

Le bus SPI

1. INTRODUCTION

1.1. ORIGINE

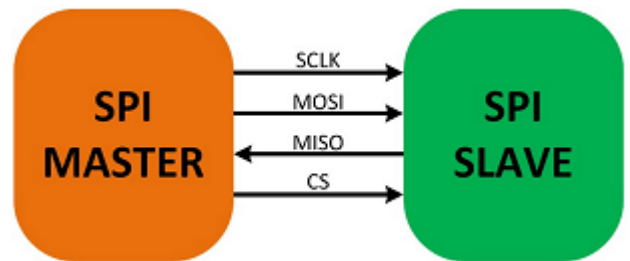
La sophistication et la complexité des appareils grand public ne permet plus l'échange entre composants électroniques par liaisons parallèles, ces liaisons devenaient de plus en plus nombreuses et encombrantes. Dans ces systèmes à base de microcontrôleurs (téléphones mobiles ...), les échanges de données entre les composants se font maintenant avec des bus série.

Dans le but de minimiser ces liaisons, et par conséquent d'en augmenter la fiabilité, a été créé le bus série SPI (Serial Peripheral Interface), initialement développé par Motorola (maintenant Freescale).

D'autres fabricants (Microchip, Atmel, Texas Instrument...) ont adopté pour ce type de liaison et de nombreux composants sont apparus (mémoires, capteurs, micro contrôleurs...).

1.2. PRESENTATION

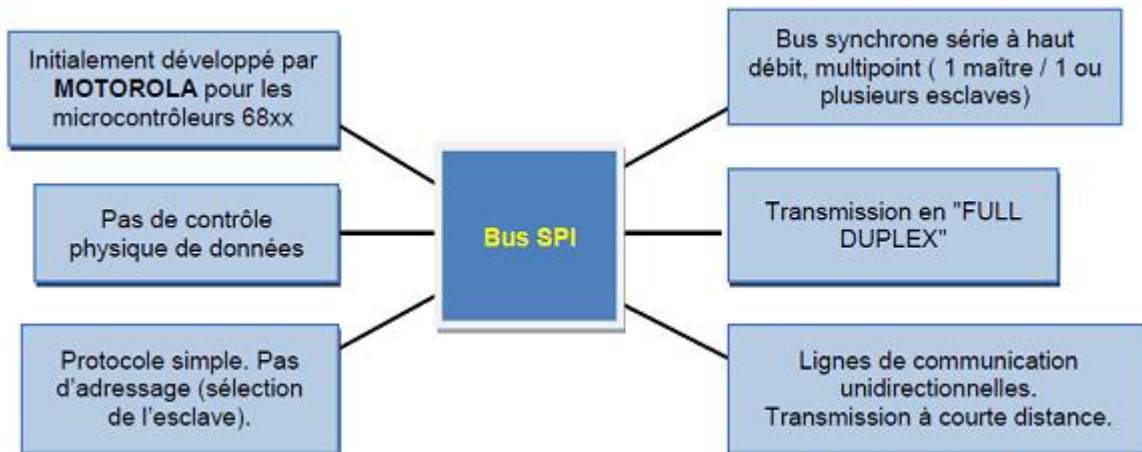
Une liaison SPI (**Serial Peripheral Interface**) est un bus de données série synchrone baptisé ainsi par Motorola, qui opère en mode full-duplex. Les circuits communiquent selon un schéma maître-esclave, où le maître s'occupe totalement de la communication. Plusieurs esclaves peuvent coexister sur un même bus, dans ce cas, la sélection du destinataire se fait par une ligne dédiée entre le maître et l'esclave appelée chip select.



L'appellation SPI concerne des signaux et des chronogrammes qui sont beaucoup moins bien normalisés que ceux du bus I2C, tout simplement parce que la liaison du bus SPI ne fait l'objet d'aucune norme officielle.

Ainsi certains circuits indiquent qu'ils sont munis d'une interface SPI, mais vous pourrez aussi trouver comme appellation « Microwire » (qui est une marque déposée de National Semiconductor) ou bien encore « bus série trois fils ». Un bus de ce type permet la connexion, sous forme série maître-esclave, de plusieurs circuits disposant d'interfaces compatibles, avec seulement trois fils de liaisons.

