

# MICROS 32 BITS

## STM - GPIO

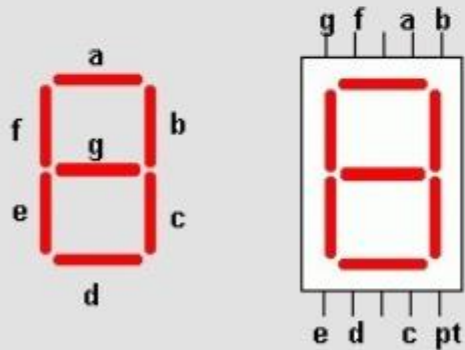
ROBINSON JIMENEZ MORENO



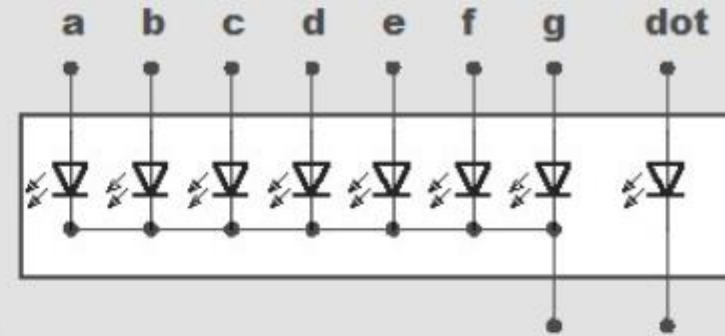
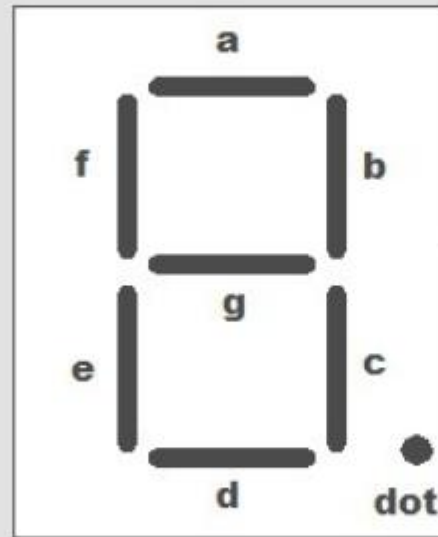
# DISPLAY 7 SEGMENTOS



