
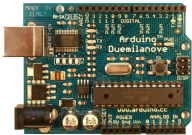


| PRÁCTICAS CON ARDUINO | | | |
|---|---|--|---|
|  | Nombres y apellidos: | Curso: Fecha: |  |

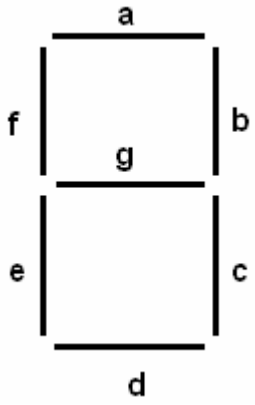
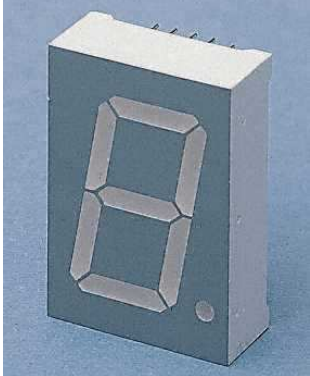
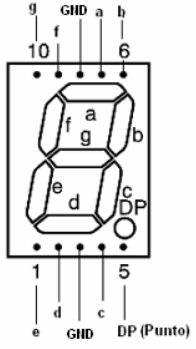
PRIMERA ACTIVIDAD

OBJETIVO: Realizar un contador con dos displays 7 segmentos (Cuenta en segundos de 0 A 99)

MATERIAL:

- 2 Displays 7 segmentos de cátodo común
- Placa arduino
- Cable.

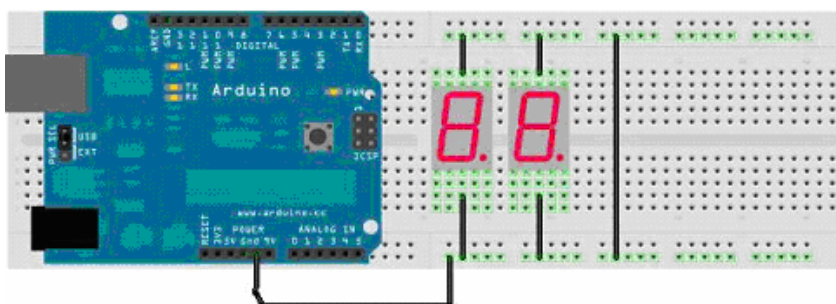
FUNDAMENTO TEÓRICO: El display 7 segmentos consiste en un conjunto de 7 leds encapsulados y conformados de tal manera que, según los que estén luciendo, muestren caracteres alfanuméricos. Pueden ser de ánodo o cátodo común, según estén unidos todos los ánodos o todos los cátodos.

| | | |
|---|---|---|
|  |  | <p>DISPLAY 7 SEGMENTOS</p>  |
|---|---|---|

MONTAJE

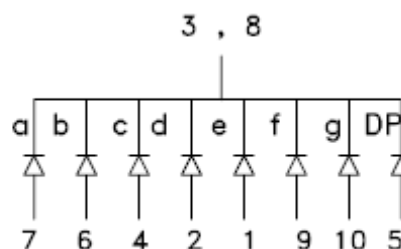
El montaje será el siguiente:

```
int D1_segmento_a = 0; // Patilla 7 del Display 1
int D1_segmento_b = 1; // Patilla 6 del Display 1
int D1_segmento_c = 2; // Patilla 4 del Display 1
int D1_segmento_d = 3; // Patilla 2 del Display 1
int D1_segmento_e = 4; // Patilla 1 del Display 1
int D1_segmento_f = 5; // Patilla 9 del Display 1
int D1_segmento_g = 6; // Patilla 10 del Display 1
int D2_segmento_a = 7; // Patilla 7 del Display 2
int D2_segmento_b = 8; // Patilla 6 del Display 2
int D2_segmento_c = 9; // Patilla 4 del Display 2
int D2_segmento_d = 10; // Patilla 2 del Display 2
int D2_segmento_e = 11; // Patilla 1 del Display 2
int D2_segmento_f = 12; // Patilla 9 del Display 2
int D2_segmento_g = 13; // Patilla 10 del Display 2
```



Las patillas 3 y 8 de cada display van conectadas al negativo (GND), ya que los displays utilizados son de cátodo común. En el caso de que fueran de ánodo común, se conectarían al positivo de la alimentación

Para enviar los datos a los displays utilizamos el puerto digital (D0-D13). Es decir, utilizamos las patillas digitales 0-13 (14 pines en total).



