

## **MICROCONTROLADORES**

### **Práctica No. 10. Contador con Dos Displays de 7 segmentos usando un 74HC595.**

#### **1. Objetivo**

- Utilizar dos 74HC595 y dos Displays de 7 segmentos.

#### **2. Material y Equipo.**

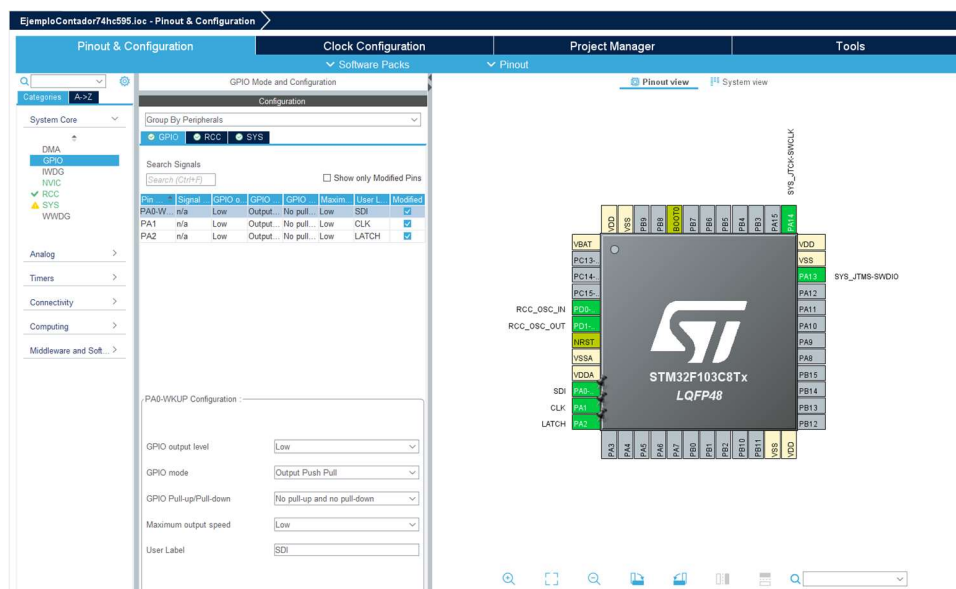
- Computador o laptop con el STM32CubeIDE.
- Dos 74HC595.
- Un Display doble o cuádruple de 7 Segmentos de cátodo o ánodo común.
- Siete resistencias de  $330\Omega$  o  $220\Omega$ .

#### **3. Marco de Referencia.**

En esta práctica aprenderemos como multiplexar dos displays de siete segmentos usando dos 74HC595. La idea principal es usar las líneas de control de cada Display (pin central) encendiendo y apagando cada Display a un intervalo de tiempo constante y que no es perceptible por el ojo humano y vemos ambos displays fijos y no notamos el parpadeo de alta velocidad que se hace cuando actualizamos cada Display individualmente.

## 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 los pines SDI, CLK y LATCH es la misma solo cambia la etiqueta. La siguiente figura muestra la configuración.

PA0-WKUP Configuration :

GPIO output level:

GPIO mode:

GPIO Pull-up/Pull-down:

Maximum output speed:

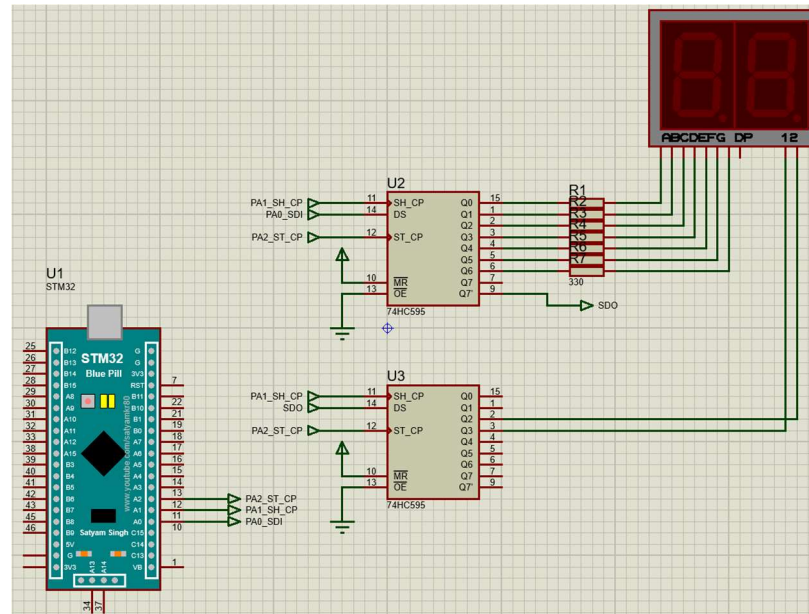
User Label:

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 #include "lib74hc595.h"
3
4 // arreglo para convertir un dato binario a 7 segmentos para un display de catodo comun.
5 uint8_t numeros[] = {0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x47, 0x7F, 0x6F};
6 uint8_t conta;
7
8 void SystemClock_Config(void);
9 static void MX_GPIO_Init(void);
10
11 // funcion para separar decenas y unidades y enviarlas a un display doble.
12 //
13 void send2Displays(uint16_t datoDisplay){
14     uint8_t datos[2];
15     uint8_t unidad, decena;
16
17     decena = datoDisplay / 10;
18     datos[0] = 0x04;
19     datos[1] = numeros[decena];
20     display_led(datos, 2);
21     HAL_Delay(1);
22     unidad = datoDisplay % 10;
23     datos[0] = 0x08;
24     datos[1] = numeros[unidad];
25     display_led(datos, 2);
26     HAL_Delay(1);
27 }
28
29 int main(void)
30 {
31     HAL_Init();
32
33     SystemClock_Config();
34
35     MX_GPIO_Init();
36
37     while (1)
38     {
39         for(uint8_t i = 0; i < 50; i++){
40             send2Displays(conta);
41         }
42         conta++;
43         if(conta > 99)
44             conta = 0;
45     }
46 }
```

## 5. Esquemático del circuito.

El circuito de la práctica se muestra a continuación.



## 6. Mejora

Cambie la práctica para que sea un contador de 0 a 9999.

## 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.