

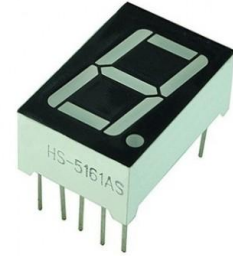


# VISUALINO

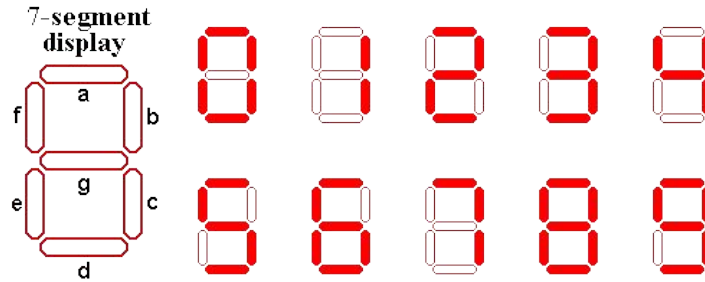
APUNTES Y PROYECTOS V

por Aurelio Gallardo Rodríguez BY – SA – NC

# DISPLAYS 7 SEGMENTOS



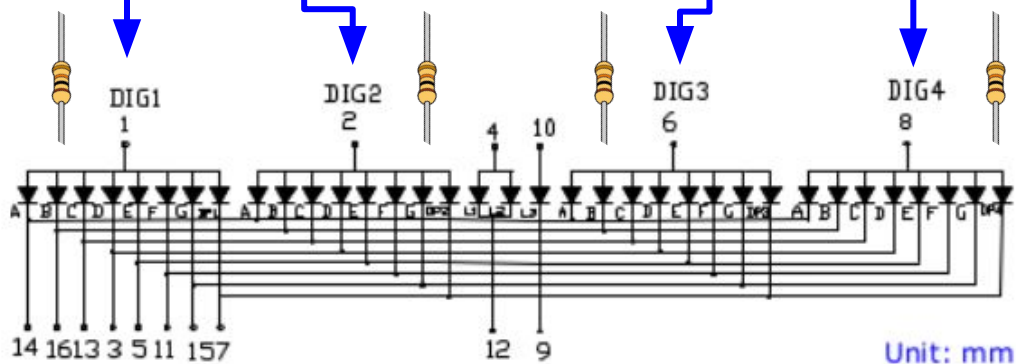
# CONEXIONADO DE UN DISPLAY DE 7 SEGMENTOS. ACTIVADO.



A cada número corresponde un conexionado distinto de 7 leds. Por ejemplo, al “1” le corresponde activarse el b y c, mientras que el “8” los tiene todos conectados.

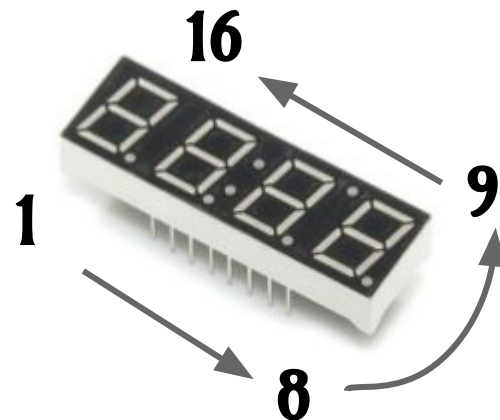
# USARÉ UN DISPLAY CUÁDRUPLE, EL YSD-439AY2B-35

