

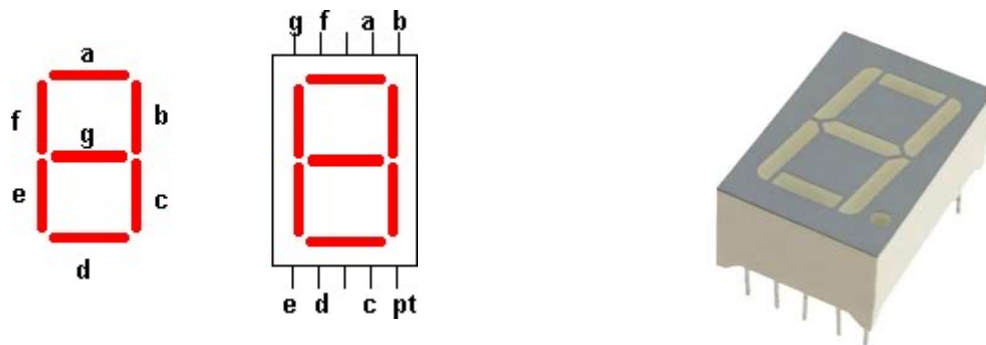
GRUPO N° 3.

PROFESORES: GONZALO VERA, JORGE MORALES.

a) Como funcionan los display 7 segmentos gigantes y cuales son sus aplicaciones.

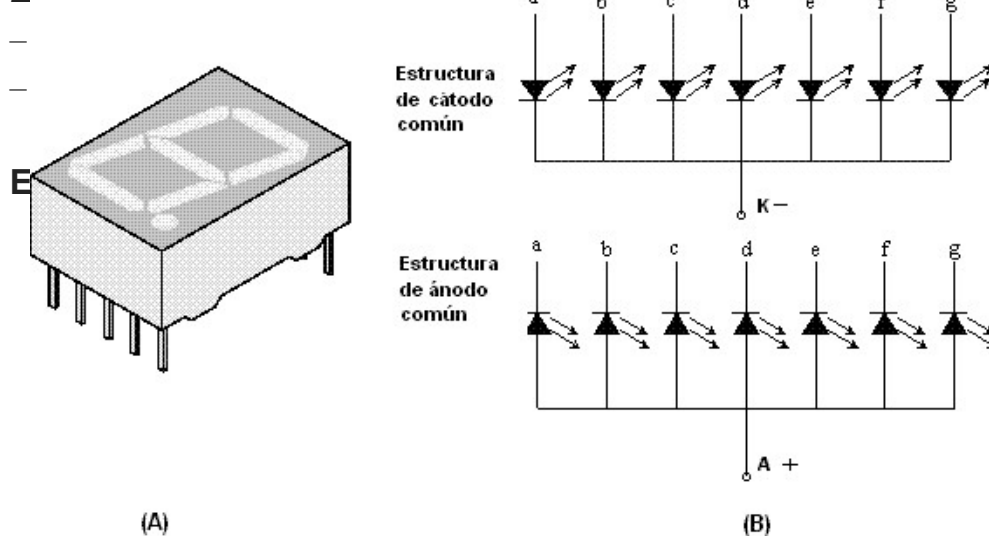
DISPLAY DE 7 SEGMENTOS

El display de 7 segmentos es un componente que se utiliza para la representación de números, es muy común encontrarlos en dispositivos electrónicos debido a su simplicidad su valor aproximado es de usd 1. Está compuesto internamente por una serie de LEDS ubicados de tal forma que forme un número 8.



A cada uno de los segmentos que forman el display se les denomina a, b, c, d, e, f y g, estando ensamblados de forma que se permita activar cada segmento por separado consiguiendo formar cualquier dígito numérico. Muchas veces aparece un octavo segmento denominado p.d. (punto decimal).

Existen dos tipos de display de 7 segmentos:



En los displays de tipo ánodo común, todos los ánodos de los LEDS están unidos internamente a una pata común que debe ser conectada a potencial positivo.

