

DISPLAY DE 7 SEGMENTOS

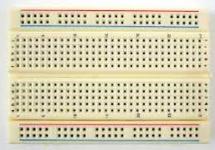
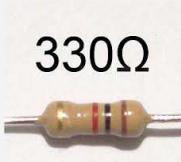
Versión: (20-9-18)

Mostrando informacion numérica

OBJETIVOS

Manejar nuestro primer display numérico de 1 dígito.

MATERIAL REQUERIDO.

	<p>Arduino Uno o similar.</p> <p><i>Esta sesión acepta cualquier otro modelo de Arduino.</i></p>
	<p>Un Protoboard.</p>
	<p>Cables de protoboard</p>
	<p>Una resistencia. (podria tener un valor menor)</p>
	<p>Un display de 1 dígito.</p>

LOS DISPLAYS LED

En cuanto empezéis a plantearos hacer un circuito que realice alguna función práctica, hay un 50% de probabilidades de que acabéis necesitando enviar información al mundo exterior (y claro no vais a dejar vuestro portátil pegado al invento).

Así que es el momento de ver que nos ofrece el mercado para mostrar información y para empezar, nada mejor que un sencillo **display LED** de 1 dígito.

Los diodos LED ya no guardan secretos para los que nos siguen, pero esto de un display parece otra cosa. Pues no. Es lo mismo pero en lugar de ser un punto luminoso como los familiares LED, tiene forma de barra: 7 barras más 1 punto.

Estos displays nos son bastante familiares en despertadores y similares, están formados por 7 líneas, que a dependiendo de qué segmentos se iluminan, muestran los dígitos del 0 al 9.

A cada uno de los 7 segmentos se les adjudica una letra para representarlos y suele ser a, b, c, e, f, g y aparte un punto (o dos, o dos puntos según el fabricante) y es un buen momento para echar un vistazo a las especificaciones de cada fabricante para estar seguros de que pin es cual.

Todos los circuitos, los displays incluidos, llevan un código de fabricante rotulado. Es buena política buscar en Internet la hoja de especificaciones(o data sheet).

- Desde que existe Internet, localizar una hoja de características es algo trivial (No como antes, que para conseguirlas había que pedir favores y mostrar las debidas muestras de respeto y sumisión a quien disponía de catálogos de componentes).
- Probad a buscar en Google algo así como XXXXX data sheet, donde XXXXX son los códigos que habéis encontrado rotulado en el display o componente que sea.

- Lo importante es que sepáis que es cada pin, y que veáis que segmento es cual. En realidad es bastante más fácil de lo que parece, localizad primero un GND. En el de arriba son el 3, 5, 11 y 16.

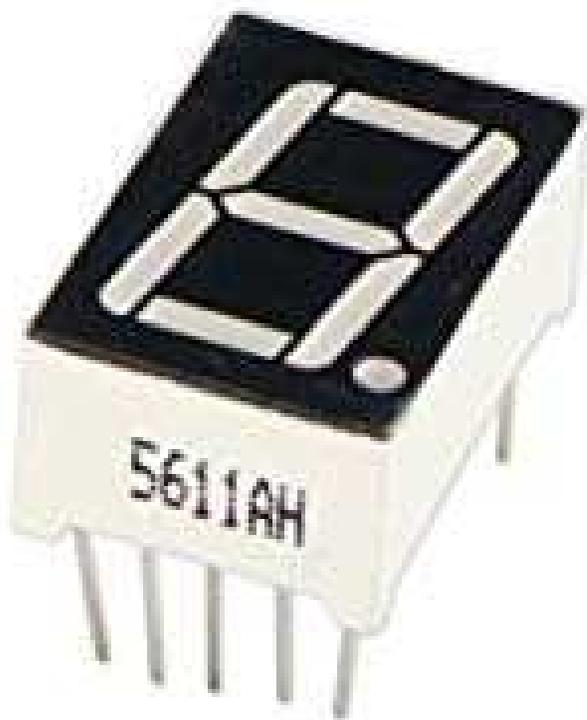
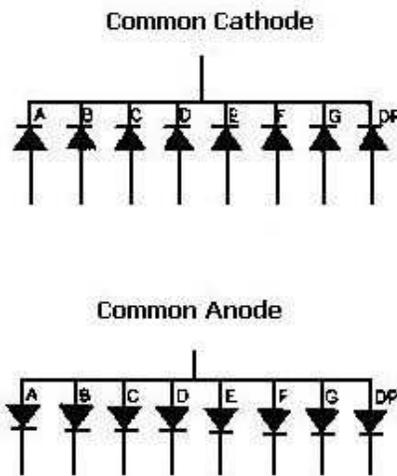
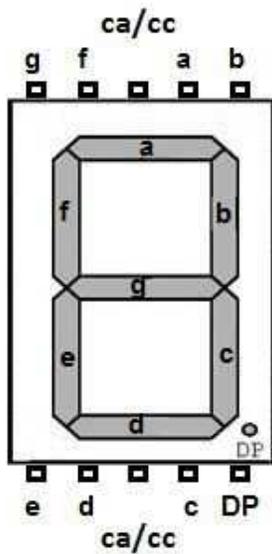
Poned una resistencia (de 330Ω o adecuada) en la salida del display a GND. Este es un display de cátodo común, o sea que todos los LEDS comparten la masa, pero necesitamos poner una resistencia de limitación o podemos dañar el display, y este vale ya lo suficiente, como para que os cueste verle la gracia al tema.

