

## STM32 SPI



**Objectifs :** Mises en œuvre du bus SPI (Serial Peripheral Interface)

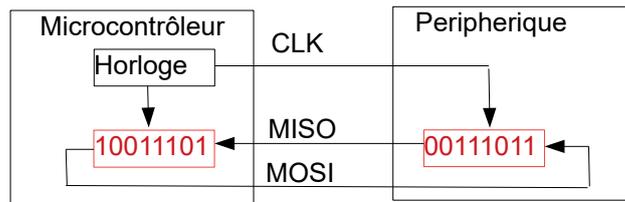
**Matériel :** Ce TP utilise une NUCLEO-F411RE, mais n'importe quelle autre carte NUCLEO convient.  
**Logiciel :** MBED

Le microcontrôleur STM32 dispose de bus de communication synchrones SPI et I2C.  
Ce TP propose la mise en œuvre du bus SPI.

Le bus SPI a été développé par Motorola dans les années 1980. Il permet l'établissement d'une communication série synchrone et full duplex entre deux terminaux, un maître et un esclave (Master-Slave), généralement sur un même circuit imprimé.

**Synchrone :** les deux terminaux possèdent la même horloge de communication qui est émise par le maître.  
**Full duplex :** comme pour l'UART, le SPI peut émettre et recevoir simultanément.

Le maître et l'esclave possèdent tous les deux un registre à décalage 8 bits. Ayant la même horloge et étant câbler comme ci-dessous, les deux registre sont échangés en 8 coups d'horloge.



**CLK :** Horloge

**MISO :** Master In Slave Out.

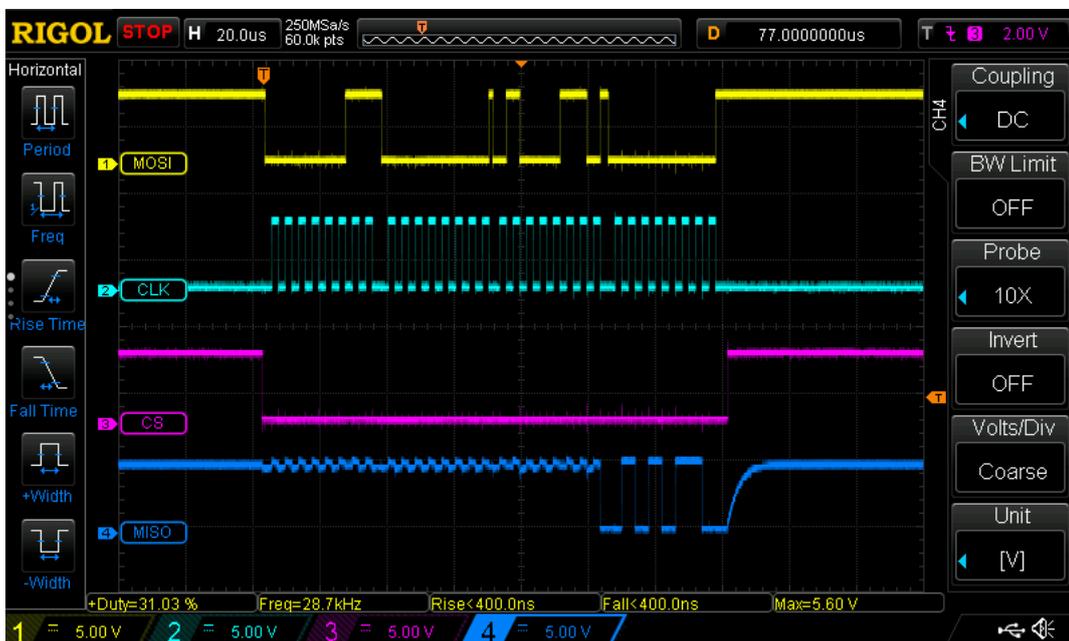
**MOSI :** Master Out Slave In.

Un signal de sélection **SS (Slave Select)** permet de sélectionner l'esclave avec lequel le maître souhaite communiquer.