1.3. PHILOSOPHIE DU BUS SPI

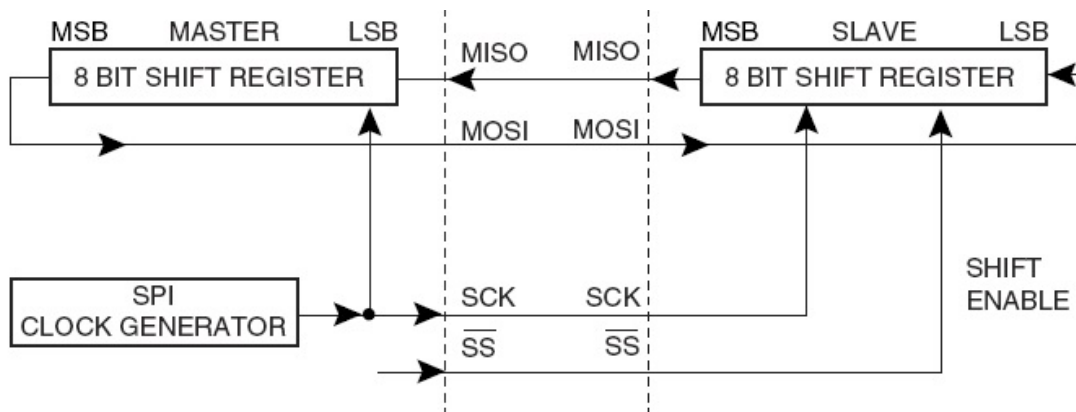


1.4. LES CHAMPS D'APPLICATION DU BUS SPI

Il est utilisé pour la communication rapide de données entre périphériques d'un appareil comme par exemple les mémoires, les systèmes d'affichage, carte SD, etc.

1.5. SCHEMA DE PRINCIPE

La figure ci-dessous correspond au schéma de principe simplifié d'une liaison de données SPI, avec ses principaux composants.



Deux registres à décalages sont généralement mis en œuvre. Ils peuvent être de type matériel ou logiciel, selon les dispositifs employés. Quelle que soit leur mise en œuvre, ces deux registres à décalage forment un buffer circulaire intercomposant, qui se trouve au cœur de la liaison SPI.

Les communications sont initiées par le maître, qui commence par sélectionner l'esclave. Au cours de chaque cycle d'horloge, le maître envoie un bit à l'esclave, qui le lit sur la ligne MOSI.

En parallèle, l'esclave envoie un bit au maître, qui le lit sur la ligne MISO. Ces lectures et écritures simultanées entre le maître et l'esclave forment une communication Full-Duplex.

2. PROPRIETES PHYSIQUES DU BUS SPI

2.1. LE SUPPORT PHYSIQUE

2.1.1. Données

Les données échangées sont des octets. La transmission s'effectue sur 2 fils monodirectionnels MOSI et MISO :

MOSI : Master Output Slave Input, pilotée par le maître (autre dénomination **SDO Serial Data Out**)

MISO : Master Input Slave Output, pilotée par l'esclave (autre dénomination **SDI Serial Data In**)

2.1.2. Horloge

Une horloge indépendante fixée par le maître synchronise les échanges (en général sur front).

SCLK : SPI Serial Clock, ligne d'horloge pilotée par le maître (parfois appelée **SCK** aussi).

La fréquence de l'horloge de transmission peut atteindre 20 MHz (selon les performances des circuits reliés au bus).

2.1.3. Sélection de boîtier

Il n'y a pas d'adressage des esclaves (comme sur un bus I2C par exemple). L'esclave devient actif au moyen d'une ligne de sélection de boîtier dédiée (généralement active à l'état bas).

/CS : Chip Select, sélection de l'esclave pilotée par le maître (autre dénomination **/SS Slave Select**).

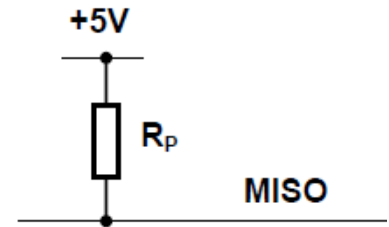
La ligne est donc constituée de 3 fils auxquels il faut ajouter les fils de sélection d'esclave.

