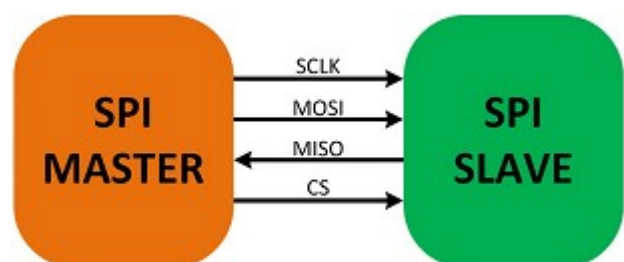


Bus SPI

Table des matières

1. Introduction.....	2
1.1. Origine.....	2
1.2. Présentation.....	2
1.3. Philosophie du bus SPI.....	3
2. Schéma de principe.....	3
3. Propriétés physiques du bus SPI.....	4
3.1. Le support physique.....	4
3.2. Synoptique d'une liaison SPI Maître-Esclave.....	5
3.3. Format de transfert de données.....	5
3.4. Synoptique d'une liaison SPI Maître-Multi-Esclaves.....	7
3.5. Bus SPI et niveaux électriques.....	8
4. Chronogramme.....	9
5. Exemple de mise en œuvre d'une communication.....	10
6. Les différents autres noms utilisés avec un bus SPI.....	11
7. Constitution d'un système Motorola.....	12
7.1. Le diagramme bloc SPI.....	12
7.2. Le registre SPCR: Serial Peripheral Control Register.....	13
7.3. Le registre SPSR: Serial Peripheral Status Register.....	13
7.4. Le registre SPDR: Serial Peripheral Data Register.....	14
7.5. Le système de détection d'erreur SPI.....	14
8. Constitution pour un système à microcontrôleur PIC.....	14
8.1. Le MSSP Control Register 1 (SSPCON1).....	14
8.2. Le MSSP Status Register (SSPSTAT).....	15
8.3. Le Serial Receive/Transmit Buffer Register SSPBUF.....	15
8.4. Le MSSP Shift Register.....	16
9. Compléments.....	16
9.1. SD card.....	16
9.2. Arduino.....	17

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-esclaves, 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.



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.

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. Présentation

Dans les circuits d'interface le bus SPI n'est pas seul est se trouve talonné depuis de nombreuses années par les circuits dotés d'une liaison série synchrone de type SPI. Cette appellation rencontre cependant 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.

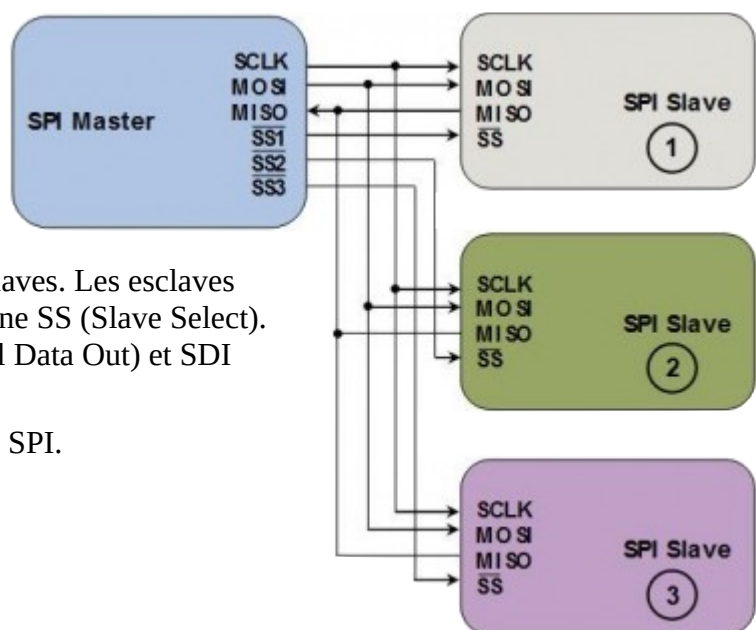
Certains circuits indiquent qu'ils sont munis d'une interface SPI, mais vous pourrez aussi trouver comme appellation « Microwire » (qui est une marque déposée de National Semiconductor) ou bien encore « bus série trois fils ». Un bus de ce type permet la connexion, sous forme série maître-esclave, de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

Les dernières versions d'Arduino incluent une bibliothèque qui permet la communication avec les périphériques SPI.

