

MICROCONTROLADORES

Práctica No. 7. Contador con Display de 7 segmentos.

1. Objetivo

- Utilizar un Display de 7 segmentos.

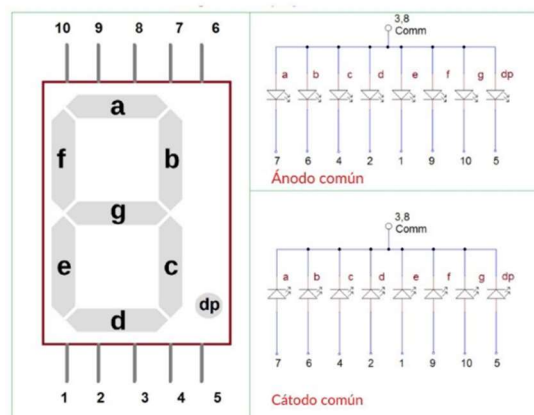
2. Material y Equipo.

- Computador o laptop con el STM32CubeIDE.
- Un Display de 7 Segmentos de cátodo común.
- Siete resistencias de 330Ω o 220Ω .

3. Marco de Referencia

Los displays de 7 segmentos son dispositivos muy utilizados para desplegar valores numéricos. Hay dos versiones de este dispositivo el de cátodo común (enciende con 1's) y el de ánodo común (enciende con 0's). La siguiente figura muestra el diagrama de un display de cátodo común y ánodo común.

Display de 7 segmentos

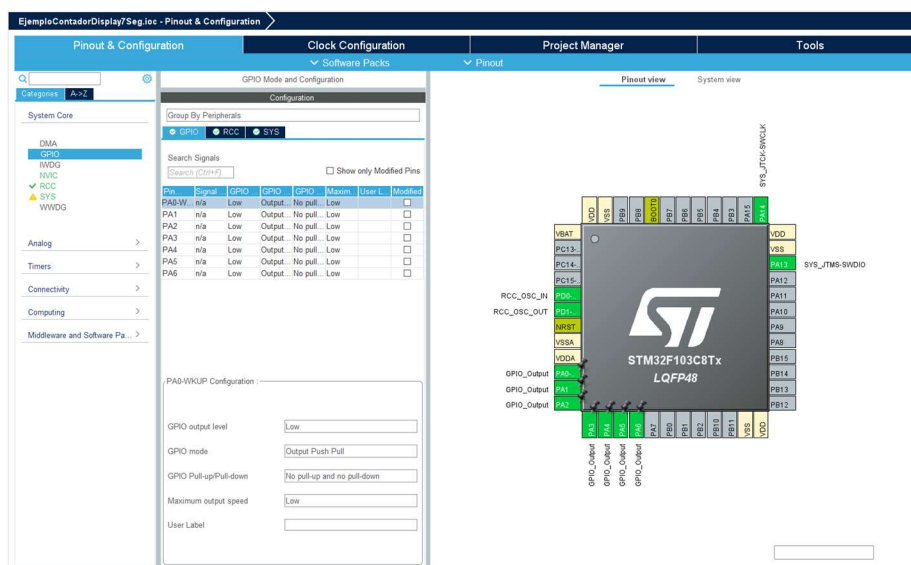


Para usar este dispositivo necesitamos 7 salidas del microcontrolador para controlar cada uno de sus segmentos. Para desplegar los numero del 0 al 9 usamos la siguiente tabla.

CÁTODO COMUN		ÁNODO COMUN	
Numero	Codigo 7 seg. (HEX)	Numero	Codigo 7 seg. (HEX)
0	0x3F	0	0x40
1	0x06	1	0x79
2	0x5B	2	0x24
3	0x4F	3	0x30
4	0x66	4	0x19
5	0x6D	5	0x12
6	0x7D	6	0x02
7	0x47	7	0x38
8	0x7F	8	0x00
9	0x6F	9	0x10

4. Desarrollo y Procedimiento.

Se creará un proyecto en el STM32CubeIDE como se indicó anteriormente. La configuración queda como se indica en la siguiente figura.



La configuración de cada pin de salida es como se muestra a continuación.