2.1.4. Le bus complet

_____	SCLK	SCK (Serial Clock) : Horloge du bus (produite par le maître)
_____	MOSI	MOSI (Master Out Slave In) : Données du maître vers l'esclave actif
_____	MISO	MISO (Master In Slave Out) : Données de l'esclave actif vers le maître
_____	SSn	SSn (Slave Select n) : Sélection de l'esclave n à destination de la transmission

Lorsque le bus est inutilisé, ce qui revient à dire qu'aucun esclave n'est sélectionné, la ligne MISO est à l'état haute impédance, ce qui ne permet pas d'en définir l'état logique.

On évite cela par l'utilisation d'une résistance de tirage, de 5 à 50 kOhms, qui n'a aucune influence sur la vitesse de transmission (contrairement à ce qui se passe pour un bus i2C).

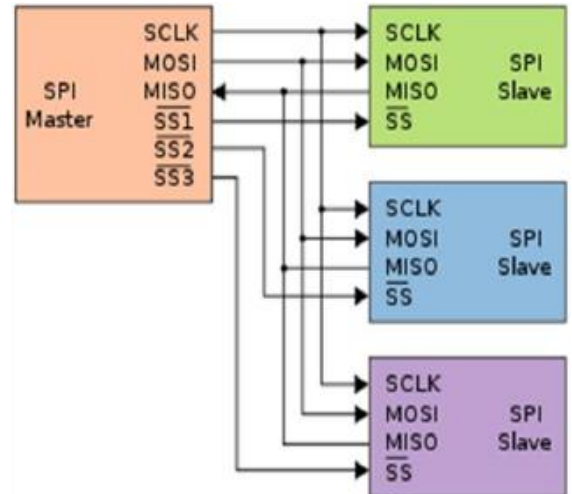


2.2.SYNOPTIQUE D'UNE LIAISON SPI MAITRE-MULTI-ESCLAVES

2.2.1. Cas général

Le maître sélectionne un seul et unique esclave avec lequel il veut rentrer en communication par la mise au niveau logique 0 de /SS1, /SS2 ou /SS3, puis, après 8 fronts d'horloge, l'octet de donnée est transféré.

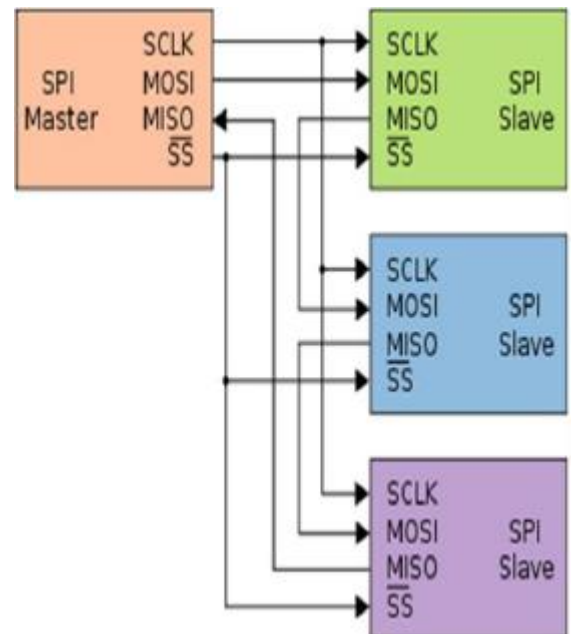
La seule limite à la quantité d'esclaves est en fait le nombre de broches SS gérées par le maître.



2.2.2. Cas de la daisy chain (esclaves en guirlande, en cascade...)

Le maître sélectionne tous les esclaves par la mise au niveau logique 0 de /SS. Il enverra à la queue leu leu les données du premier esclave, du deuxième et enfin du troisième.

Cette disposition permet de réduire le nombre de lignes /SS, mais en contrepartie il faudra un "buffer" plus grand dans le maître (ou une gestion du soft plus élaborée).



