MICROCONTROLADORES

Práctica No. 23. BUS SPI.

1. Objetivo

- Uso del módulo SPI del STM32F103.

2. Material y Equipo.

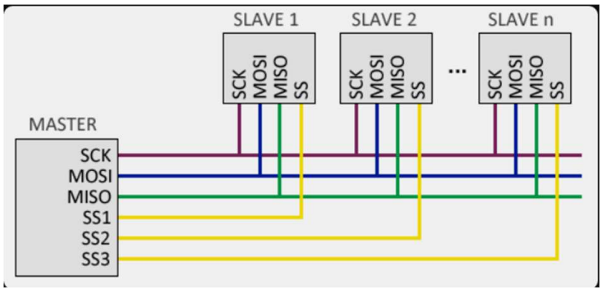
- Computador o laptop con STM32CubeIDE.
- Una barra de cuatro matrices de LEDS 8x8 controladas por el MAX7219.
- Un sensor LM35.

3. Marco de Referencia.

El bus SPI fue desarrollado por Motorola en la década de los 80. Este bus esta orientado a una conexión Maestro – Esclavo, donde el maestro (el STM32F103) controla las comunicaciones con los esclavos a través de 3 o 4 líneas de control. Estas líneas de control son SDI (Serial Data In), SDO (Serial Data Out), SCK (Serial Clock) y CS (Chip Select), en alguna documentación el SDI también se le conoce como MOSI (Master Out – Slave In), el SDO como MISO (Master In – Slave Out), SCK como SCLK (Serial Clock) y el CS como SS (Slave Slect).

En el bus SPI la comunicación es iniciada por el maestro ya que el es el encargado de controlar y enviar la señal de reloj al esclavo seleccionado, esta selección se realiza a través de la línea CS, el maestro solo puede habilitar un esclavo a la vez con la línea CS ya que los dispositivos esclavos no cuentan con una dirección única que serviría para direccionar a diferentes

esclavos (como el I^2C). La siguiente figura ilustra la conexión entre maestro y esclavo (o esclavos).



Si solo se tiene un esclavo conectado en el bus se puede usar el bus de 3 líneas sin usar el CS y mantener el CS del esclavo permanentemente a cero (GND).

Para iniciar una comunicación el maestro debe seleccionar al esclavo (inicialmente el CS debe estar en alto) bajando la línea de CS, después de habilitar al esclavo el maestro ya puede enviar datos (escribir) o recibir datos (leer) desde el esclavo. El SPI cuenta con un registro de corrimientos encargado de sacar el dato y recibir el dato serialmente, cuando se escribe al esclavo el maestro enviar bit a bit a través de su línea SDO al esclavo empezando por su bit mas significativo y puede recibir al mismo tiempo datos seriales por su línea SDI a este tipo de comunicación se le conoce como Full – Duplex (enviar y recibir al mismo tiempo) y también soporta comunicación Half – Duplex (primero enviar y después de enviar empezar a recibir enviando un byte especial llamado Dummy – Byte).

En el SPI existen cuatro formas de operación del bus, el modo se elije dependiendo de las especificaciones del dispositivo esclavo ya que el maestro y el esclavo deben coincidir en el modo de operación del bus. También debe considerarse la velocidad de comunicación del bus, esta velocidad se controla a través de la seña de reloj del SPI la velocidad máxima puede variar entre los 80MHz y llegar incluso a los 100MHz mucho más rápido que el bus I^2C, solo se tiene que considerar que a mayor velocidad las líneas de comunicación (SDI y SDO) pueden presentar capacitancias parasitas.

Los modos de operación del SPI se muestran en la siguiente tabla.

SPI mode	Clock polarity (CPOL)	Clock phase (CPHA)	Data is shifted out on	Data is sampled on
0	0	0	falling SCLK, and when $\overline{SS}$ activates	rising SCLK
1	0	1	rising SCLK	falling SCLK
2	1	0	rising SCLK, and when $\overline{SS}$ activates	falling SCLK
3	1	1	falling SCLK	rising SCLK

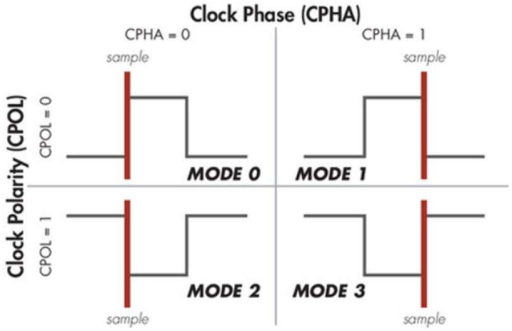
En el modo 0 el dato es transferido en el flanco de bajada y es leído en el flanco de subida de la señal de reloj mientras que el estado inactivo del reloj es en bajo.

El modo 1 el dato es transferido en el flanco de subida y es leído en el flanco de bajada de la señal de reloj mientras que el estado inactivo del reloj es en bajo.

El modo 2 el dato es transferido en el flanco de subida y es leído en el flanco de bajada de la señal de reloj mientras que el estado inactivo del reloj es en alto.

