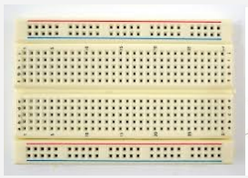
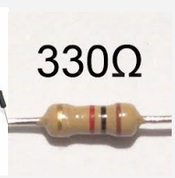
DISPLAY DE 7 SEGMENTOS x 4 *(versión 16-10-18)*

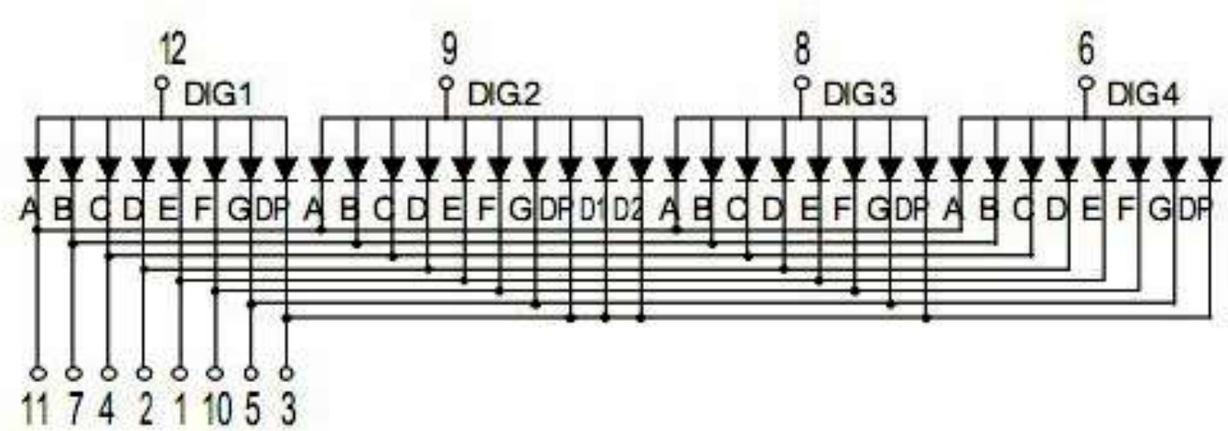
Mostrando informacion numérica

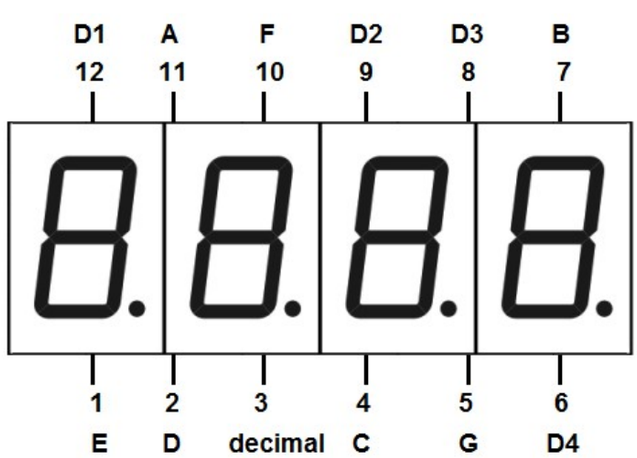
OBJETIVOS Manejar nuestro primer display numérico de 4 dígitos.

MATERIAL REQUERIDO.

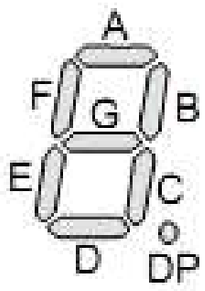
	**Arduino Uno o similar.** *Esta sesión acepta cualquier otro modelo de **Arduino**.*
  	Un Protoboard.- Cables de protoboard Cuatro resistencia. (podria tener un valor menor)
	Un display de 4 dígitos.