SPI a des lignes labellisés « MOSI » : Master Output Slave Input, généré par le maître, des lignes labellisées « MISO » : Master Input Slave Output, généré par l'esclave et une ligne d'horloge (SCK : SPI Serial Clock).

Ces trois lignes sont connectées à leurs lignes respectives sur un ou plusieurs esclaves. Les esclaves sont identifiées grâce au signal de leur ligne SS (Slave Select). Autres nommages possibles : SDO (Serial Data Out) et SDI (Serial Data Input).

La figure ci-contre illustre les connexions SPI.

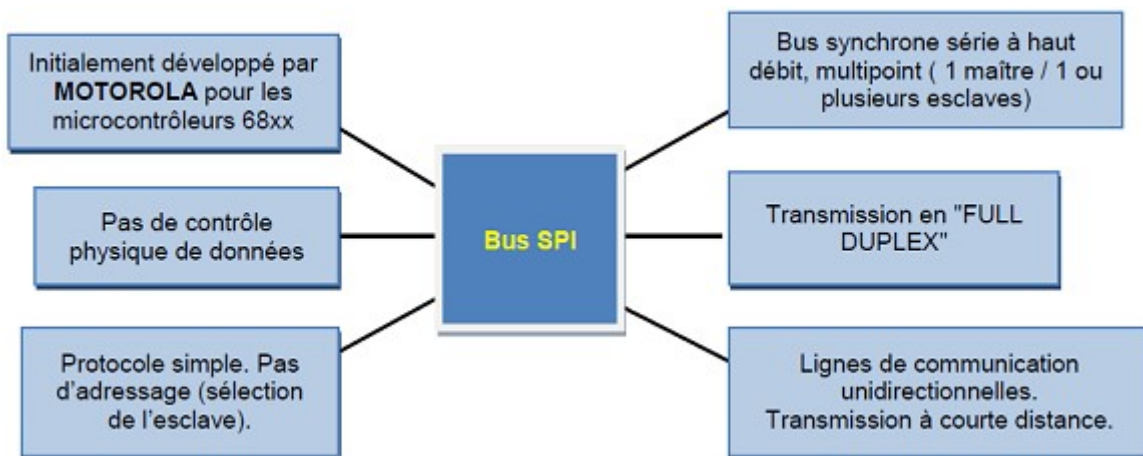


Maître SPI avec plusieurs esclaves SPI

Les numéros de broches à utiliser pour les broches SPI sont listés dans le tableau ci-dessous.

Broches numériques utilisées pour SPI		
Signal SPI	Carte Arduino Uno	Carte Arduino Mega
SS (choix de l'esclave)	10	S3
MOSI (entrées de données)	11	S1
MISO (sorties de données)	12	S0
SCLK (horloge)	13	S2