Ánodos que activan cada display 5V.  
Resistencias  $\sim 470\Omega$



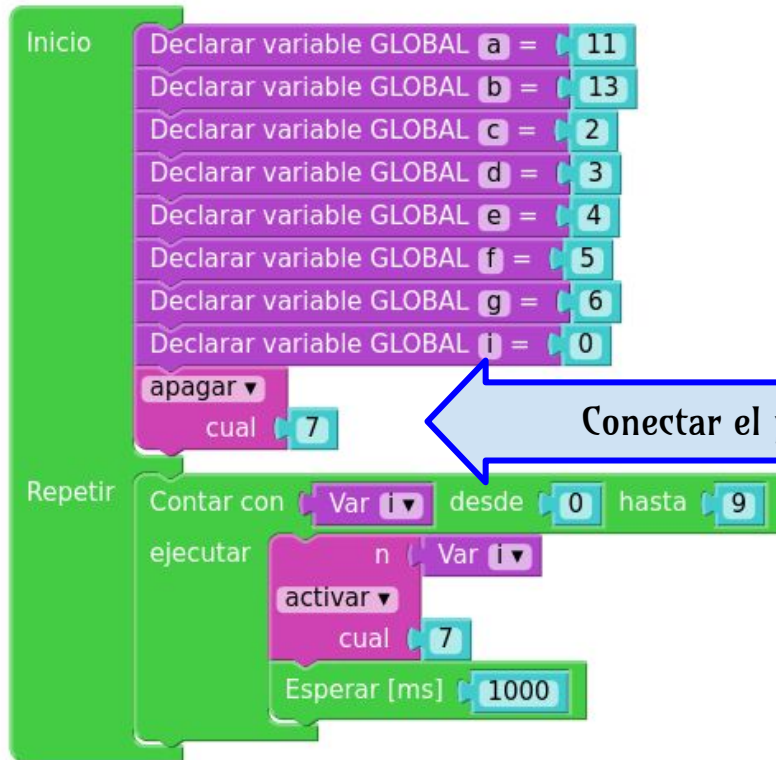
Cátodos comunes.  
Activar a 0V

Separador minutos-  
segundos y punto  
decimal



Importante: cada display se activa escribiendo un “alto” en su ánodo y un “bajo” en cada segmento a,b,c,d,e,f

# CONTANDO DEL 0 AL 9 (PROGRAMA PRINCIPAL)



Conecto cada segmento a un pin de ARDUINO

Conectar el pin 8 del display al 7 de ARDUINO. Función de apagado

Bucleo principal. Cuenta del 0 al 9. Pasa el número a una función de activación e indica qué ánodo activar. Se espera un poco.

# CONTANDO DEL 0 AL 9 (ACTIVAR)

```
★ activar int n, int cual
ejecutar ★ si Var n caso 0
ejecutar cero cual Var cual
caso 1
ejecutar uno cual Var cual
caso 2
ejecutar dos cual Var cual
caso 3
ejecutar tres cual Var cual
caso 4
ejecutar cuatro cual Var cual
caso 5
ejecutar cinco cual Var cual
```

Se escoge un bucle de control “switch case”; en función del número n, activará una función, una por cada número. Va transmitiendo el ánodo a activar “cual”.

```
caso 6
ejecutar seis cual Var cual
caso 7
ejecutar siete cual Var cual
caso 8
ejecutar ocho cual Var cual
caso 9
ejecutar nueve cual Var cual
```

# CONTANDO DEL 0 AL 9 (Nº 8 - TODOS- Y APAGAR)


```
★ apagar int cual
ejecutar
Escribir en el pin digital PIN# 3 estado ALTO ▼
Escribir en el pin digital PIN# 2 estado ALTO ▼
Escribir en el pin digital PIN# 4 estado ALTO ▼
Escribir en el pin digital PIN# 5 estado ALTO ▼
Escribir en el pin digital PIN# 6 estado ALTO ▼
Escribir en el pin digital PIN# 11 estado ALTO ▼
Escribir en el pin digital PIN# 13 estado ALTO ▼
Escribir en el pin digital PIN# Var cual estado ALTO ▼
```

En la función apagar, el ánodo y todos los pines se ponen en ALTO (daría igual ponerlos en BAJO). Al no haber **diferencia de tensión** entre ánodo y cátodo, no circula intensidad y no se encienden.

