

## MICROCONTROLADORES

### Práctica No. 6. Contador Binario con Botón de Incremento.

#### 1. Objetivo

- Eliminar los rebotes mecánicos en los botones.
- Conectar 8 LEDs al puerto A y un PUSHBUTTON conectado al PB0.

#### 2. Material y Equipo.

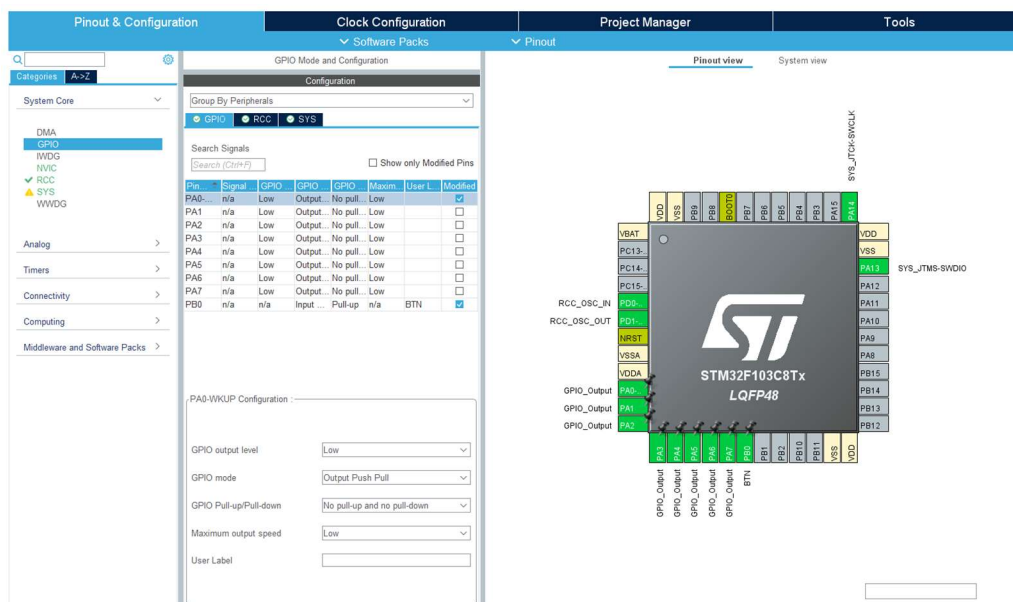
- Computador o laptop con el STM32CubeIDE.
- Ocho leds de 5mm o 3mm.
- Ocho resistencias de  $330\Omega$  o  $220\Omega$ .
- Un PUSHBUTTON.

#### 3. Marco de Referencia

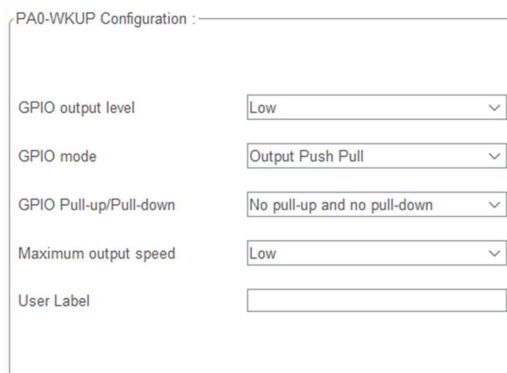
Los rebotes mecánicos son comunes en los push buttons y esto se debe al contacto que hacen las placas metálicas del botón cuando lo presionamos y después cuando dejamos de presionarlo, en el cambio de estado de alto a bajo o bajo a alto no se presenta un único cambio si no que hay varias transiciones que el microcontrolador detecta. Para eliminar este problema podemos hacerlo por software o hardware. Por hardware es conectar una resistencia y un capacitor a la entrada donde va conectado el botón y por software es utilizando retardos.

## 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.



La configuración del pin de entrada donde conectamos el botón es la siguiente.

PB0 Configuration :

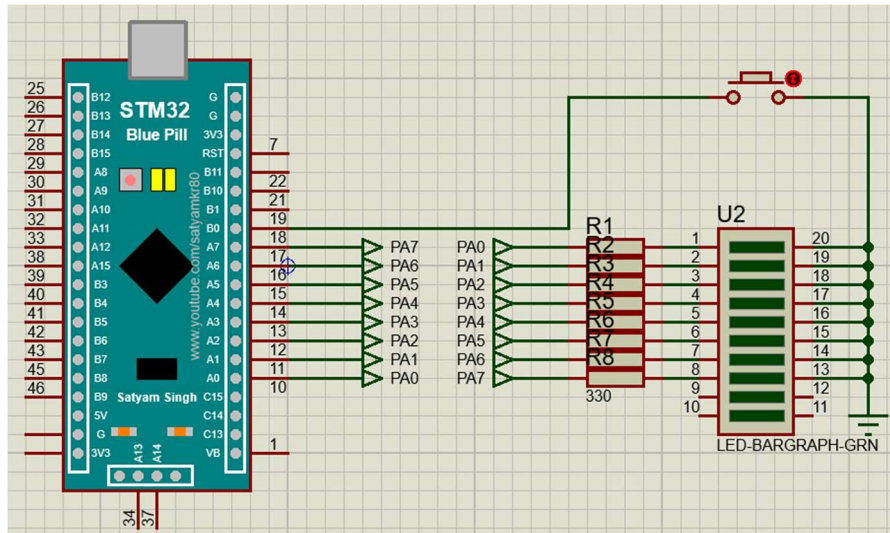
GPIO mode	<input type="text" value="Input mode"/>
GPIO Pull-up/Pull-down	<input type="text" value="Pull-up"/>
User Label	<input type="text" value="BTN"/>

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 conta;
4
5  void SystemClock_Config(void);
6  static void MX_GPIO_Init(void);
7
8  int main(void)
9  {
10     HAL_Init();
11
12     SystemClock_Config();
13
14     MX_GPIO_Init();
15
16     GPIOA->ODR = 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             GPIOA->ODR = conta;
24             while(HAL_GPIO_ReadPin(BTN_GPIO_Port, BTN_Pin) == GPIO_PIN_RESET);
25         }
26     }
27 }
```

## 5. Esquemático del circuito.

El circuito de la práctica se muestra a continuación.



## 6. Mejora

Agregar otro PUSHBUTTON al PB1. Con el botón en PB0 se incrementa hasta llegar a 255. Cuando se presione el botón en PB1 la cuenta será descendente hasta cero.

## 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.