Una vez que hayáis conectado la masa común con una resistencia, empezad cableando solo un LED, el (a) por ejemplo y conectadlo al pin 2 de Arduino. Escribid un programa en Arduino para iluminar el LED en el pin 2 y veréis como se enciende el segmento a.

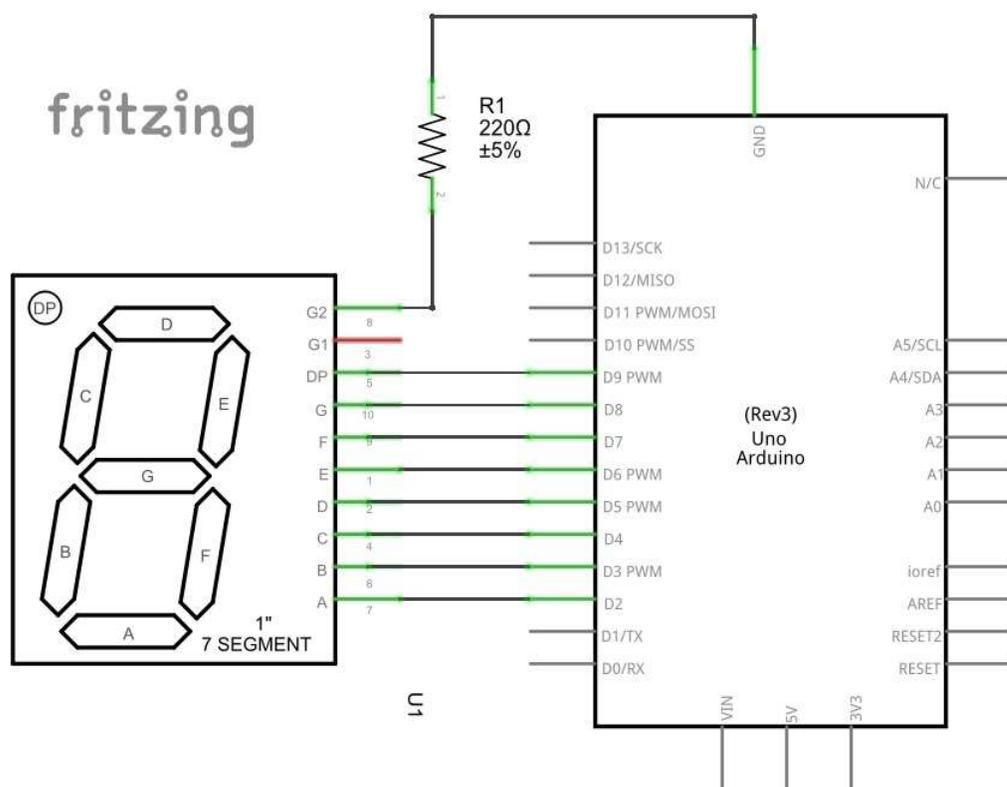
Podéis ir conectando y probando el resto de los pines poco a poco, asegurando las conexiones. Es muy fácil equivocarse si no andáis con tiento.

El display 7 segmentos que viene en los kit de Arduino Uno

El display que viene es un **5161AH** que tiene un pinout:



ESQUEMA ELÉCTRICO DEL CIRCUITO



El circuito no puede ser más sencillo, son como cuando hicimos un circuito con 8 diodos, solo que esta vez los 8 diodos están en una cajita y con forma para dibujar números.