**Forma física de un display**

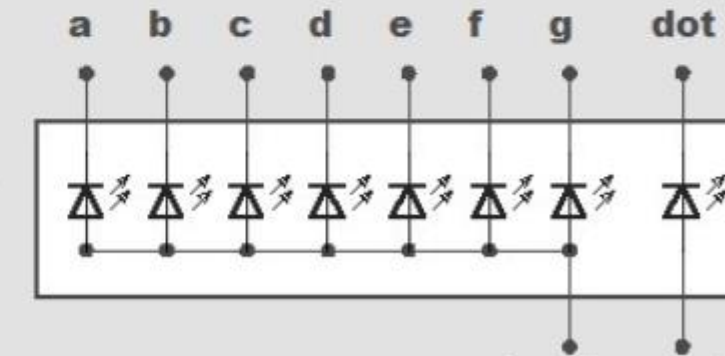


**Clasificación de pines**



**cátodo común**

**los segmentos encienden con uno lógico**



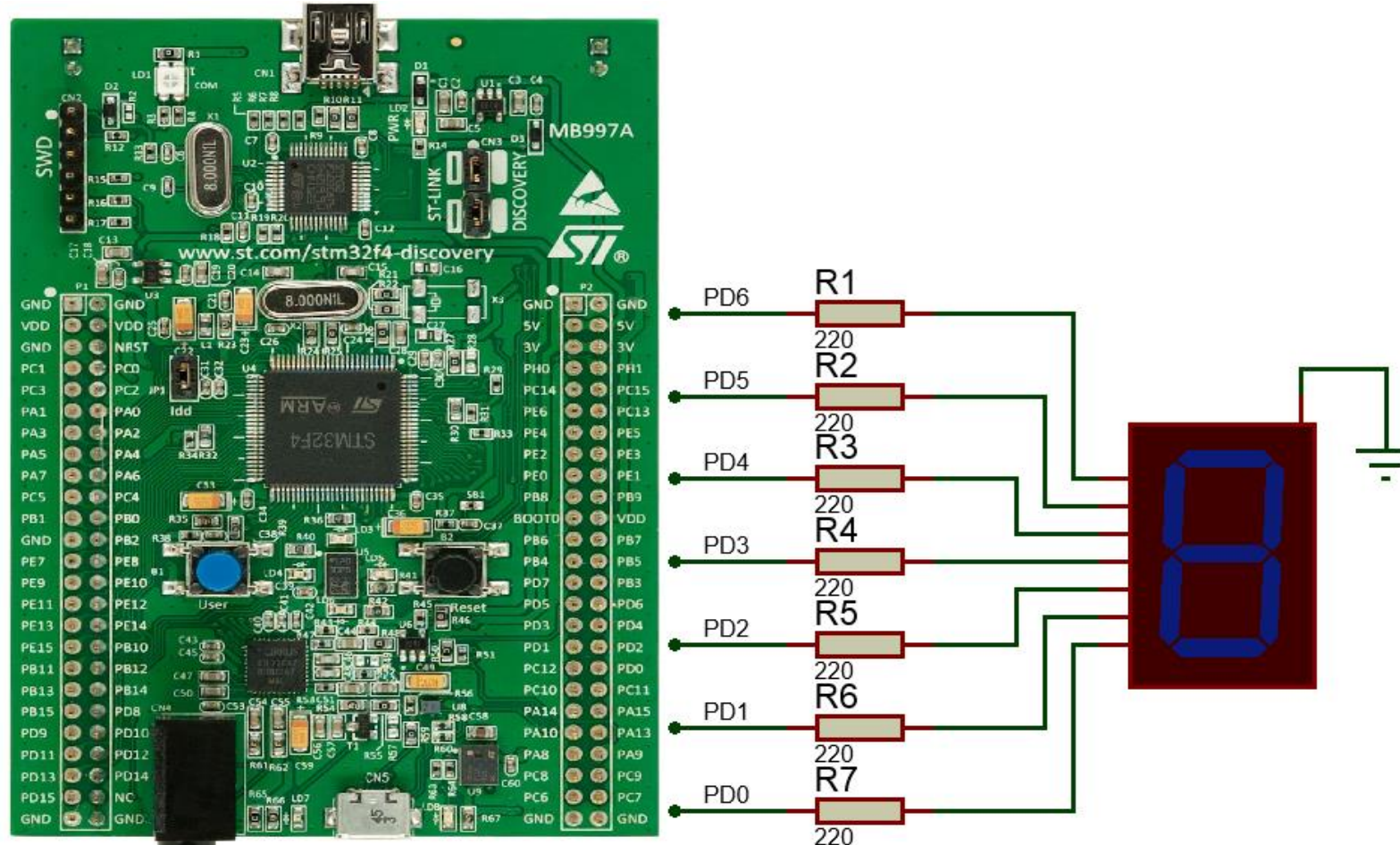
**ánodo común**

**los segmentos encienden con cero lógico**

[https://articulo.mercadolibre.cl/MLC-456827403-5-display-7-segmentos-anodo-comun-\\_JM?quantity=1](https://articulo.mercadolibre.cl/MLC-456827403-5-display-7-segmentos-anodo-comun-_JM?quantity=1)

# Visualización en Display 7 Seg

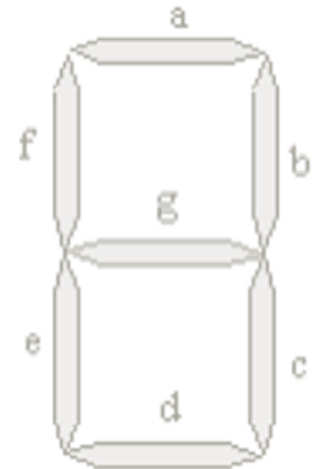
Visualizar los números del 0 al 9 en un display 7 segmentos



