vez que sea ingresada. Esta sería la principal función de un teclado matricial, el ingreso de datos. Aunque siempre es necesario cuando se empieza, iniciarse desde lo más básico. Haber hecho funcionar el teclado matricial al menos de forma elemental mediante un solo pin de entrada ya es, de por sí, un gran logro. A seguir avanzando!

[\[Más información sobre teclados matriciales \]](#)

[\[Adquirir producto online \]](#)

[\[Archivo .zip: modelo alternativo para construir \]](#)

[\[Enlace al artículo \]](#)

Por Txapuzas.blogspot.com

Sketch de prueba, base

```
1 // El siguiente sketch es de una mayor complejidad.
2 // Considere un desafío su interpretación ya que agregar comentarios
3 // haría que el programa quede muy sobrecargado de informacion.
4
5 int entrada_analogica_teclado = A1;
6 int valor_analogico_teclado = 0;
7
8 void setup(){
9   Serial.begin(9600);
10 }
11
12 void loop(){
13   if(analogRead(entrada_analogica_teclado)>10){
14     valor_analogico_teclado = analogRead(entrada_analogica_teclado);
15     Serial.println(valor_analogico_teclado);
16     while(analogRead(entrada_analogica_teclado)>10) delay(1);
17   }
18 }
```

gistfile1.ino hosted with ♥ by GitHub

[view raw](#)

Sketch de prueba, base 2 (mayor complejidad)

```
1 // El siguiente sketch es de una mayor complejidad.
2 // Considere un desafío su interpretación ya que agregar comentarios
3 // haría que el programa quede muy sobrecargado de informacion.
4
5 int entrada_analogica_teclado = A1;
6 int valor_analogico_teclado = 0;
7 int boton = 0;
8
9 void setup(){
10   Serial.begin(9600);
11 }
12
13 void loop(){
14   if(analogRead(entrada_analogica_teclado)>10){
15     valor_analogico_teclado = analogRead(entrada_analogica_teclado);
16     while(analogRead(entrada_analogica_teclado)>10) delay(1);
17     int tolerancia = 2;
18     if (valor_analogico_teclado<=(508+tolerancia) && valor_analogico_teclado>=(508-
19     if (valor_analogico_teclado<=(853+tolerancia) && valor_analogico_teclado>=(853-
```

```

20   if (valor_analogico_teclado<=(591+tolerancia) && valor_analogico_teclado>=(591.
21   if (valor_analogico_teclado<=(502+tolerancia) && valor_analogico_teclado>=(502.
22   if (valor_analogico_teclado<=(794+tolerancia) && valor_analogico_teclado>=(794.
23   if (valor_analogico_teclado<=(562+tolerancia) && valor_analogico_teclado>=(562.
24   if (valor_analogico_teclado<=(481+tolerancia) && valor_analogico_teclado>=(481.
25   if (valor_analogico_teclado<=(729+tolerancia) && valor_analogico_teclado>=(729.
26   if (valor_analogico_teclado<=(528+tolerancia) && valor_analogico_teclado>=(528.
27   if (valor_analogico_teclado<=(456+tolerancia) && valor_analogico_teclado>=(456.
28   if (valor_analogico_teclado<=(691+tolerancia) && valor_analogico_teclado>=(691.
29   if (valor_analogico_teclado<=(441+tolerancia) && valor_analogico_teclado>=(441.
30   if (valor_analogico_teclado<=(331+tolerancia) && valor_analogico_teclado>=(331.
31   if (valor_analogico_teclado<=(322+tolerancia) && valor_analogico_teclado>=(322.
32   if (valor_analogico_teclado<=(310+tolerancia) && valor_analogico_teclado>=(310.
33   if (valor_analogico_teclado<=(303+tolerancia) && valor_analogico_teclado>=(303.
34   Serial.println(valor_analogico_teclado);
35   Serial.println(boton);
36   Serial.println("");
37 }
38 }

```

gistfile1.ino hosted with ♥ by GitHub

[view raw](#)