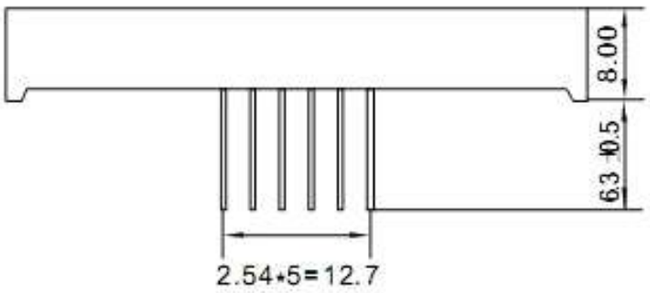
Se trata de un display de ánodo común (en el que viene generalmente en los kit de Arduino). Además se puede ver que los pines de cada segmento están compartidos para los 4 dígitos. Por tanto, para poder mostrar un número de 4 dígitos es necesario multiplexar la señal, es decir, iluminar secuencialmente cada uno de los dígitos en una sucesión muy rápida, creando la ilusión de que todos los dígitos están encendidos a la vez.

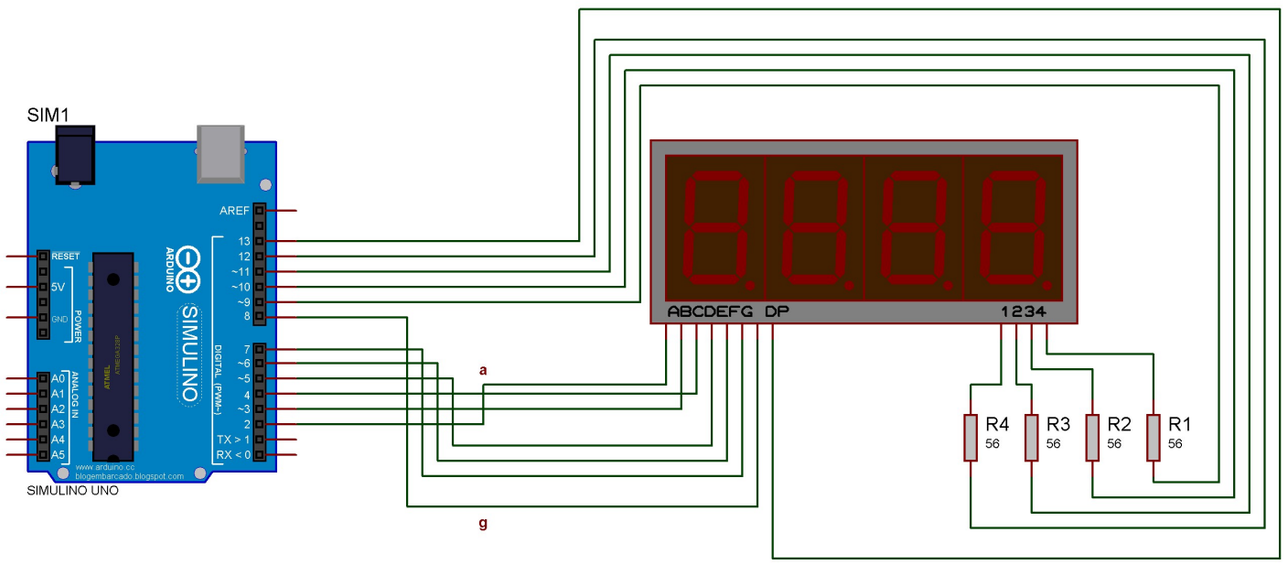




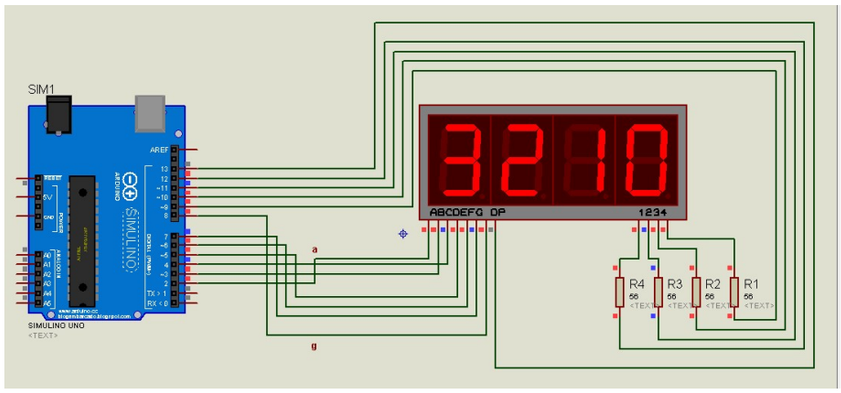
Esta figura es fundamental para el conexionado.