1.3. Philosophie du bus SPI



2. Schéma de principe

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

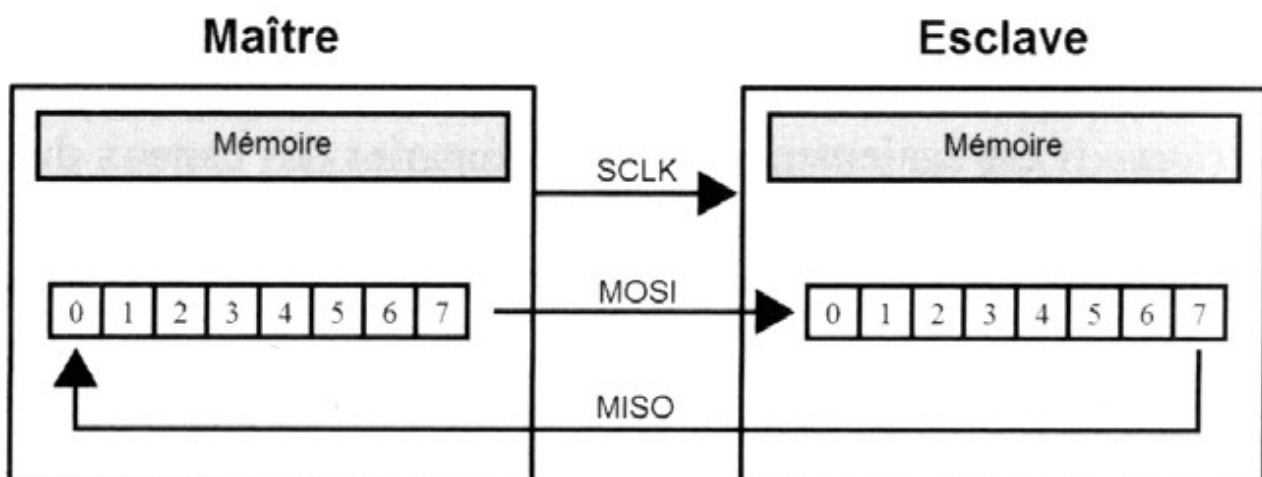


Schéma de principe simplifié d'une liaison SPI

Deux registres à décalages sont généralement mis en œuvre. Ils peuvent être de type matériel ou logiciel, selon les dispositifs employés. Par exemple, le RasPi implémente son registre à décalage

de façon logiciel alors que le MCP3008 (ADC) est équipé d'un registre à décalage matériel. Quelque soit leur mise en œuvre, ces deux registres à décalage forment un buffer circulaire interpuce, 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.

La fréquence de l'horloge dépend principalement des possibilités de réponse de l'esclave. Par exemple, le MCP3008 alimenté en 5 V est capable de gérer des fréquences de données allant jusqu'à 3,6 MHz.

3. Propriétés physiques du bus SPI

3.1. Le support physique

Les données échangées sont des octets. La transmission s'effectue sur 2 fils monodirectionnels (nommés MOSI, MISO).

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

La fréquence de l'horloge de transmission est comprise entre 1 Mhz et 20 Mhz (selon les performances des circuits reliés au bus).

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).

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

	SCLK	SCLK (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

La fréquence d'horloge de transmission (maître) est paramétrable :

Par exemple pour un AT80C5112 en mode maître, les 3 bits du registre SPCON (Serial Peripheral CONTROL register) SPR2, SPR1 et SPR0 permettent de choisir parmi 7 fréquences, obtenues par division de la fréquence de l'horloge du microcontrôleur.

SPR2	SPR1	SPR0	Fréquence de l'horloge SPI
0	0	0	$F_{\mu c}/2$
0	0	1	$F_{\mu c}/4$
0	1	0	$F_{\mu c}/8$
0	1	1	$F_{\mu c}/16$
1	0	0	$F_{\mu c}/32$

1	0	1	F μ c/64
1	1	0	F μ c/128

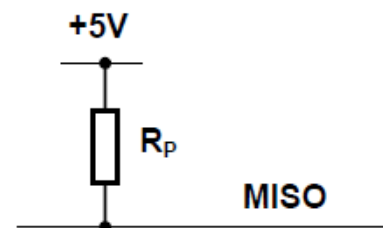
3.2. Synoptique d'une liaison SPI Maître-Esclave

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.

La couche 2 est réalisée par le contrôleur SPI intégré dans un microcontrôleur.

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 polarisation, 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).



Dans les caractéristiques du bus SPI du microcontrôleur AT80C5112, on peut déterminer, grâce à deux paramètres, les fronts où les données sont transmises (acquisitions des valeurs) et les moments où elles peuvent être modifiées.

Ces deux paramètres sont les bits CPOL (ClockPolarity) et CPHA (ClockPhase).

Il existe donc 4 modes de transmission différents (voir tableau ci-dessous). Pour une transmission correcte il faut que ces paramètres soient réglés de la même manière pour tous les composants reliés au bus.

Mode SPI	CPOL	CPHA
0	0	0
1	0	1
2	1	0
3	1	1

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 transmises. CPHA=0 les données sont valides au premier front d'horloge, CPHA=1 elles sont valides au deuxième front.

