

MICROCONTROLADORES

Práctica No. 8. Contador con Botón, Display de 7 Segmentos y Arreglos.

1. Objetivo

- Utilizar un Display de 7 segmentos.
- Conectar un PUSHBUTTON en PB0.

2. Material y Equipo.

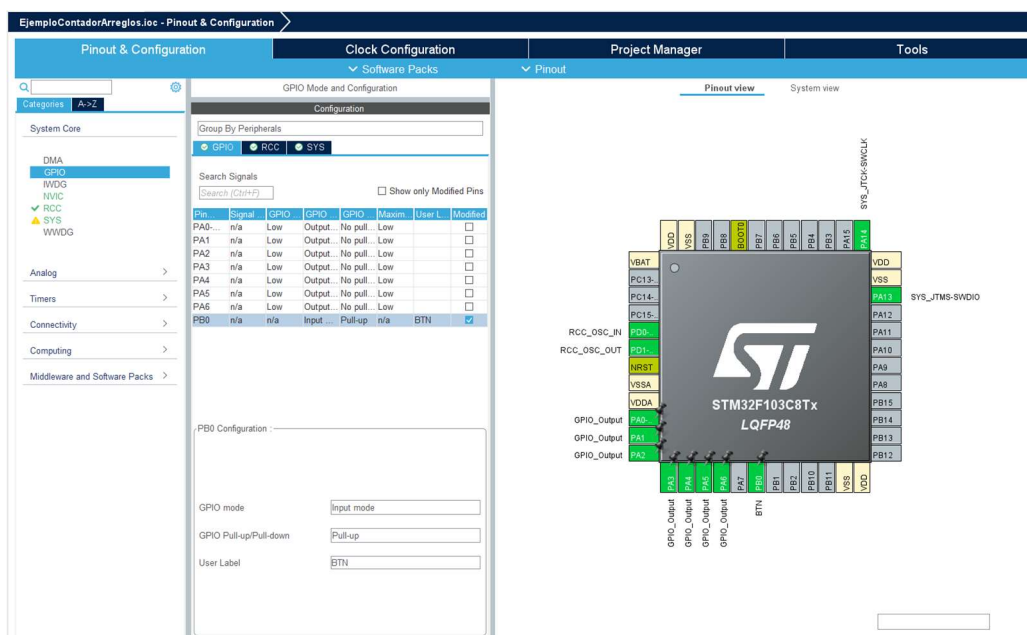
- Computador o laptop con el STM32CubeIDE.
- Un Display de 7 Segmentos de cátodo común.
- Siete resistencias de 330Ω o 220Ω .
- Un PUSHBUTTON.

3. Marco de Referencia.

En esta practica seguimos viendo las entradas y salidas usando un Display de 7 segmentos. En esta practica se usa un arreglo para simplificar el código de la practica anterior. Los arreglos son las estructuras de datos mas sencillas que tiene C. Hay que recordar que un arreglo es una estructura donde todos sus elementos son del mismo tipo y tienen un tamaño fijo este tamaño se le da a la hora de ser declarado.

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:

GPIO mode:

GPIO Pull-up/Pull-down:

Maximum output speed:

User Label:

La configuración del pin PB0 donde se conectará el botón es la siguiente.

PB0 Configuration :

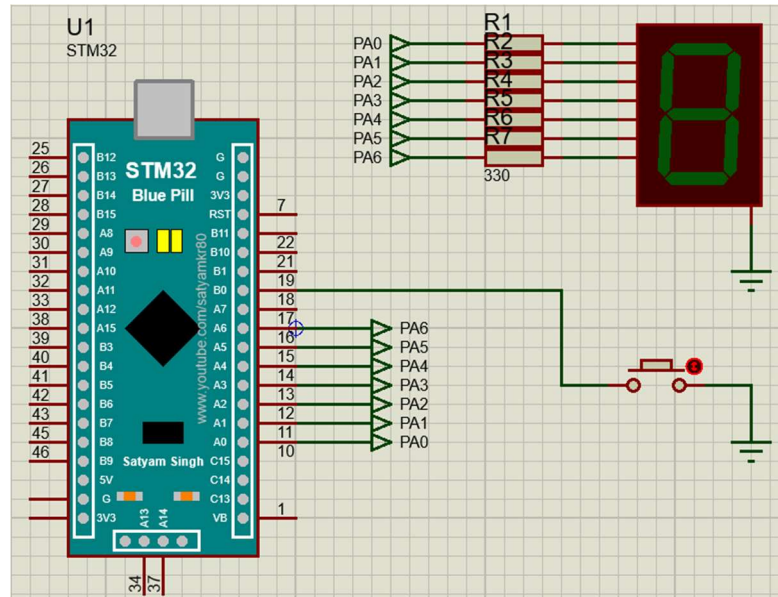
GPIO mode	<input type="text" value="Input mode"/>
GPIO Pull-up/Pull-down	<input type="text" value="Pull-up"/>
User Label	<input type="text" value="BTN"/>

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 uint8_t numeros[] = {0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x47, 0x7F, 0x6F};
4 uint8_t conta;
5
6 void SystemClock_Config(void);
7 static void MX_GPIO_Init(void);
8
9 int main(void)
10 {
11     HAL_Init();
12
13     SystemClock_Config();
14
15     MX_GPIO_Init();
16     GPIOA->ODR = numeros[conta];
17
18     while (1)
19     {
20         if(HAL_GPIO_ReadPin(BTN_GPIO_Port, BTN_Pin) == GPIO_PIN_RESET){
21             HAL_Delay(25);
22             conta++;
23             if(conta > 9)
24                 conta = 0;
25             GPIOA->ODR = numeros[conta];
26             while(HAL_GPIO_ReadPin(BTN_GPIO_Port, BTN_Pin) == GPIO_PIN_RESET);
27             HAL_Delay(25);
28         }
29     }
30 }
```

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 conectándole un botón en el pin PB1 para decrementar el contador y al llegar a cero y presionar otra vez el botón de PB1 el contador ira a nueve.

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 práctica como se anteriormente y anexar al manual de prácticas que se entregará a final del curso.