- Es importante que sigáis el esquema de aquí encima para que luego el programa coincida con vuestras conexiones.
- Por eso no es importante que pin de vuestro display es el segmento C (Que dependerá del fabricante y modelo), pero si que es importante que el segmento C vaya al pin digital 4 de Arduino y lo mismo para los demás.
- No vamos a incluir el diagrama de protoboard, porque es difícil dejarlo más claro

PROGRAMA DE CONTROL

Lo siguiente es mostrar los dígitos del 0 al 9.

Para ello, lo más rápido es hacerte una tabla para ver que segmentos tienes que iluminar para cada dígito decimal, y vas apuntando cuales son 1 (encendido) o 0 (apagado).

	DIGITO	A	B	C	D	E	F	G	PUNTO
	0	1	1	1	1	1	1	0	0
	1	0	1	1	0	0	0	0	0
	2	1	1	0	1	1	0	1	0
	3	1	1	1	1	0	0	1	0
	4	0	1	1	0	0	1	1	0
	5	1	0	1	1	0	1	1	0
	6	1	0	1	1	1	1	1	0
	7	1	1	1	0	0	0	0	0
	8	1	1	1	1	1	1	1	0
	9	1	1	1	0	0	1	1	0

Aunque os damos los deberes hechos, hacéis la prueba con alguno de los dígitos, simplemente, para asegurarnos de que entendéis la idea.

Ahora solo queda escribir el programa que dibuja los números en el display, así que vamos a ello.

PROGRAMA QUE MUESTRA ENCENDIDA BARRA POR BARRA

```
//Enciende barra a barra una a la vez

int tiempo = 150;//definimos una variable de tipo entera para fijar retardo

void setup()
{
    // Coloque el codigo de configuracion (setup) aqui - se ejecutara solo una vez:

    pinMode(2, OUTPUT); // configura 'pin' como salida
    pinMode(3, OUTPUT);
    pinMode(4, OUTPUT);
    pinMode(5, OUTPUT);
    pinMode(6, OUTPUT);
    pinMode(7, OUTPUT);
    pinMode(8, OUTPUT);

}

void loop()
{
    // Coloque el codigo principal (main) aqui - se ejecutara repetidamente (loop):

    digitalWrite(2,HIGH );// PIN es colocado en estado alto (HIGH)
    delay(tiempo); // retardo
    digitalWrite(2,LOW ); // PIN es colocado en estado bajo (LOW)
    delay(tiempo); // otro retardo de 1000 mseg o 1 seg

    digitalWrite(3,HIGH );// PIN es colocado en estado alto (HIGH)
    delay(tiempo); // retardo
    digitalWrite(3,LOW ); // PIN es colocado en estado bajo (LOW)
    delay(tiempo); // otro retardo de 1000 mseg o 1 seg

    digitalWrite(4,HIGH );// PIN es colocado en estado alto (HIGH)
    delay(tiempo); // retardo
    digitalWrite(4,LOW ); // PIN es colocado en estado bajo (LOW)
    delay(tiempo); // otro retardo de 1000 mseg o 1 seg

    digitalWrite(5,HIGH );// PIN es colocado en estado alto (HIGH)
    delay(tiempo); // retardo
    digitalWrite(5,LOW ); // PIN es colocado en estado bajo (LOW)
    delay(tiempo); // otro retardo de 1000 mseg o 1 seg

    digitalWrite(6,HIGH );// PIN es colocado en estado alto (HIGH)
    delay(tiempo); // retardo
    digitalWrite(6,LOW ); // PIN es colocado en estado bajo (LOW)
    delay(tiempo); // otro retardo de 1000 mseg o 1 seg

    digitalWrite(7,HIGH );// PIN es colocado en estado alto (HIGH)
    delay(tiempo); // retardo
    digitalWrite(7,LOW ); // PIN es colocado en estado bajo (LOW)
    delay(tiempo); // otro retardo de 1000 mseg o 1 seg

    digitalWrite(8,HIGH );// PIN es colocado en estado alto (HIGH)
    delay(tiempo); // retardo
    digitalWrite(8,LOW ); // PIN es colocado en estado bajo (LOW)
    delay(tiempo); // otro retardo de 1000 mseg o 1 seg

}
```