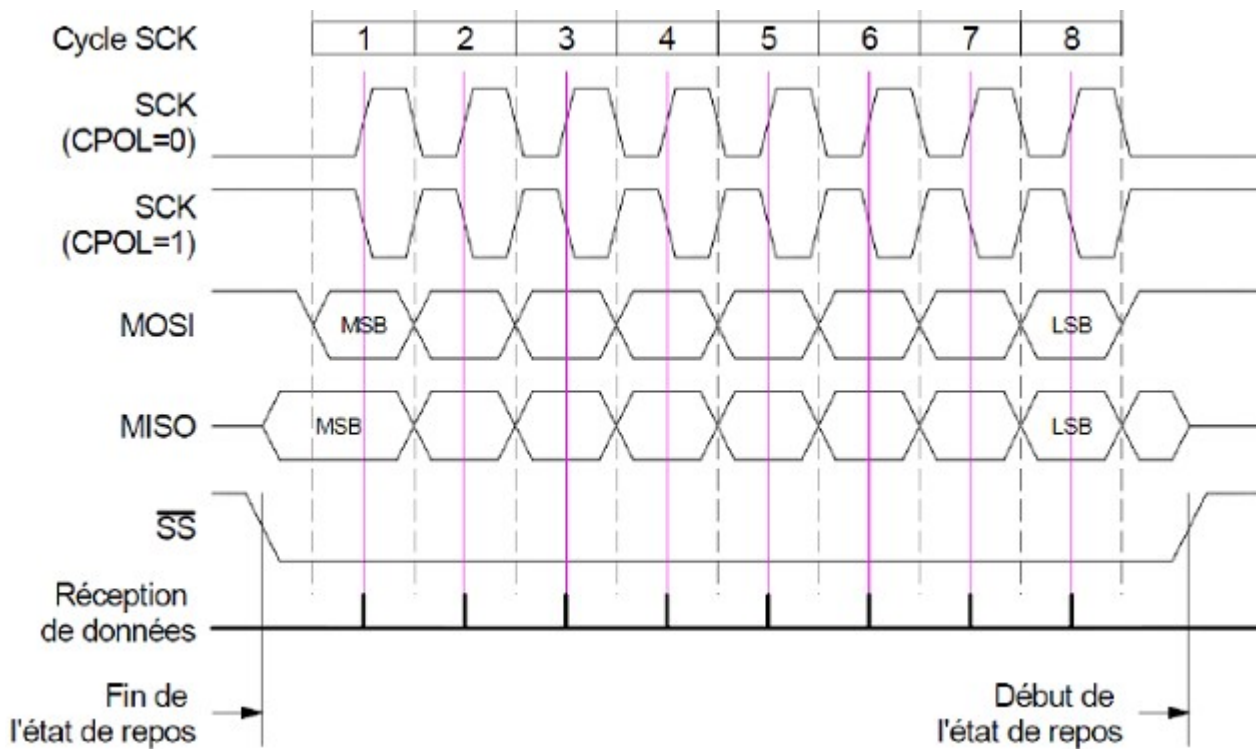
3.3. Format de transfert de données

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

En effet, pour CPOL=0, au repos, l'horloge est au niveau BAS; le premier front est donc un front montant.

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

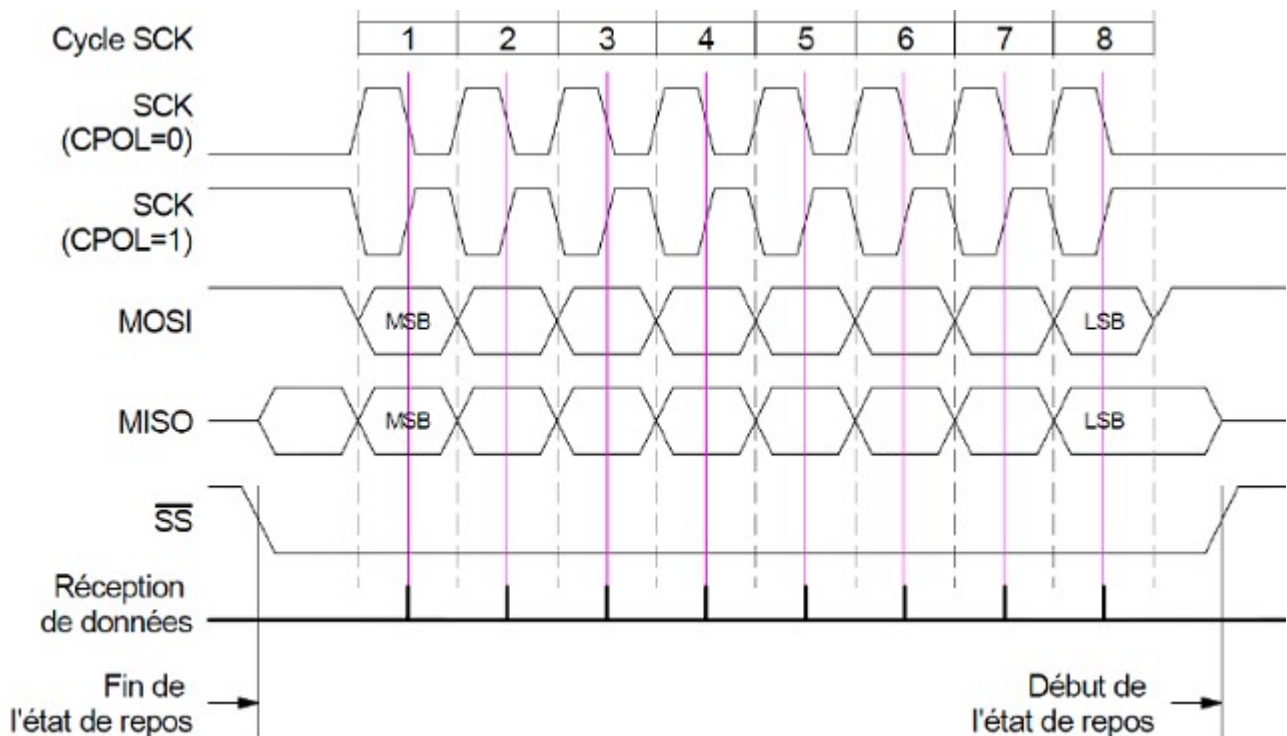
La polarité de l'horloge n'ayant pas d'influence sur le moment où le premier bit de données est valide elle n'a pas d'effet sur le format du transfert de données (voir figure ci-dessous).