2.3.CHRONOGRAMME D'UNE LIAISON SPI MAITRE-ESCLAVE

La liaison SPI peut être effectuée suivant 4 modes différents : mode 0, 1, 2 ou 3.

Les modes sont paramétrés suivant 2 bits des registres de configuration du maître et de l'esclave. Pour une transmission correcte il faut bien évidemment que ces paramètres soient réglés de la même manière pour tous les composants reliés au bus.

Ces deux bits sont nommés **CPOL** (clock polarity) et **CPHA** (clock phase).

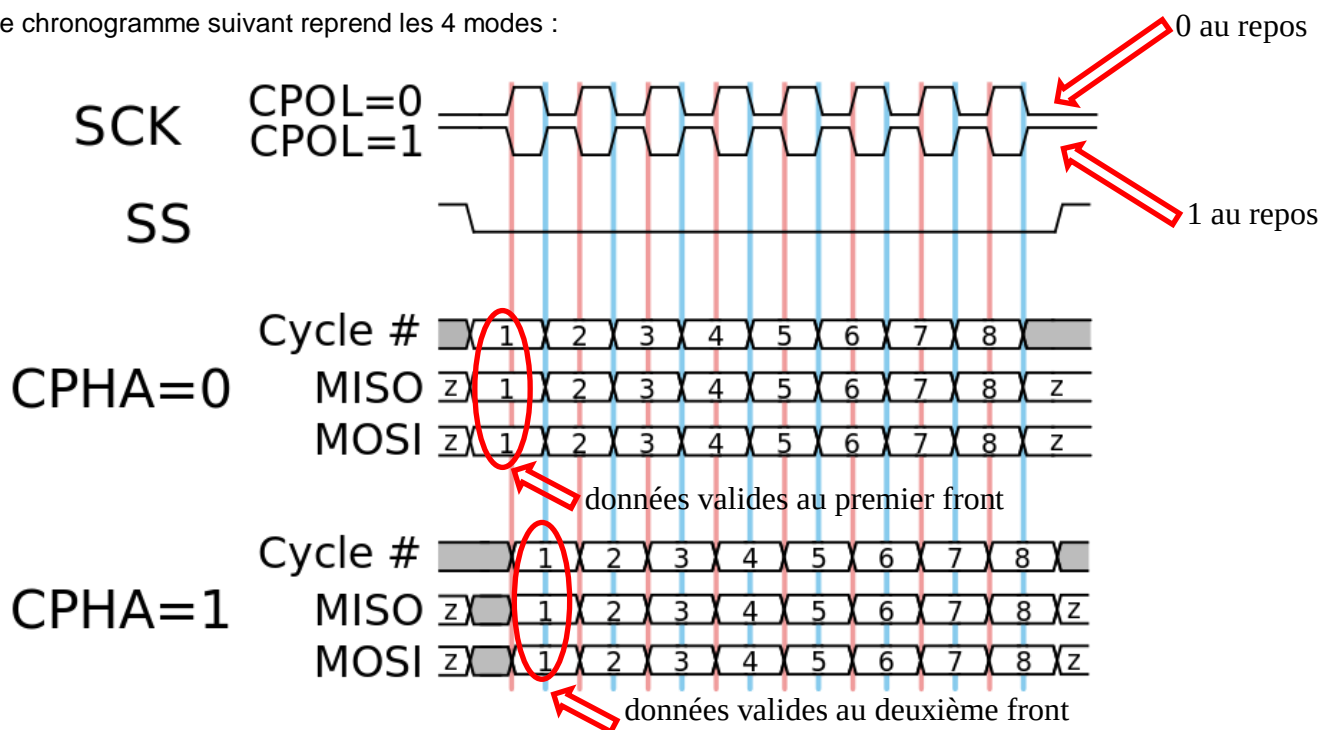
Le **CPOL** détermine si au repos l'horloge est au niveau BAS (CPOL=0) ou HAUT (CPOL=1).

Le **CPHA** détermine à quel front de l'horloge les données sont valides. CPHA=0 → les données sont valides au premier front d'horloge, CPHA=1 → elles sont valides au deuxième front.