PA0-WKUP Configuration :

GPIO output level	<input type="text" value="Low"/>
GPIO mode	<input type="text" value="Output Push Pull"/>
GPIO Pull-up/Pull-down	<input type="text" value="No pull-up and no pull-down"/>
Maximum output speed	<input type="text" value="Low"/>
User Label	<input type="text"/>

El código de la práctica es el siguiente. Recuerde que el siguiente código debe estar entre los comentarios “USER CODE BEGIN” y “USER CODE END”.

```
1 #include "main.h"
2
3 #define RETARDO    500
4
5 uint8_t conta;
6
7 void SystemClock_Config(void);
8 static void MX_GPIO_Init(void);
9
10 uint8_t desplegar(uint8_t dato){
11     uint8_t valor;
12
13     switch(dato){
14         case 0:
15             valor = 0x3F;        // 0
16             break;
17         case 1:
18             valor = 0x06;        // 1
19             break;
20         case 2:
21             valor = 0x5B;        // 2
22             break;
23         case 3:
24             valor = 0x4F;        // 3
25             break;
26         case 4:
27             valor = 0x66;        // 4
28             break;
29         case 5:
30             valor = 0x6D;        // 5
31             break;
32         case 6:
33             valor = 0x7D;        // 6
34             break;
35         case 7:
36             valor = 0x47;        // 7
37             break;
38         case 8:
39             valor = 0x7F;        // 8
40             break;
41         case 9:
42             valor = 0x6F;        // 9
43     }
44
45     return valor;
46 }
```

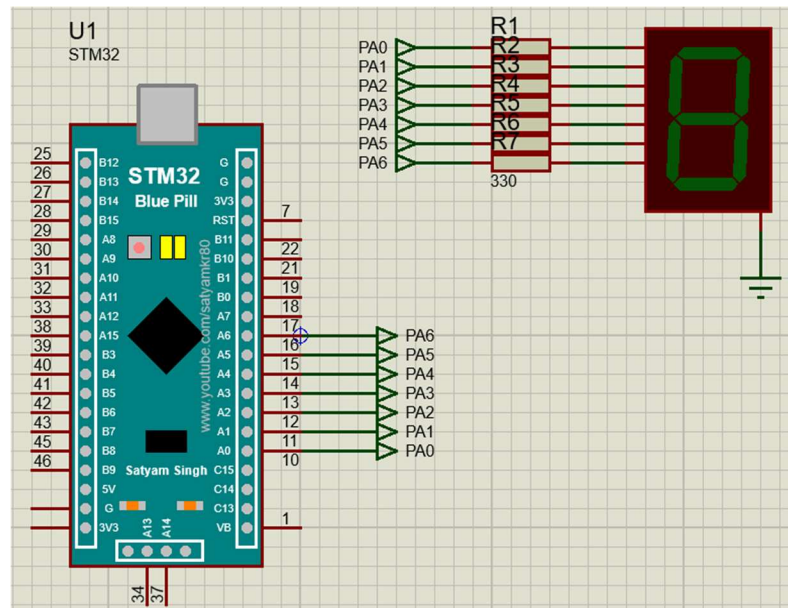
```

48 int main(void)
49 {
50     HAL_Init();
51
52     SystemClock_Config();
53
54     MX_GPIO_Init();
55     GPIOA->ODR = desplegar(conta);
56     HAL_Delay(RETARDO);
57
58     while (1)
59     {
60         conta++;
61         if(conta > 9)
62             conta = 0;
63         GPIOA->ODR = desplegar(conta);
64         HAL_Delay(RETARDO);
65     }
66 }

```

5. Esquemático del circuito.

El circuito de la práctica se muestra a continuación.





6. Mejora

Cambie la practica para que sea un contador ascendente y descendente.

7. Observaciones.

Esta sección es para que el alumno anote sus observaciones.

8. Conclusiones.

Esta sección es para que el alumno anote sus conclusiones.

9. Importante.

La práctica deberá ser validada en el salón de clases antes de anexar el reporte al manual de prácticas. Una vez validada realizar el reporte de practica como se anteriormente y anexar al manual de prácticas que se entregara a final del curso.