En la función ocho, TODOS los cátodos están BAJO y el ánodo en ALTO, luego circula la corriente por todas las barras del LED, iluminándolas todas

```
★ ocho int cual
ejecutar
Escribir en el pin digital PIN# Var a estado BAJO ▼
Escribir en el pin digital PIN# Var b estado BAJO ▼
Escribir en el pin digital PIN# Var c estado BAJO ▼
Escribir en el pin digital PIN# Var d estado BAJO ▼
Escribir en el pin digital PIN# Var e estado BAJO ▼
Escribir en el pin digital PIN# Var f estado BAJO ▼
Escribir en el pin digital PIN# Var g estado BAJO ▼
Escribir en el pin digital PIN# Var cual estado ALTO ▼
```

# CONTANDO DEL 0 AL 9 (ALGUNOS NÚMEROS)



```
función uno int cual
ejecutar
  cual ← Var cual
  apagar
  Escribir en el pin digital PIN# Var cual estado ALTO
  Escribir en el pin digital PIN# Var b estado BAJO
  Escribir en el pin digital PIN# Var c estado BAJO
```

En el uno, por ejemplo, primero se apagan todos y después se activan los que hace falta. El 7 es muy parecido, aparte de activar b y c, como en el uno, se activa el a

En la función cero, se aprovecha que casi todos están encendidos. Así pues se activa el 8 (todos) y se “apaga” poniendo en ALTO el sector g. También así, de forma parecida, se pueden hacer 2, 3, 4, 5, 6 y 9



```
función cero int cual
ejecutar
  cual ← Var cual
  apagar
  cual ← Var cual
  ocho
  Escribir en el pin digital PIN# Var g estado ALTO
```



# PROYECTOS CON DISPLAYS 7 SEGMENTOS

- ❑ **Proyecto 1 (fácil):** hacer un contador con un potenciómetro (o un sensor de luz o similar) que cuente del 0 al 9 según la entrada de la señal analógica.
- ❑ **Proyecto 2 (difícil):** hacer también un contador, pero esta vez del cero al 99. Es más difícil porque implica dos ánodos (pines 6 y 8 del display - 7 y 8 de ARDUINO, por ejemplo) que hay que SINCRONIZAR, ya que sólo se puede activar uno a la vez.
- ❑ **Proyecto 3 (basado en el 2):** realizar un contador de números de espera. Es normal ver en la carnicería, en la farmacia o en otros establecimientos que hay unos números de espera. Se recoge el ticket y el dependiente, pulsando un botón, hace avanzar un número del 0 al 99. Cuando acaba el 99 se empieza de nuevo.
- ❑ **Proyecto 4 (muy, muy difícil) reloj:** implica sincronizar todos los 4 dígitos del display, sincronizarlos, contar las horas y los minutos (¡no se pueden usar delays!), etc.
- ❑ Otros: termómetro, voltímetro, etc.

# CONTANDO DEL 0 AL 99 (POTENCIÓMETRO)

```
Inicio
Declarar variable GLOBAL a = 11
Declarar variable GLOBAL b = 13
Declarar variable GLOBAL c = 2
Declarar variable GLOBAL d = 3
Declarar variable GLOBAL e = 4
Declarar variable GLOBAL f = 5
Declarar variable GLOBAL g = 6
Declarar variable GLOBAL i = 0
Declarar variable GLOBAL valor = 0
Declarar variable GLOBAL unidades = 0
Declarar variable GLOBAL decenas = 0
Declarar variable GLOBAL tiempoSincronizado = 5
apagarTodos
```

Definir variables valor, unidades, decenas, tiempo de sincronizado (comprobar pero óptimo 5ms)

```
★ apagarTodos
ejecutar
  cual 7
  apagar
  cual 8
  apagar
```