Le tableau suivant résume l'utilisation de CPOL et CPHA :

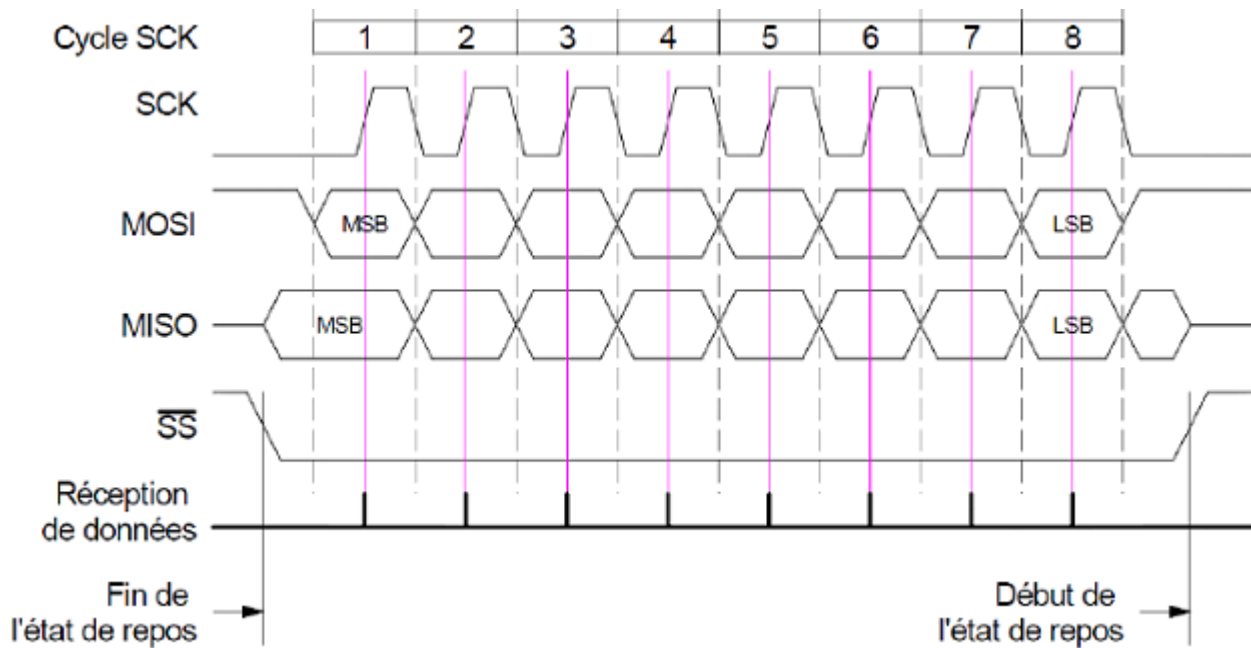
Mode SPI	CPOL	CPHA	SCK	Repos SCK	Donnée valide
0	0	0		0	1 ^{er} front
1	0	1		0	2 ^e front
2	1	0		1	1 ^{er} front
3	1	1		1	2 ^e front

Le chronogramme suivant reprend les 4 modes :