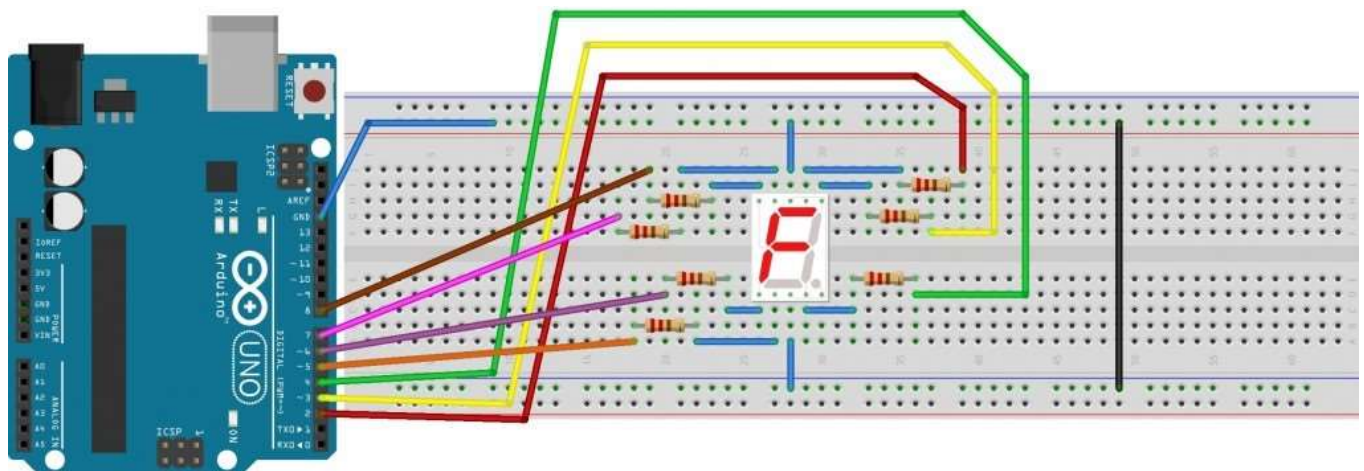
En esta ocasión utilizaremos el display de **cátodo común**. Éste tiene todos los cátodos de los diodos unidos internamente a una pata en común la cual se conecta a ground. El encendido de cada LED se realiza aplicando voltaje a la pata correspondiente a través de una resistencia.

Las 2 patas del medio del display serán conectadas al potencial positivo si se trata de un display de ánodo común y en nuestro caso utilizamos un display de cátodo común éstas patas serán el ground.

En esta experiencia utilizamos:

- Arduino Uno
- display de siete segmentos (Cátodo común)
- 7 resistencias de 330 Ω

Cuando tenemos todos los materiales, procedemos a realizar las siguientes conexiones:



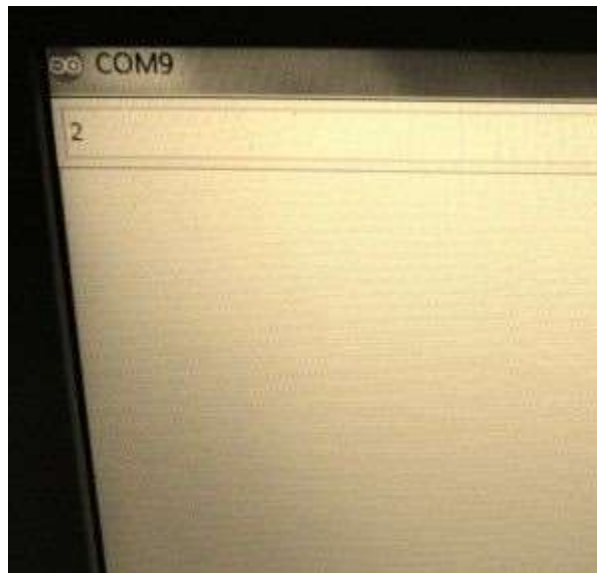
Abrimos Arduino IDE, y copiamos el siguiente código:

```
1 int mensaje=0;
2 void setup(){
3
4   Serial.begin(9600);
5   pinMode(2, OUTPUT);
6   pinMode(3, OUTPUT);
7   pinMode(4, OUTPUT);
8   pinMode(5, OUTPUT);
9   pinMode(6, OUTPUT);
10  pinMode(7, OUTPUT);
11  pinMode(8, OUTPUT);
12
13 }
14 void display (int a, int b, int c, int d, int e, int f, int g) // Función del display
15 {
16   digitalWrite (2,a);
17   digitalWrite (3,b);
18   digitalWrite (4,c);
19   digitalWrite (5,d);
20   digitalWrite (6,e);
21   digitalWrite (7,f);
22   digitalWrite (8,g);
23 }
24
25 void loop(){
26
27   if (Serial.available()>0){
28     mensaje=Serial.read();
29
30     if (mensaje=='0'){
31       display (1,1,1,1,1,1,0);
32     }
33
34     if (mensaje=='1'){
35       display (0,1,1,0,0,0,0);
36     }
37     if (mensaje=='2'){
38       display (1,1,0,1,1,0,1);
39     }
40     if (mensaje=='3'){
41       display (1,1,1,1,0,0,1);
42     }
43     if (mensaje=='4'){
44       display (0,1,1,0,0,1,1);
45     }
46     if (mensaje=='5'){
47       display (1,0,1,1,0,1,1);
48     }
49     if (mensaje=='6'){
50       display (1,0,1,1,1,1,1);
51     }
52     if (mensaje=='7'){
53       display (1,1,1,0,0,0,0);
54     }
55     if (mensaje=='8'){
56       display (1,1,1,1,1,1,1);
57     }
58     if (mensaje=='9'){
59       display (1,1,1,0,0,1,1);
60     }
61
62   }
63   delay(2000);
64 }
```

Utilizamos la función ***void display*** para establecer los parámetros que van a controlar el encendido de cada LED o segmento.

Con este conjunto de instrucciones podemos mostrar en el display cualquier número del 0 al 9 a través de una comunicación serial.

Después de cargar el código, entramos al monitor serial y escribimos cualquier número de un dígito.



Presionamos ENTER e inmediatamente podemos observar en el display el número que colocamos.