# DISPLAY 7 SEGMENTOS

gfedcba	codigo	signif	abcdefg	codigo	signif
0111111	63	# 0	?	?	# 0
0000110	6	# 1	?	?	# 1
1011011	91	# 2	?	?	# 2
1001111	79	# 3	?	?	# 3
1100110	102	# 4	?	?	# 4
0101101	45	# 5	?	?	# 5
0111101	61	# 6	?	?	# 6
0000111	7	# 7	?	?	# 7
1111111	127	# 8	?	?	# 8
1101111	111	# 9	?	?	# 9

PX0=Segmento a  
 PX1=Segmento b  
 PX2=Segmento X  
 PX3=Segmento d  
 PX4=Segmento e  
 PX5=Segmento f  
 PX6=Segmento g  
 PX7=Segmento .



```
1  #include <stdio.h>
2  #include "stm32f7xx.h"
3  int numeros[10]={126,48,109,121,51,91,95,112,127,123};
4  int a;
5  int main(void) {
6
7      RCC -> AHB1ENR = 0X6; //HABILITAR LOS PUERTOS B Y C
8
9      GPIOB -> MODER = 0X000000FF; //
10     GPIOB -> OTYPER = 0X0; //PUSH PULL
11     GPIOB -> OSPEEDR = 10004001; //VELOCIDAD MEDIA
12     GPIOB -> PUPDR = 0X100000FF; //PULL UP
13
14     while(1) {
15
16         for( a = 0; a < 10; a++){};
17         GPIOB -> ODR ^= numeros[a];
18         for(int i = 0; i < 100000; i++){};
19     }
20 }
21
22
```





```
#include "STM32F7xx.h"

int time=100000,cont=0;

char BCD [6] = {0xFE,0xFD,0xFB,0xF7,0xEF,0xDF};

int main(void)
[ { int i=0;
    //CONFIGURACION "CLOCK"
    RCC->AHB1ENR =0xFFFF;
    //CONFIGURACION DE PINES
    GPIOF->MODER    = 0x55555555;
    GPIOF->OTYPER   = 0;
    GPIOF->OSPEEDR  = 0x55555555;
    GPIOF->PUPDR    = 0x55555555;

    GPIOG->MODER    = 0x55555555;
    GPIOF->ODR=2;
    GPIOG->ODR=0;

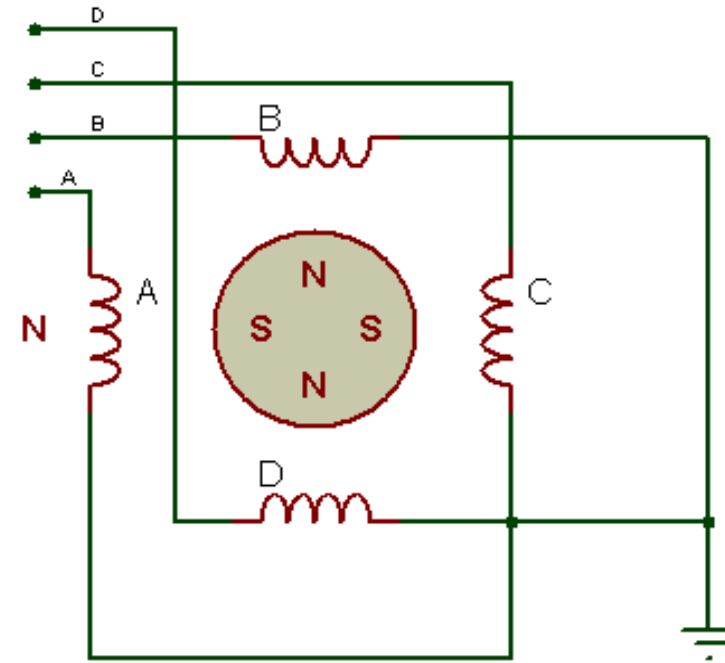
] while(true){           //bucle infinito
] for(cont=0;cont<6;cont++){ //FOR2
    GPIOG->ODR=BCD[cont];
    for(i=0;i<time;i++);
} //FIN FOR 1
- }
}
```