2.4. CHRONOGRAMME MODE 0

CPOL = 0 et CPHA = 0



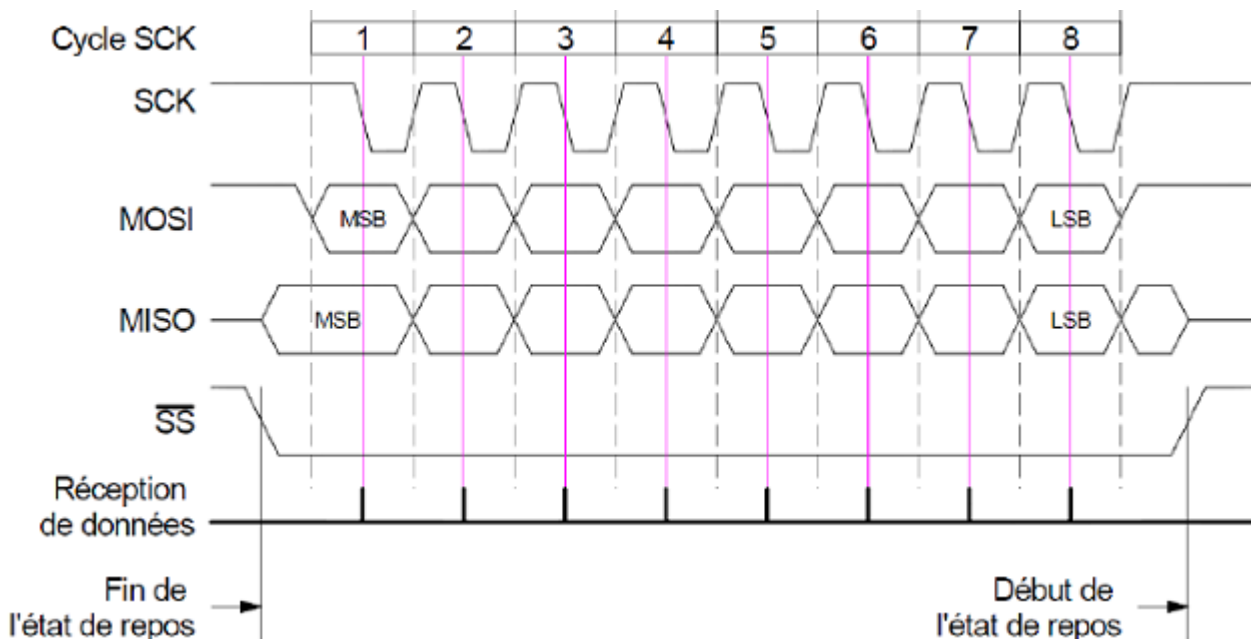
On utilise le principe du registre à décalage. Dans le cas ci-dessus, en 8 périodes d'horloge, l'octet passe du registre du maître à celui de l'esclave et réciproquement le contenu du registre d'esclave est passé dans celui du maître (full-duplex : simultanéité des transferts). Puisqu'il ne peut pas y avoir de collisions lors du transfert, il n'y a pas besoin d'arbitrage.

Comme CPHA = 0, les données sont valides au premier front du signal d'horloge.

La polarité CPOL détermine qu'il s'agit d'un front montant. En effet, pour CPOL=0, au repos, l'horloge est au niveau BAS, le premier front est donc un front montant.