Il y a quatre mode de configuration de l'horloge, grâce à deux bits , CPOL (Clock Polarity) et CPHA (Clock Phase) suivant le front actif et l'état de repos.

Pour en savoir plus : [https://fr.wikipedia.org/wiki/Serial\\_Peripheral\\_Interface](https://fr.wikipedia.org/wiki/Serial_Peripheral_Interface) (attention, préférer la version anglaise de l'article, beaucoup plus complète que la version française).

**Trames SPI** ( mansfield-devine.com )

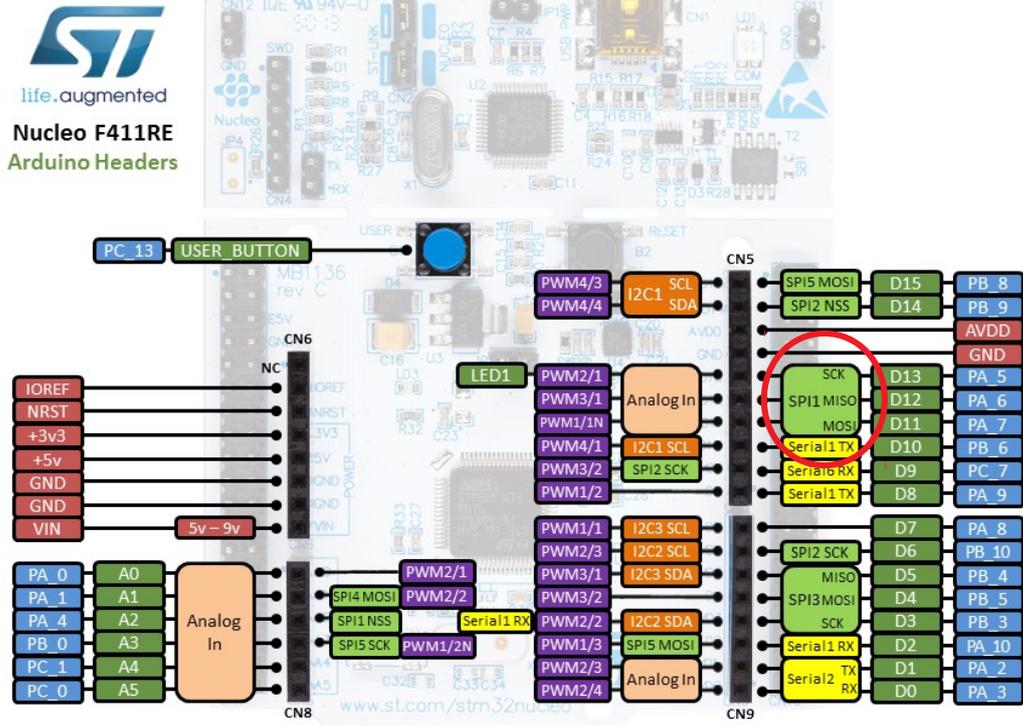


# STM32 SPI



La carte Nucleo STM32F411RE dispose de cinq bus SPI sur ses connecteurs Arduino. Dans ce TP nous utiliserons le SPI1.

Le TP



propose la mise en œuvre d'un afficheur 7 segments 6 digits « **Serial 7-Segments 8 digit** » **MIKROELEKTRONIKA**

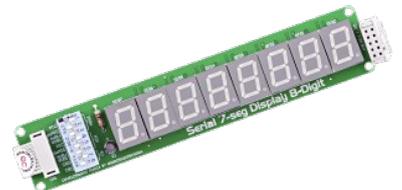
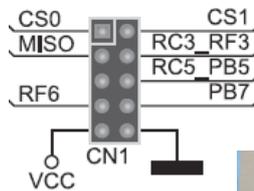
<https://www.mikroe.com/serial-7-seg-8-digit-board>

équipée d'un **MAX7219**.