Las resistencias en el caso real fueron de aproximadamente 100 ohms.



NOTA: Se utilizaron en el caso real resistencias de 100 ohms.

Con el efecto explicado anteriormente y lo que vimos para encender display de un solo digito, estamos en condiciones de hacer una rutina (programa o sketch) para mostrar mas dígitos.

Afortunadamente de todo esto se encarga una librería de Arduino, llamada **SevSeg**, por lo que lo único que debemos decirle son los pines utilizados, y el número que queremos mostrar.
En el siguiente código mostramos la implementación de un contador de 0 a 100 segundos, con una precisión de centésimas de segundo.

Biblioteca SevSeg

Escrito por Dean Reading.
¡Esta biblioteca convierte tu Arduino en un controlador de pantalla de siete segmentos! Úselo para mostrar fácilmente los números en su pantalla de siete segmentos sin ningún controlador adicional.
Admite pantallas de cátodos comunes y ánodos comunes, y el uso de transistores de conmutación. Se pueden usar pantallas con cualquier número de dígitos, y se admiten decimales.

La versión actual no es compatible con las versiones anteriores de la biblioteca SevSeg , que están disponibles desde 2012. Puede descargar la versión anterior para que sea compatible con los programas escritos anteriormente. .

Mas detalles

HARDWARE

Siete pines de visualización de segmento
Uno para cada dígito. Estos son los 'pins comunes'. Serán cátodos (pines negativos) para pantallas de cátodos comunes, o ánodos (pines positivos) para pantallas de ánodos comunes. O según pantalla, 8 pines de segmento : uno para cada uno de los siete segmentos más el punto decimal.

Conexiones Arduino
Todos los pines de dígitos y pines de segmento se pueden conectar a cualquiera de los pines digitales o analógicos de Arduino; ¡Solo asegúrate de tomar nota de tus conexiones!

Resistores limitadores de corriente
No olvide que la pantalla utiliza LED , por lo que debe usar resistencias limitadoras de corriente en serie con los pines de dígitos *. 330 ohms es un valor seguro si tiene dudas al respecto. Si utiliza resistencias limitadoras de corriente en los* pines *del* segmento *, abra el archivo SevSeg .h y establezca RESISTORS_ON_SEGMENTS en 1 para un brillo óptimo.*

Configuración de hardware
Tienes que especificar tu configuración de hardware a la biblioteca. Las opciones se detallan a continuación.
Pantallas simples de baja potencia
Estas pantallas se alimentan directamente a través de los pines de salida de Arduino.
COMMON_CATHODE Para pantallas comunes de cátodos. Estas pantallas requieren un voltaje bajo en el pin del dígito para iluminar el dígito.
COMMON_ANODE Para pantallas comunes de ánodos. Estas pantallas requieren un voltaje alto en el pin del dígito para iluminar el dígito.

SOFTWARE

Instalada la librería, a continuación se muestra como se utiliza en un ejemplo de contador.