Sketch de prueba

```

1 // El siguiente sketch es de una mayor complejidad.
2 // Considere un desafío su interpretación ya que agregar comentarios
3 // haría que el programa quede muy sobrecargado de informacion.
4
5 #include <LiquidCrystal.h>
6 LiquidCrystal lcd(8, 9, 4, 5, 6, 7);
7
8 int modo_lcd = 1;
9 int boton = 0;
10 int valor_pwm_led = 0;
11 int led = 3;
12 int i = 0;
13 int entrada_analogica_teclado = A1;
14 int valor_analogico_teclado = 0;
15
16 // Barra
17 byte newChar3[8] = {
18   B01001,
19   B10010,
20   B00100,
21   B01001,
22   B10010,
23   B00100,
24   B01001,
25   B10010
26 };
27
28 void setup(){
29   pinMode(led, OUTPUT);
30   lcd.createChar(3, newChar3);
31   lcd.begin(16, 2);
32 }
33
34 void loop()
35 {
36   if(analogRead(entrada_analogica_teclado)>10){ /// Inicia bloque de recepción
37     valor_analogico_teclado = analogRead(entrada_analogica_teclado);
38     while(analogRead(entrada_analogica_teclado)>10) delay(1);
39     int tolerancia = 3;
40     if (valor_analogico_teclado<=(508+tolerancia) && valor_analogico_teclado>=(508.
41     if (valor_analogico_teclado<=(853+tolerancia) && valor_analogico_teclado>=(853.
42     if (valor_analogico_teclado<=(591+tolerancia) && valor_analogico_teclado>=(591.
43     if (valor_analogico_teclado<=(502+tolerancia) && valor_analogico_teclado>=(502.
44     if (valor_analogico_teclado<=(794+tolerancia) && valor_analogico_teclado>=(794.
45     if (valor_analogico_teclado<=(562+tolerancia) && valor_analogico_teclado>=(562.
46     if (valor_analogico_teclado<=(481+tolerancia) && valor_analogico_teclado>=(481.
47     if (valor_analogico_teclado<=(729+tolerancia) && valor_analogico_teclado>=(729.
48     if (valor_analogico_teclado<=(528+tolerancia) && valor_analogico_teclado>=(528.
49     if (valor_analogico_teclado<=(456+tolerancia) && valor_analogico_teclado>=(456.
50     if (valor_analogico_teclado<=(691+tolerancia) && valor_analogico_teclado>=(691.
51     if (valor_analogico_teclado<=(441+tolerancia) && valor_analogico_teclado>=(441.