Función que los apaga todos

```
★ uno int cual
ejecutar
  Escribir en el pin digital PIN# Var a estado ALTO
  Escribir en el pin digital PIN# Var b estado BAJO
  Escribir en el pin digital PIN# Var c estado BAJO
  Escribir en el pin digital PIN# Var d estado ALTO
  Escribir en el pin digital PIN# Var e estado ALTO
  Escribir en el pin digital PIN# Var f estado ALTO
  Escribir en el pin digital PIN# Var g estado ALTO
  Escribir en el pin digital PIN# Var cual estado ALTO
```

```
★ siete int cual
ejecutar
  cual Var cual
  uno
  Escribir en el pin digital PIN# Var a estado BAJO
```

Redefinir UNO y SIETE

# CONTANDO DEL 0 AL 99 (POTENCIÓMETRO)

```
Repetir
  Var valor = Potenciómetro PIN# Pin analógico A0
  Var valor = Mapear Var valor De [ 0 - 1023 ] a [ 0 - 99 ]
  Var unidades = Var valor % 10
  Var decenas = Var valor ÷ 10
  apagarTodos
  Esperar [ms] Var tiempoSincronizado
  n Var decenas
  cual 8
  activar
  Esperar [ms] Var tiempoSincronizado
  apagarTodos
  Esperar [ms] Var tiempoSincronizado
  n Var unidades
  cual 7
  activar
  Esperar [ms] Var tiempoSincronizado
```

Obtención y "mapeo"

Unidades: módulo de la resta entre diez  
Decenas: dividir entre 10 (¿parte entera?)

Apago, espero, enciendo las decenas, espero,  
apago, enciendo las unidades, espero

# LCDs

(16x2)



