

Protocole SPI

Introduction

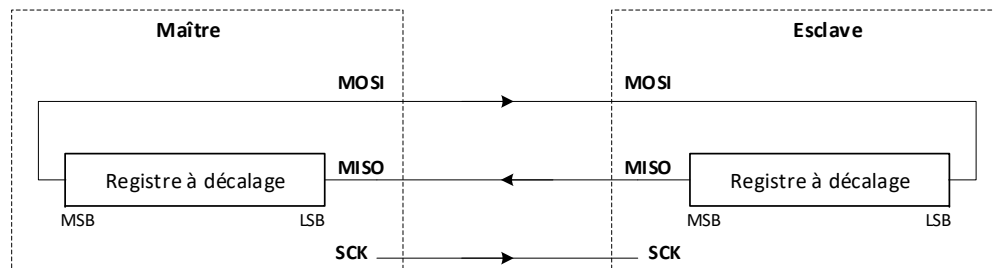
Le mode SPI permet la **transmission et la réception simultanée** (full duplex) de 8 bits de données de manière synchrone. Selon le protocole SPI, l'interconnexion du microcontrôleur avec un autre périphérique nécessite généralement, trois broches :

- Master Out Slave In (MOSI)
- Master In slave Out (MISO)
- Serial Clock (SCK)

En outre, une quatrième broche peut être utilisée en mode esclave :

- Slave Select ($\overline{SS}$).

La figure suivante illustre le schéma bloc simplifié d'une liaison entre le maître et l'esclave.



Les communications sont initiées par le maître. A chaque cycle d'horloge, l'envoi d'un bit par le maître, s'accompagne d'envoi d'un bit par l'esclave.

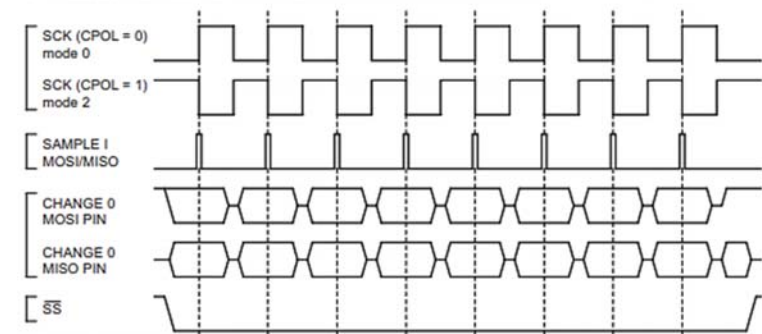
Modes de fonctionnement

Il existe quatre combinaisons de phase et de polarité d'horloge SCK par rapport aux données série, qui sont déterminées par les bits de contrôle CPHA et CPOL. Le tableau suivant résume les valeurs associées aux bits CPOL et CPHA pour chaque mode de fonctionnement.

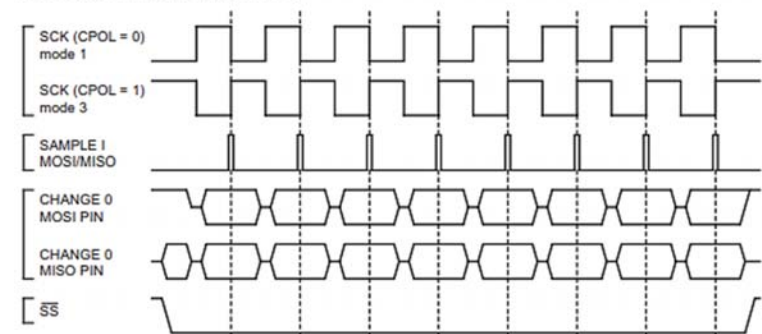
Mode SPI	Conditions	Front d'envoi	Front de lecture
0	CPOL=0, CPHA=0	descendant	montant
1	CPOL=0, CPHA=1	montant	descendant
2	CPOL=1, CPHA=0	montant	descendant
3	CPOL=1, CPHA=1	descendant	montant

Les formats de transfert de données SPI sont illustrés dans les figures suivantes. Les bits de données sont décalés et verrouillés sur les fronts opposés du signal SCK, ce qui garantit un temps suffisant pour que les signaux de données se stabilisent.

SPI Transfer Format with CPHA=0



SPI Transfer Format with CPHA=1



Librairie arduino

La librairie SPI (spi.h) comporte les fonctions suivantes :

SPI.setBitOrder(dataOrder) : définit le sens de transfert, c-à-d, le bit à transférer en premier MSB ou LSB. Ce paramètre peut être : **MSBFIRST** ou **LSBFIRST**.

SPI.setClockDivider(speedMaximun) : l'horloge de bus SPI est dérivée de la fréquence système. La fréquence du système peut être divisée par : 2, 4, 8, 16, 32, 64 ou 128. Le paramètre *speedMaximun* peut être : **SPI_CLOCK_DIV2**, **SPI_CLOCK_DIV4**, **SPI_CLOCK_DIV8**, **SPI_CLOCK_DIV16**, **SPI_CLOCK_DIV32**, **SPI_CLOCK_DIV64** ou **SPI_CLOCK_DIV128**.

SPI.setDataMode(dataMode) : définit le mode SPI. Ce paramètre peut être : **SPI_MODE0**, **SPI_MODE1**, **SPI_MODE2**, ou **SPI_MODE3**.

SPI.begin() : Initialise le bus SPI :

- SCK, MOSI et SS en sortie,
- SCK et MOSI au niveau bas,
- SS au niveau haut.

SPI.end() : Désactive le bus SPI (en laissant les modes de broche inchangés).

SPI.transfer(), **SPI.transfer16()** : Le transfert SPI est basé sur un envoi et une réception simultanée : les données reçues sont renvoyées dans *receiveVal* (ou *receiveVal16*). En cas de transfert de tampon, les données reçues sont stockées dans la mémoire tampon en place (les anciennes données sont remplacées par les données reçues).

receivedVal = SPI.transfer(val) ;

receivedVal16 = SPI.transfer16(val16) ;

SPI.transfer(buffer, size) ;