```

```

52  if (valor_analogico_teclado<=(331+tolerancia) && valor_analogico_teclado>=(331-
53  if (valor_analogico_teclado<=(322+tolerancia) && valor_analogico_teclado>=(322-
54  if (valor_analogico_teclado<=(310+tolerancia) && valor_analogico_teclado>=(310-
55  if (valor_analogico_teclado<=(303+tolerancia) && valor_analogico_teclado>=(303-
56
57  /// CASO TECLA A ///
58  if (boton == 51) {
59      lcd.clear();
60      while (true) {
61          if (modo_lcd == 1) {
62              modo_lcd = 2;
63              break;
64          }
65          if (modo_lcd == 2) {
66              modo_lcd = 1;
67              break;
68          }
69      }
70  }
71
72  /// CASO TECLAS C D ///
73  if (modo_lcd == 2) {
74      if (valor_pwm_led <= 850) {
75          if (boton == 53) valor_pwm_led = valor_pwm_led + 50;
76      }
77      if (valor_pwm_led >= 50) {
78          if (boton == 54) valor_pwm_led = valor_pwm_led - 50;
79      }
80  }
81
82  /// CASO TECLAS 0-9 ///
83  if (modo_lcd == 1) {
84      if (boton >= 30 && boton <=39) {
85          valor_pwm_led = map(boton, 30, 39, 0, 900);
86      }
87  }
88  } /// Finaliza bloque de recepción
89
90  if (modo_lcd == 1) {
91      lcd.setCursor(0,0);
92      lcd.print("SET 0-9:");
93      if (boton >= 30 && boton <= 39) lcd.print(valor_pwm_led);
94      lcd.print(" ");
95
96      const int periodo_min = 0;
97      const int periodo_max = 900;
98      int periodo_lcd_bar = map(valor_pwm_led, periodo_min, periodo_max, 0, 16);
99      int resta_periodo_lcd_bar = 16 - periodo_lcd_bar;
100
101      for (int i = 0; i < periodo_lcd_bar; i++) {
102          lcd.setCursor(i,1);
103          lcd.write(3);
104      }
105      for (int i = 0; i < resta_periodo_lcd_bar; i++) {
106          lcd.setCursor(periodo_lcd_bar,1);
107          if (periodo_lcd_bar == 0) {
108              lcd.setCursor(0,1);
109          }
110          lcd.print(" ");
111      }
112  }
113
114  if (modo_lcd == 2) {
115      lcd.setCursor(0,0);
116      lcd.print("SET +/-: ");
117      lcd.print(valor_pwm_led);
118      lcd.print(" ");
119
120      const int periodo_min = 0;
121      const int periodo_max = 900;
122      int periodo_lcd_bar = map(valor_pwm_led, periodo_min, periodo_max, 0, 16);
123      int resta_periodo_lcd_bar = 16 - periodo_lcd_bar;
124
125      for (int i = 0; i < periodo_lcd_bar; i++) {
126          lcd.setCursor(i,1);
127          lcd.write(3);
128      }
129      for (int i = 0; i < resta_periodo_lcd_bar; i++) {
130          lcd.setCursor(periodo_lcd_bar,1);

```

```

131     if (periodo_lcd_bar == 0) {
132         lcd.setCursor(0,1);
133     }
134     lcd.print("          ");
135 }
136 }
137
138 analogWrite(led,(map(valor_pwm_led,0,900,0,255)));
139 }

```

gistfile1.ino hosted with ♥ by GitHub

[view raw](#)Publicadas por [Libre Arduino](#) a la/s 12:00 a. m.[Recomendar esto en Google](#)Etiquetas: [C. Interacción con Periféricos](#)

3 comentarios:

**Anónimo** 10 de abril de 2014, 2:45 p. m.

no me despliega los valores en el puerto serial pero no se porque si se supone que conecté todo de manera correcta

[Responder](#)

Respuestas

**Libre Arduino** 11 de abril de 2014, 9:34 p. m.

Recordá que este tutorial está probado por mí únicamente para el modelo de teclado-membrana que se publica en los enlaces, muy posiblemente también trabaje con otro teclado 4x4 similar o hecho a mano. El tutorial recomendado de Txapuzas y su modelo de teclado presenta sus propios sketches, a no confundir. Otro punto: fijate que la entrada analógica realmente sea por A1, respetá los valores de las resistencias.. Comenzá por el primer sketch de todos, ese siempre algún dato va a publicar siempre y cuando le llegue tensión a A1 (medí con tester, revisá que el monitor serial esté a 9600 baudios...) Creería que con esta información podés solucionarlo.

**Libre Arduino** 11 de abril de 2014, 9:50 p. m.

Otro aporte: (ver "Detalle" en el esquema) Todas las columnas deberán ser conectadas a 5V mediante sus respectivas resistencias y todas las filas deberán ser conectadas a A1 mediante sus respectivas resistencias. Desde A1 saldrá, en paralelo, una conexión a GND mediante R9.

Responder

Escribe un comentario...

Comentar como: Cuenta de Gox ▼[Publicar](#)[Vista previa](#)

[Entrada más reciente](#)

[Página Principal](#)

[Entrada antigua](#)

Suscribirse a: [Comentarios de la entrada \(Atom\)](#)

Plantilla Picture Window. Tecnología de [Blogger](#).