# CONEXIONADO

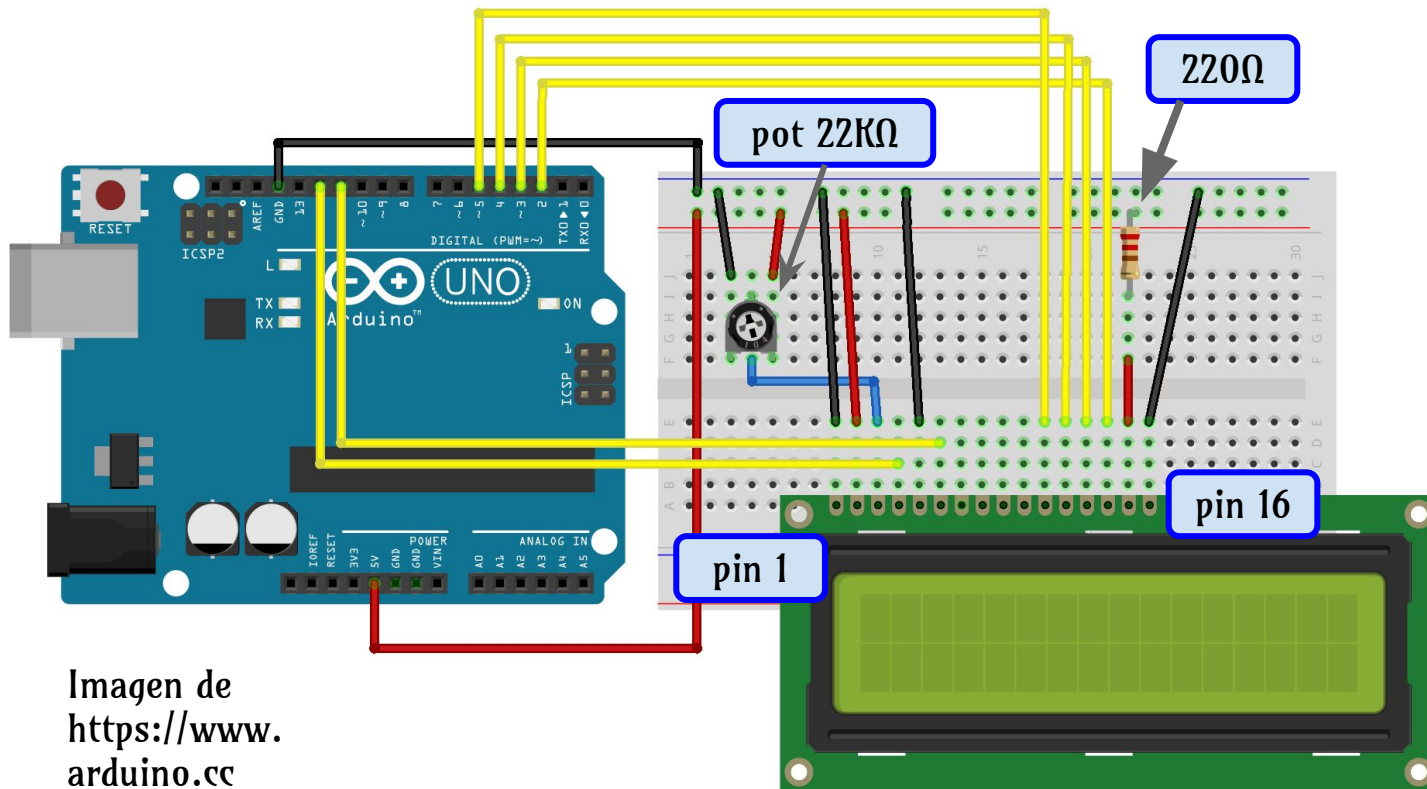


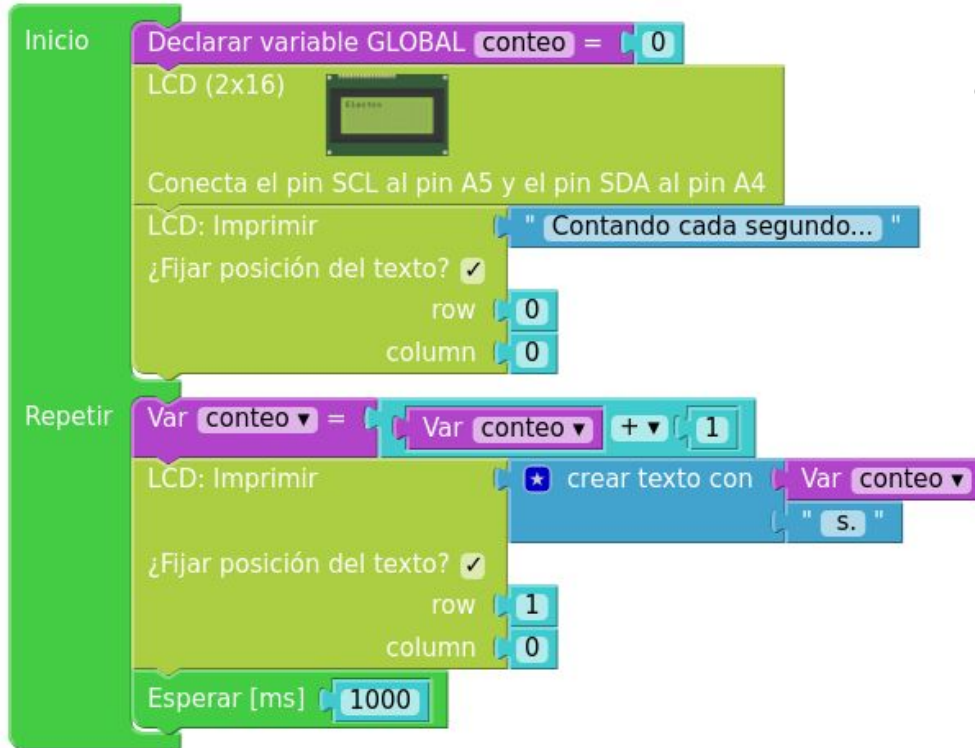
Imagen de  
<https://www.arduino.cc>

LCD	Arduino
RS (4)	12
Enable (6)	11
LCD D4	5
LCD D5	4
LCD D6	3
LCD D7	2

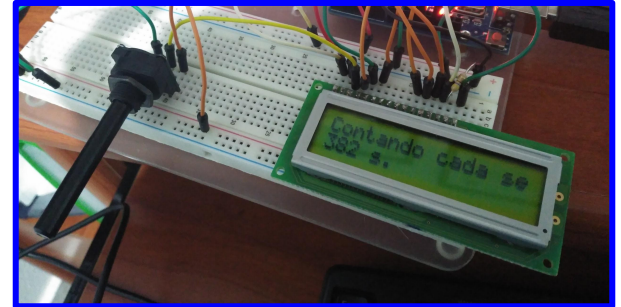
Aurelio Gallardo Rodríguez



# EJEMPLO: CUENTA DE SEGUNDOS.



- El bloque AJUSTAR RETROILUMINACIÓN está obsoleto. No funcionará.
- Los desarrolladores de VISUALINO comentan que este bloque tendrá en un futuro la posibilidad de cambiar los pines de conexión. Por ahora, sólo acepta la combinación para la librería LiquidCrystal 12, 11, 5, 4, 3 y 2.



# LCD KEYPAD SHIELD





# CONEXIÓN.

Conectar la placa es fácil. Sólo hay que alinear los pines de nuestro arduino y los de la placa, y encajarlos. Los pines del LCD y de ARDUINO se corresponden con los de la tabla:

Pin	Function
Analog 0	Button (select, up, right, down and left)
Digital 4	DB4
Digital 5	DB5
Digital 6	DB6
Digital 7	DB7
Digital 8	RS (Data or Signal Display Selection)
Digital 9	Enable
Digital 10	Backlit Control

Tenemos aún los pines del D0 al D7 libres, y los analógicos del A1 al A5. La entrada A0 analógica controla los botones: funciones UP, DOWN, LEFT, RIGHT y SELECT. Se recomienda usar un pequeño programa para averiguar los valores correctos de cada botón ANTES de empezar a trabajar, por si los valores varían de modelo en modelo.