2.5. CHRONOGRAMME MODE 2

CPOL = 1 et CPHA = 0

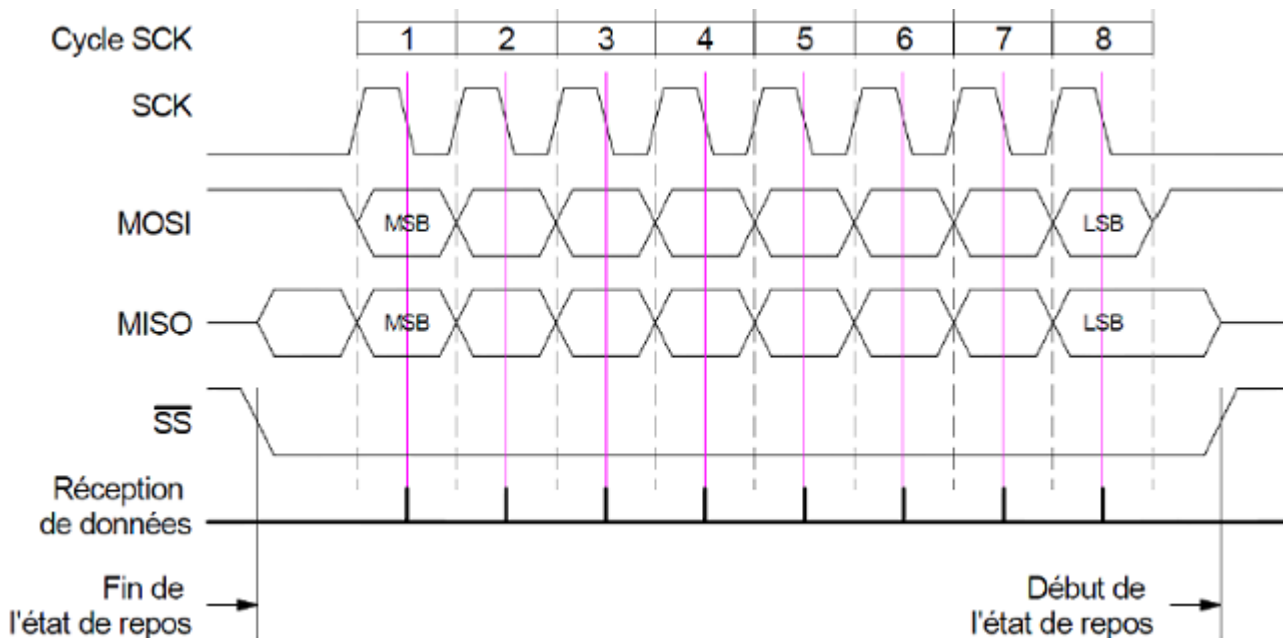


Comme CPHA = 0, les données sont toujours valides au premier front du signal d'horloge. La polarité CPOL détermine s'il s'agit d'un front montant ou descendant.

Comme CPOL=1, au repos, l'horloge est au niveau HAUT, le premier front est donc un front descendant.

2.6.CHRONOGRAMME MODE 1

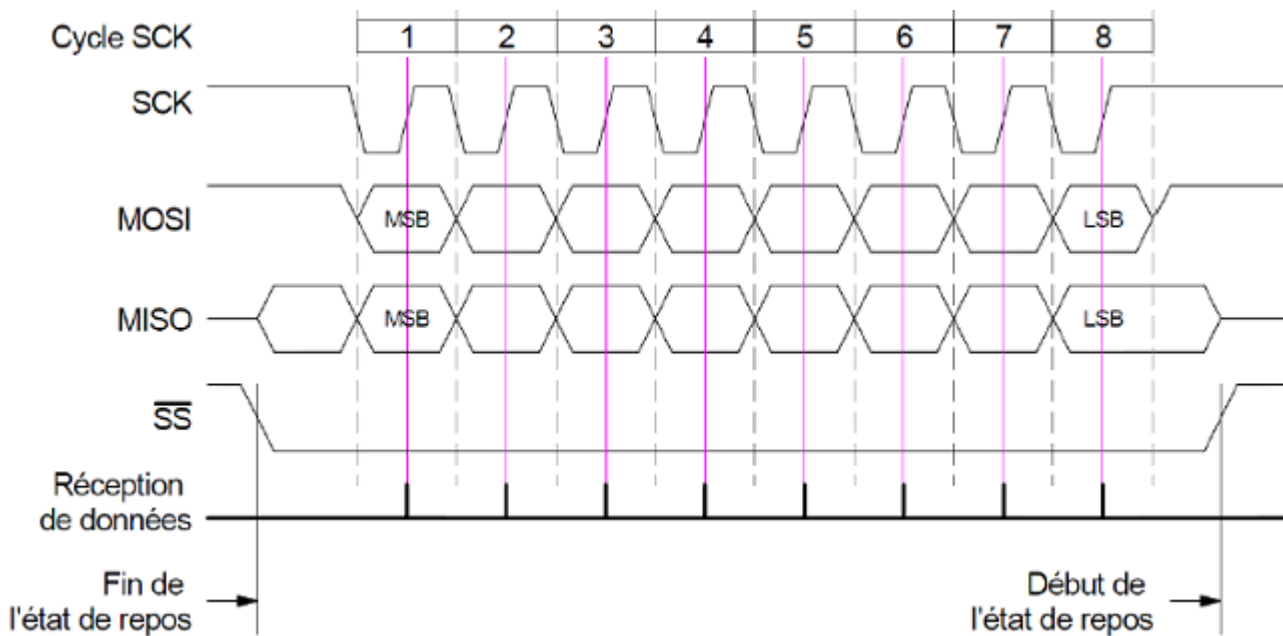
CPOL = 0 et CPHA = 1



Le niveau de repos de SCK est 0, les données sont valides sur le deuxième front qui est donc descendant.

2.7.CHRONOGRAMME MODE 3

CPOL = 1 et CPHA = 1



Le niveau de repos de SCK est 1, les données sont valides sur le deuxième front qui est donc montant.