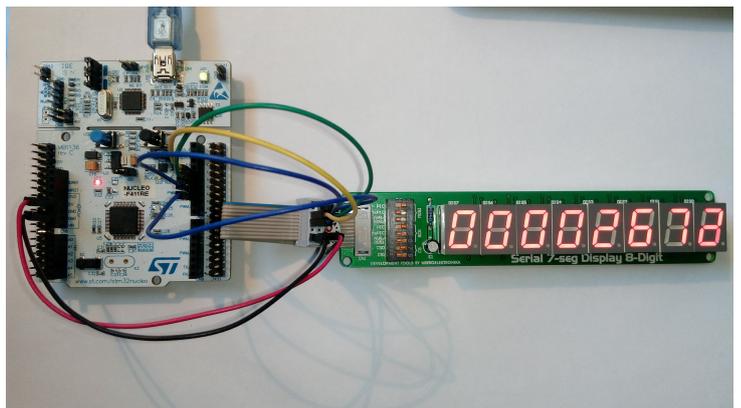
<https://www.maximintegrated.com/en/products/power/display-power-control/MAX7219.html>

Connexions à réaliser 7SEG-Nucleo: VCC=5v

Attention, vue du connecteur mâle.



- STM32 MISO sur RC5\_PB5 (MOSI)
- STM32 MOSI sur MISO
- STM32 SCK sur RC3\_RF3 (SCK)
- STM32 CS sur CS0
- 5v
- GND



MBED propose une bibliothèque de gestion du MAX7219.



<https://os.mbed.com/teams/Maxim-Integrated/code/MAX7219/>

Créer un nouveau programme de type "mbed-os-example-blinky"  
 Importer la bibliothèque du MAX7219 dans ce programme.

Remplacer main.c par le programme ci dessous

```

/* C.Dupaty
 * 03-2018
 * Demo MAX7219 library on MIKROELEKTRONIKA Serial 7-Seg Display 8-Digit
 * https://www.mikroe.com/serial-7-seg-8-digit-board
 * https://www.maximintegrated.com/en/products/power/display-power-control/MAX7219.html
 * see library here https://os.mbed.com/teams/Maxim-Integrated/code/MAX7219/
 */

#include "mbed.h"
#include "max7219.h"

Max7219 max7219(MOSI, MISO, SCK, CS); // standard SPI on NUCLEO BOARD

// print decimal number on the display
void affN(unsigned int n)
{
    // 7 segments code for print numbers [0..9] see MAX7219 datasheet
    const unsigned char code7seg[]={0,0,0,0,0,0,0,0,0,0};
    int i = 1;
    while (n != 0)
    {
        max7219.write_digit(1, i++, code7seg[n % 10]);
        n /= 10;
    }
    while(i<9) max7219.write_digit(1, i++, code7seg[0]);
}

int main()
{
    int nb=0;
    max7219_configuration_t cfg = {
        .device_number = 1, // device number (only one in this demo)
        .decode_mode = 0, // mode 0, no decode (see MAX7219 datasheet page 7)
        .intensity = Max7219::MAX7219_INTENSITY_4, // intensity of light ( 1 to 8)
        .scan_limit = Max7219::MAX7219_SCAN_8 // nb of digits to print
    };
    max7219.init_device(cfg);
    max7219.enable_device(1);
    max7219.set_display_test(); // flash all segments for test
    wait(1);
    max7219.clear_display_test();
    // a simple 32 bits counter
    while (1) {
        afficher le nombre nb
        incrémenter nb
        si nb > 4000000000 remettre nb à 0
        attendre 100ms
    }
}
    
```



**Exercice 1 :**

- Remplacer le texte rouge sur fond jaune de manière à réaliser un compteur de 100mS.
- Rechercher les noms des broches MOSI,MISO,SCK et CS de votre NUCLEO
- A partir du datasheet du MAX7219 compléter les valeurs du tableau code7seg[]
- Compléter la boucle sans fin de manière à réaliser un compteur de 100mS
- A l'aide d'un oscilloscope ou d'un analyseur logique, relever et interpréter un échange de données

**Exercice 2 :**

Réaliser un chronomètre heure/minutes/secondes démarré et arrêté par le bouton bleu en interruption sur front descendant.