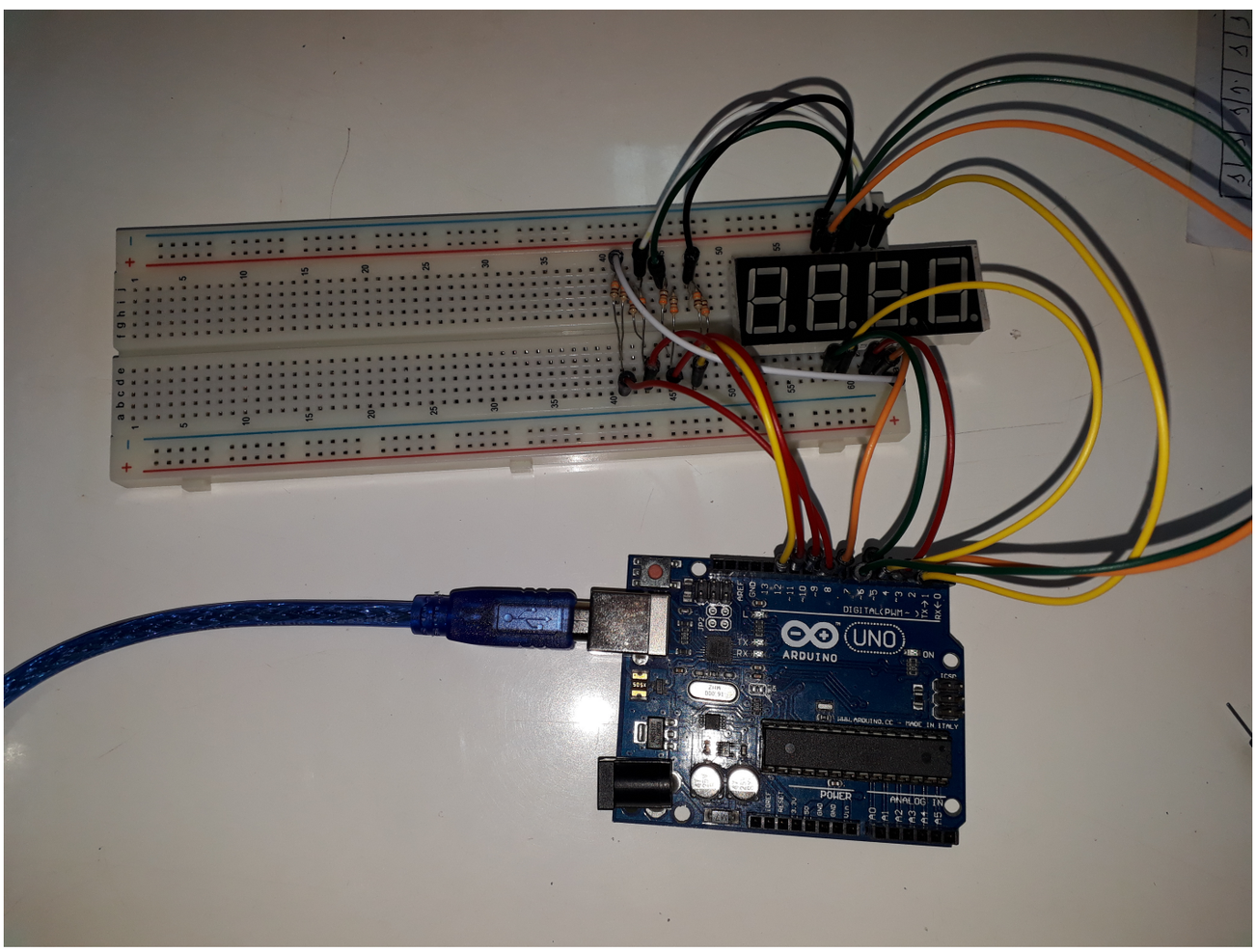
Encontrara en el material del docente documentación de esta librería. Originalmente esta en ingles, pero se ha agregado una traducción de cada párrafo mediante el traductor de Google.