2.8.AVANTAGES ET INCONVENIENTS DU BUS SPI

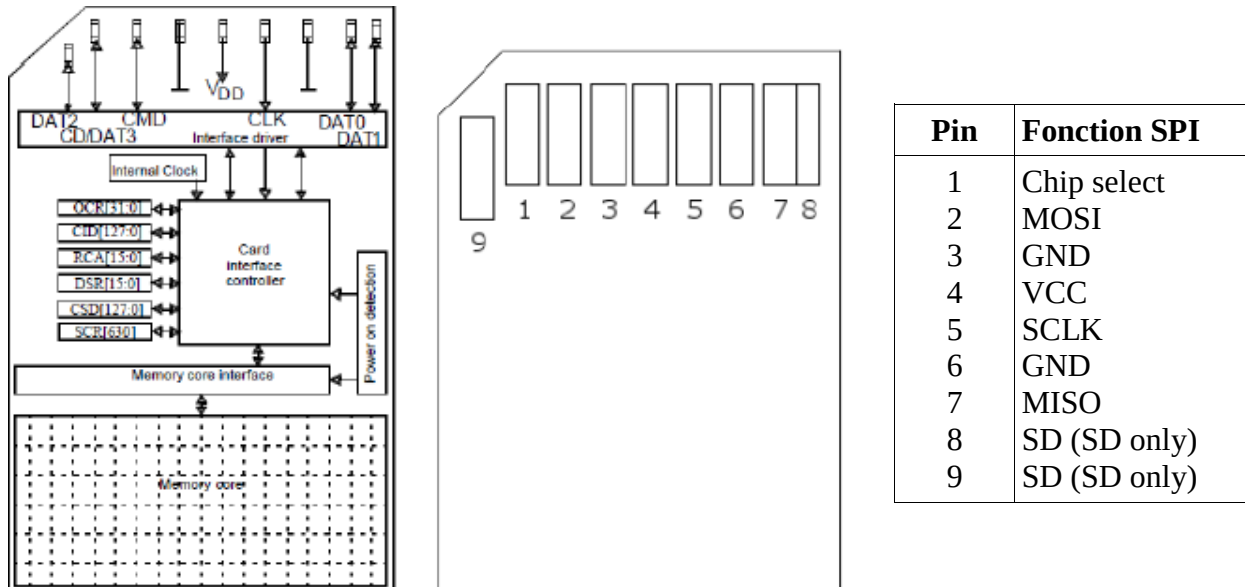
Avantages	Inconvénients
Communication en Full Duplex	Pas d'adressage possible
"Indépendant" du nombre de bits à transmettre	Utilisation sur très courte distance (même carte)
Pas de collision possible	Nécessite plus de fils que I ² C
Les esclaves utilisent l'horloge du maître, pas de problème de précision de quartz.	Pas d'acquittement (le maître ne sait pas s'il est écouté)
Beaucoup plus rapide que I ² C en mode standard	
Possibilité de configuration à plusieurs maîtres	

3. COMPLEMENTS

3.1.SD CARD

Les cartes SD possèdent plusieurs interfaçages possibles dont un bus SPI. Le schéma ci-dessous présente la connectique des cartes SD, on y retrouve les broches spécifiques du bus SPI.

Avec un microcontrôleur incluant une interface SPI spécifique et une programmation pour les routines supplémentaires d'accès à la mémoire de la carte, on peut développer des applications sur ce type de carte. Cependant, le protocole utilisé est complexe et, si l'on souhaite avoir une structure de fichier utilisable sur un ordinateur, on est confronté au problème de la FAT...



3.2.ARDUINO

La gestion d'une liaison SPI avec l'Arduino est facile à mettre en œuvre. Les micro-contrôleurs d'Atmel Atmega disposent d'une interface SPI dont les entrées/sorties se répartissent comme suit :

SCK sur 13, MISO sur 12, MOSI sur 11 et /SS sur 10 pour les Uno.

L'environnement de développement contient une bibliothèque, nommée SPI.h, qui permet une gestion logicielle très simple du protocole de dialogue des liaisons SPI.