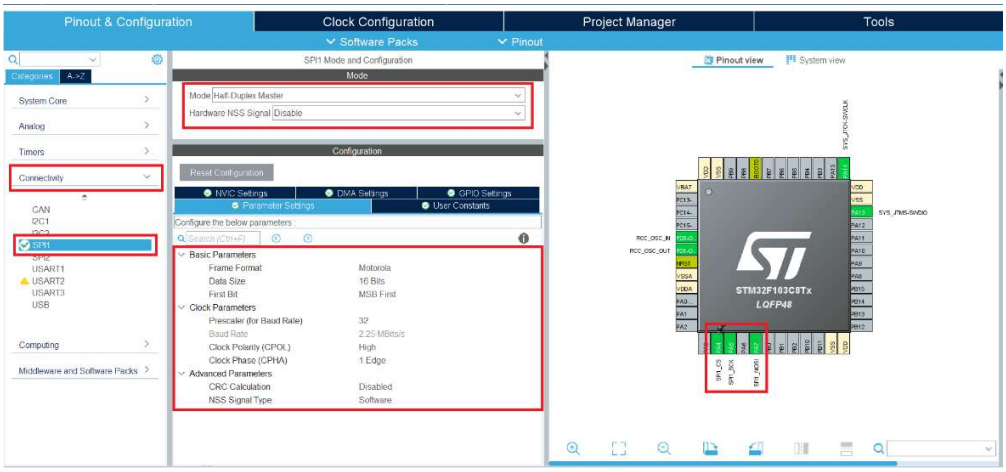
El modo 3 el dato es transferido en el flanco de bajada y es leído en el flanco de subida de la señal de reloj mientras que el estado inactivo del reloj es en alto.

La siguiente figura muestra la fase (CPHA) y polaridad (CPOL) de la señal de reloj.

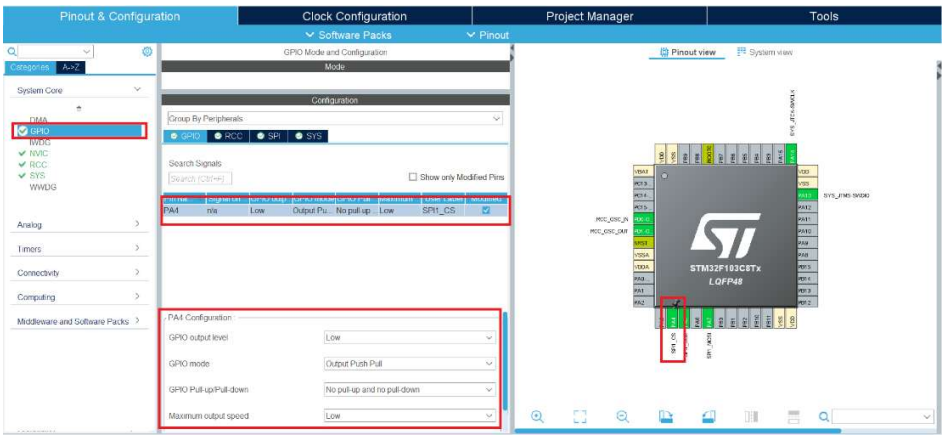


4. Desarrollo y Procedimiento.

Se creará un proyecto en el STM32CubeIDE como se indicó anteriormente.



Configuramos el pin PA4 como “Chip Select (CS)” y le asignamos el nombre de “SPI1_CS”.

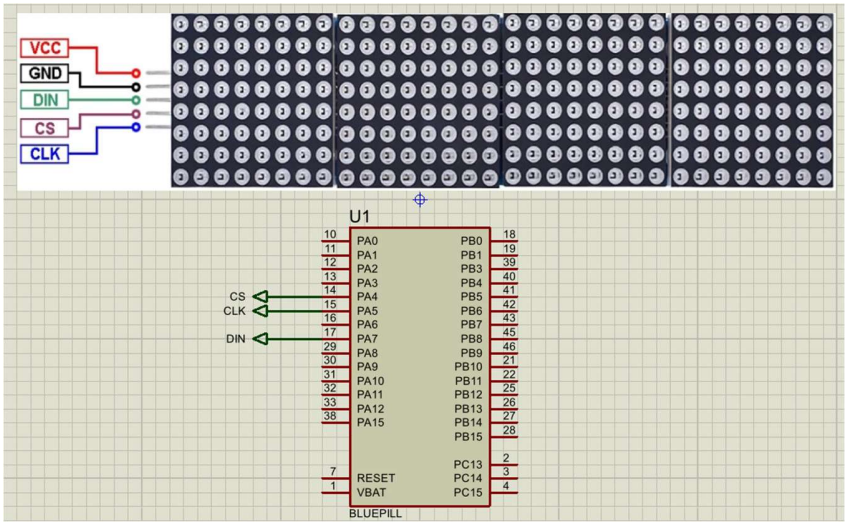


El código de la práctica es el siguiente en el archivo main.c.

```c
#include "main.h"
#include "max7219_matrix.h"

SPI_HandleTypeDef hspi1;

void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_SPI1_Init(void);

int main(void)
{
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();
    MX_SPI1_Init();
    matrixInit();
    clearDisplay();

    while (1)
    {               //1234567890123456789
        scrollString("Hola Mundo!!!     ", 100);
    }
}
```

5. Esquemático del circuito.

El esquemático de la práctica se muestra a continuación.



Nota: La barra de LEDs debe alimentarse externamente con una fuente de 5V ya que el ST-Link no proporciona la suficiente corriente para las matrices de leds.

6. Mejora.

Realice el cambio para desplegar la temperatura usando el sensor LM35.

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