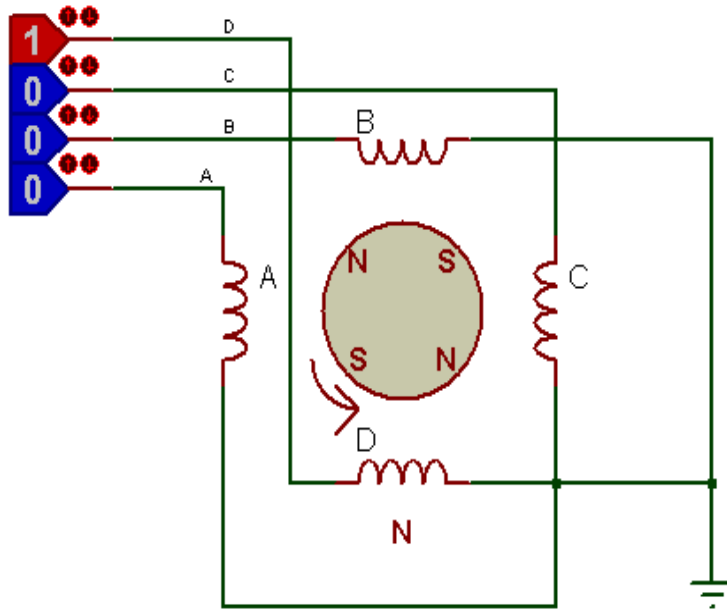
```
////////////////////////////////////  
////////////////////NUMEROS BASICO INCREMENTO////////////////////  
  
#include "STM32F7xx.h"  
  
int time=500000,cont=0;  
  
char BCD [14] = {0XC0,0xF9,0XA4,0XB0,0X99,0X92,0X83,0XF8,0X80,0X98,0xBF,0x7F};  
  
int main(void)  
{ int i=0;  
  //CONFIGURACION "CLOCK"  
  RCC->AHB1ENR =0xFFFF;  
  //CONFIGURACION DE PINES  
  GPIOF->MODER    = 0x55555555;  
  GPIOF->OTYPER    = 0;  
  GPIOF->OSPEEDR   = 0x55555555;  
  GPIOF->PUPDR     = 0x55555555;  
  
  GPIOG->MODER     = 0x55555555;  
  GPIOF->ODR=2;  
  GPIOG->ODR=0;  
  
  while(true){          //bucle infinito  
  
for(cont=0;cont<10;cont++){ //FOR2  
    GPIOG->ODR=BCD[cont];  
    for(i=0;i<time;i++);  
  } //FIN FOR 2  
}  
}
```



# MOTOR PASO A PASO







MOTOR PASO PASO				
A	B	C	D	GIRO
1	0	0	0	45°
0	1	0	0	90°
0	0	1	0	135°
0	0	0	1	180°
1	0	0	0	225°
0	1	0	0	270°
0	0	1	0	315°
0	0	0	1	360°
1	0	0	0	45°
0	1	0	0	90°

```

2  #include <stdio.h>
3  #include "stm32f7xx.h"
4  int pasos[10]={1,2,4,8};
5  int tiempo=0;
6
7  int main(void){
8
9      RCC -> AHB1ENR = 0X2; //PUERTO B
10     GPIOB -> MODER = 0X555555; //SALIDA BOBINAS
11     GPIOB -> OTyPER = 0X0; //PUSH PULL
12     GPIOB -> OSPEEDR = 0X555555; //VELOCIDAD MEDIA
13     GPIOB -> PUPDR = 0; //PULL UP;
14
15     while(1){
16         for(int a=0;a<4;a++){
17             GPIOB->ODR= pasos[a];
18             for(tiempo=0;tiempo<100000;tiempo++);
19         }
20     }
21 }

```



# TAREA

Realizar un programa que actúe como calculadora con operaciones de dos dígitos (salida por display), suma, resta, multiplicación, división, elevado al cuadrado, al cubo, tecla de factorial y tecla de igual. (se evidencia que debe existir una tecla de segunda función)