# TRUCO PARA USAR LA LCD KEYPAD SHIELD CON VISUALINO.

Aunque sé de buena tinta que los desarrolladores de VISUALINO están trabajando para mejorar y, entre otras cosas, poder optar por los pines del conexionado del LCD, por ahora, la traducción que hace VISUALINO al código ARDUINO, al usar la librería LiquidCrystal es:

```
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
```

Para poder usar VISUALINO con nuestro KEYPAD SHIELD LCD deberemos:

- Archivo >> Exportar como. Y guardarlo como un archivo .ino de ARDUINO.
- Abrir con ARDUINO y cambiar la línea anterior por `LiquidCrystal lcd(8, 9, 4, 5, 6, 7)`
- Cargar el código desde el IDE de ARDUINO.

# PROGRAMA QUE RECONOCE BOTONES Y CUENTA EL TIEMPO.

```
Inicio
  Declarar variable GLOBAL contar = 0
  Declarar variable GLOBAL boton = 0
  LCD (2x16)
  Conecta el pin SCL al pin A5 y el pin SDA al pin A4
  LCD: Imprimir " Pula un boton "
  ¿Fijar posición del texto?
  row 0
  column 0

Repetir
  Var boton = Leer el pin analógico PIN# 0
  Var contar = Var contar + 1
  LCD: Imprimir crear texto con botonPulsado valor Var boton
  " temp: "
  Var contar
  " "
  ¿Fijar posición del texto?
  row 1
  column 0
  Esperar [ms] 1000
```

```
* botonPulsado int valor
ejecutar Declarar variable devolver = " - "
* si Var valor es: 0
ejecutar Var devolver = " RIGHT "
caso 254
ejecutar Var devolver = " DOWN "
caso 97
ejecutar Var devolver = " UP "
caso 407
ejecutar Var devolver = " LEFT "
caso 639
ejecutar Var devolver = " SELECT "
devuelve Var devolver
```