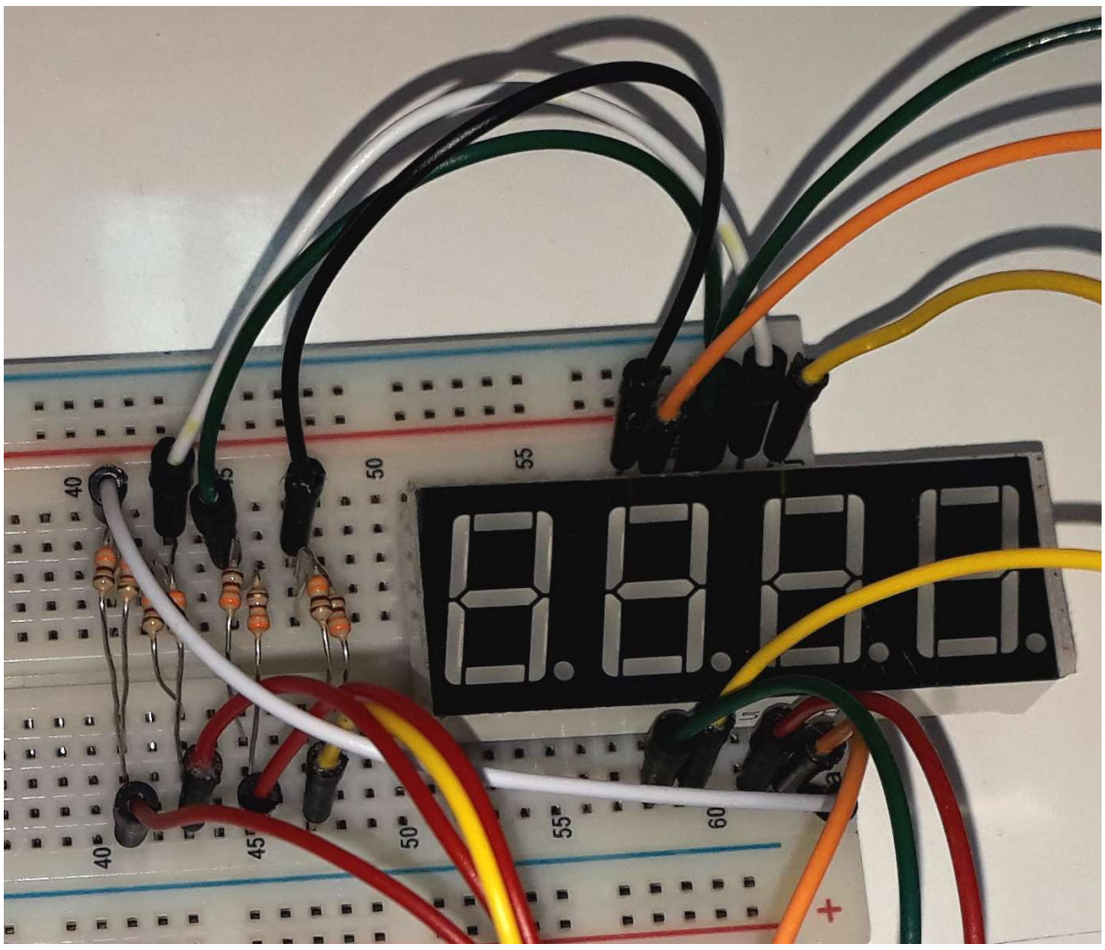
EJEMPLO DE UN CONTADOR

```
//Mostramos aqui un contador
//Usando la libreria
#include <SevSeg.h>
 SevSeg sevseg; //creamos una instancia llamada sevseg- el nuevo nombre

// Variables globales

int Contador = 0;// cero si inicia

int espera =0;// contador espera

int esperamax =3000;//Define tiempo entre cuenta

void setup()
{
  //I am using a common anode display, with the digit pins connected
  //from 2-5 and the segment pins connected from 6-13

   byte numDigits = 4;
   byte digitPins [] = {12,11,10,9};
   byte segmentPins [] = {2,3,4,5,6,7,8,13};


bool resistorsOnSegments = false; // 'false'" significa que las resistencias están en los pines de dígitos
                          //pines de digitos son solo 4
   byte hardwareConfig = COMMON_CATHODE; // Catodo comun ->COMMON_CATHODE
                                //ANODO COMUN ->COMMON_ANODE
   sevseg.begin(hardwareConfig, numDigits, digitPins, segmentPins, resistorsOnSegments);
```

```
  // Define la luminosidad de 0 a 100
 sevseg.setBrightness(99);

}

void loop()
{

    // Produce una salida en el display
       sevseg.refreshDisplay();
      // Actualiza el número mostrado, con el punto decimal
     sevseg.setNumber(Contador, -1);//Muestra numero con punto en 2do digito si 2, 0 si primero
                                    // Indicar -1 si ninguno
      espera = espera +1; //incrementa espera para definir tiempo entre cuenta y cuenta

if (espera == esperamax)
  {
  espera =0;
  Contador = Contador + 1;

  }

  }
```

La versión de la biblioteca empleada se podrá encontrar también en la pagina del docente.





NOTA: Se utilizaron para los digitos 2 resistencias de 330 ohms en paralelo por no contar en ese momento con alguna menor.

FUENTE:

http://playground.arduino.cc/Main/SevenSegmentLibrary

http://www.leandroruiz.com/blog/display-reloj-despertador/