PROGRAMA QUE MUESTRA DIGITO A DIGITO

```
//Enciende de un digito a la vez

int tiempo = 500;//definimos una variable de tipo entera para fijar retardo

void setup()
{
    // Coloque el codigo de configuracion (setup) aqui - se ejecutara solo una vez:

    pinMode(2, OUTPUT); // configura 'pin' como salida
    pinMode(3, OUTPUT);
    pinMode(4, OUTPUT);
    pinMode(5, OUTPUT);
    pinMode(6, OUTPUT);
    pinMode(7, OUTPUT);
    pinMode(8, OUTPUT);

}

void loop()
{
    // Coloque el codigo principal (main) aqui - se ejecutara repetidamente (loop):

    //Muestra el cero

    digitalWrite(2,HIGH );// PIN es colocado en estado alto (HIGH)
    digitalWrite(3,HIGH );// PIN es colocado en estado alto (HIGH)
    digitalWrite(4,HIGH );// PIN es colocado en estado alto (HIGH)
    digitalWrite(5,HIGH );// PIN es colocado en estado alto (HIGH)
    digitalWrite(6,HIGH );// PIN es colocado en estado alto (HIGH)
    digitalWrite(7,HIGH );// PIN es colocado en estado alto (HIGH)
    digitalWrite(8,LOW ); // PIN es colocado en estado bajo (LOW)
    delay(tiempo); // retardo

    //Muestra el uno

    digitalWrite(2,LOW );
    digitalWrite(3,HIGH );
    digitalWrite(4,HIGH );
    digitalWrite(5,LOW );
    digitalWrite(6,LOW );
    digitalWrite(7,LOW );
    digitalWrite(8,LOW );
    delay(tiempo); // retardo

    //Muestra el dos

    digitalWrite(2,HIGH );
    digitalWrite(3,HIGH );
    digitalWrite(4,LOW );
    digitalWrite(5,HIGH );
    digitalWrite(6,HIGH );
    digitalWrite(7,LOW );
    digitalWrite(8,HIGH );
    delay(tiempo); // retardo
```

```
//Muestra el tres
```

```
digitalWrite(2,HIGH );  
digitalWrite(3,HIGH );  
digitalWrite(4,HIGH );  
digitalWrite(5,HIGH );  
digitalWrite(6,LOW );  
digitalWrite(7,LOW );  
digitalWrite(8,HIGH );  
delay(tiempo); // retardo
```

```
//Muestra el cuatro
```

```
digitalWrite(2,LOW );  
digitalWrite(3,HIGH );  
digitalWrite(4,HIGH );  
digitalWrite(5,LOW );  
digitalWrite(6,LOW );  
digitalWrite(7,HIGH );  
digitalWrite(8,HIGH );  
delay(tiempo); // retardo
```

```
//Muestra el cinco
```

```
digitalWrite(2,HIGH );  
digitalWrite(3,LOW );  
digitalWrite(4,HIGH );  
digitalWrite(5,HIGH );  
digitalWrite(6,LOW );  
digitalWrite(7,HIGH );  
digitalWrite(8,HIGH );  
delay(tiempo); // retardo
```

```
//Muestra el seis
```

```
digitalWrite(2,HIGH );  
digitalWrite(3,LOW );  
digitalWrite(4,HIGH );  
digitalWrite(5,HIGH );  
digitalWrite(6,HIGH );  
digitalWrite(7,HIGH );  
digitalWrite(8,HIGH );  
delay(tiempo); // retardo
```

```
//Muestra el siete
```

```
digitalWrite(2,HIGH );  
digitalWrite(3,HIGH );  
digitalWrite(4,HIGH );  
digitalWrite(5,LOW );  
digitalWrite(6,LOW );  
digitalWrite(7,LOW );  
digitalWrite(8,LOW );  
delay(tiempo); // retardo
```

```
//Muestra el ocho
```

```
digitalWrite(2,HIGH );  
digitalWrite(3,HIGH );  
digitalWrite(4,HIGH );  
digitalWrite(5,HIGH );
```

```
digitalWrite(6,HIGH );  
digitalWrite(7,HIGH );  
digitalWrite(8,HIGH );  
delay(tiempo); // retardo
```

```
//Muestra el nueve
```

```
digitalWrite(2,HIGH );  
digitalWrite(3,HIGH );  
digitalWrite(4,HIGH );  
digitalWrite(5,LOW );  
digitalWrite(6,LOW );  
digitalWrite(7,HIGH );  
digitalWrite(8,HIGH );  
delay(tiempo); // retardo
```

```
}
```