Nota:

Patillaje del display de cátodo común

EXPLICACIÓN DEL PROGRAMA

1. Definimos cada segmento de cada display como una variable entera

```
int D1_segmento_a = 0; // Patilla 7 del Display 1
int D1_segmento_b = 1; // Patilla 6 del Display 1
int D1_segmento_c = 2; // Patilla 4 del Display 1
int D1_segmento_d = 3; // Patilla 2 del Display 1
int D1_segmento_e = 4; // Patilla 1 del Display 1
int D1_segmento_f = 5; // Patilla 9 del Display 1
int D1_segmento_g = 6; // Patilla 10 del Display 1
int D2_segmento_a = 7; // Patilla 7 del Display 2
int D2_segmento_b = 8; // Patilla 6 del Display 2
int D2_segmento_c = 9; // Patilla 4 del Display 2
int D2_segmento_d = 10; // Patilla 2 del Display 2
int D2_segmento_e = 11; // Patilla 1 del Display 2
int D2_segmento_f = 12; // Patilla 9 del Display 2
int D2_segmento_g = 13; // Patilla 10 del Display 2
```

2. Definimos las patillas digitales como salidas

```
void setup() {
  pinMode(D1_segmento_a, OUTPUT);
  pinMode(D1_segmento_b, OUTPUT);
  pinMode(D1_segmento_c, OUTPUT);
  pinMode(D1_segmento_d, OUTPUT);
  pinMode(D1_segmento_e, OUTPUT);
  pinMode(D1_segmento_f, OUTPUT);
  pinMode(D1_segmento_g, OUTPUT);
  pinMode(D2_segmento_a, OUTPUT);
  pinMode(D2_segmento_b, OUTPUT);
  pinMode(D2_segmento_c, OUTPUT);
  pinMode(D2_segmento_d, OUTPUT);
  pinMode(D2_segmento_e, OUTPUT);
  pinMode(D2_segmento_f, OUTPUT);
  pinMode(D2_segmento_g, OUTPUT);
}
```

3. La función principal LOOP ejecuta la función creada D2_num (para las decenas-display 2-) y la función unidades. Se le introduce un retardo de un segundo.

```
void loop()
{
  delay(5000);
  D2_num_0();
  unidades();
  D2_num_1();
  unidades();
  D2_num_2();
  unidades();
  D2_num_3();
  unidades();
  D2_num_4();
  unidades();
  D2_num_5();
  unidades();
  D2_num_6();
  unidades();
  D2_num_7();
  unidades();
  D2_num_8();
  unidades();
  D2_num_9();
  unidades();
}
```

4. Las funciones particulares activan los leds correspondientes con el número que se quiere mostrar. Por ejemplo, el 9 del display 2:

```
void D1_num_9()
{
  digitalWrite(D1_segmento_a, HIGH);
  digitalWrite(D1_segmento_b, HIGH);
  digitalWrite(D1_segmento_c, HIGH);
  digitalWrite(D1_segmento_d, LOW);
  digitalWrite(D1_segmento_e, LOW);
}
```

```
digitalWrite(D1_segmento_f, HIGH);  
digitalWrite(D1_segmento_g, HIGH);  
}
```

5. La función unidades() está implementada agrupando las funciones parciales de cada dígito.

```
void unidades()  
{  
  D1_num_0();  
  delay(1000);  
  D1_num_1();  
  delay(1000);  
  D1_num_2();  
  delay(1000);  
  D1_num_3();  
  delay(1000);  
  D1_num_4();  
  delay(1000);  
  D1_num_5();  
  delay(1000);  
  D1_num_6();  
  delay(1000);  
  D1_num_7();  
  delay(1000);  
  D1_num_8();  
  delay(1000);  
  D1_num_9();  
  delay(1000);  
}
```