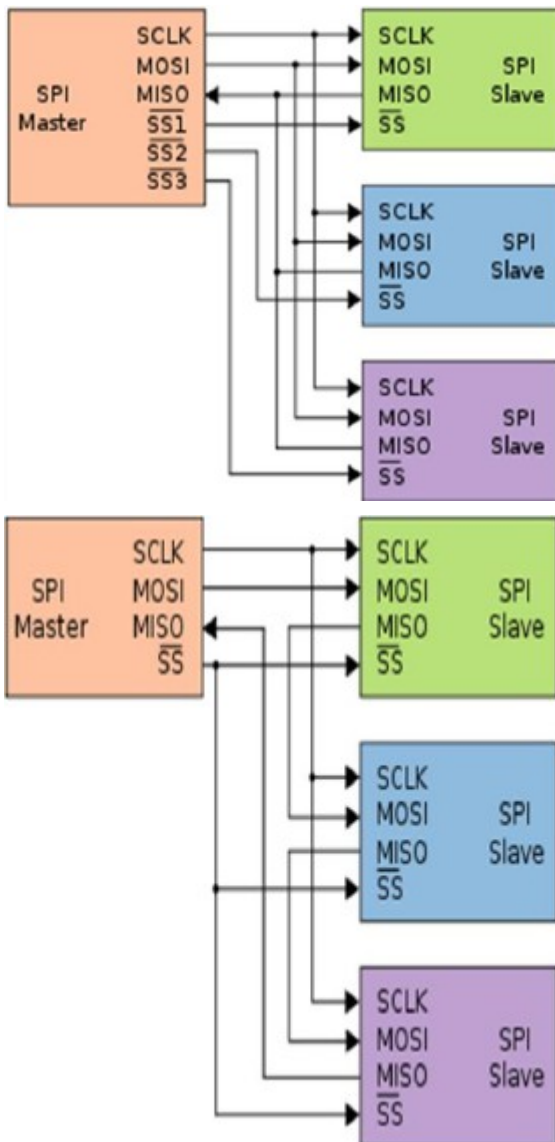
Lorsque $CPHA = 1$, les données sont réceptionnées avec le deuxième front du signal d'horloge.

Pour $CPOL=0$, au repos, l'horloge est au niveau BAS et monte au niveau HAUT après le premier front, le deuxième front est donc un front descendant.

Pour $CPOL=1$, au repos, l'horloge est au niveau HAUT et descend au niveau BAS après le premier front; le deuxième front est donc un front montant.



3.4. Synoptique d'une liaison SPI Maître-Multi-Esclaves



Le maître sélectionne un seul et unique esclave avec lequel il veut rentrer en communication par la mise à niveau logique zéro de /SS 1 2 3, puis, après 8 fronts d'horloge, l'octet de donnée est transféré.

La patte MISO de l'esclave non sélectionné est à l'état haute impédance.

La seule limite au nombre d'esclaves est en fait la possibilité de broches SS du maître.

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

Le maître sélectionne tous les esclaves par la mise à niveau logique zéro de /SS, puis après 3*8 fronts d'horloge, les 3 octets de données sont transférés (dans le cas d'un octet par esclave).

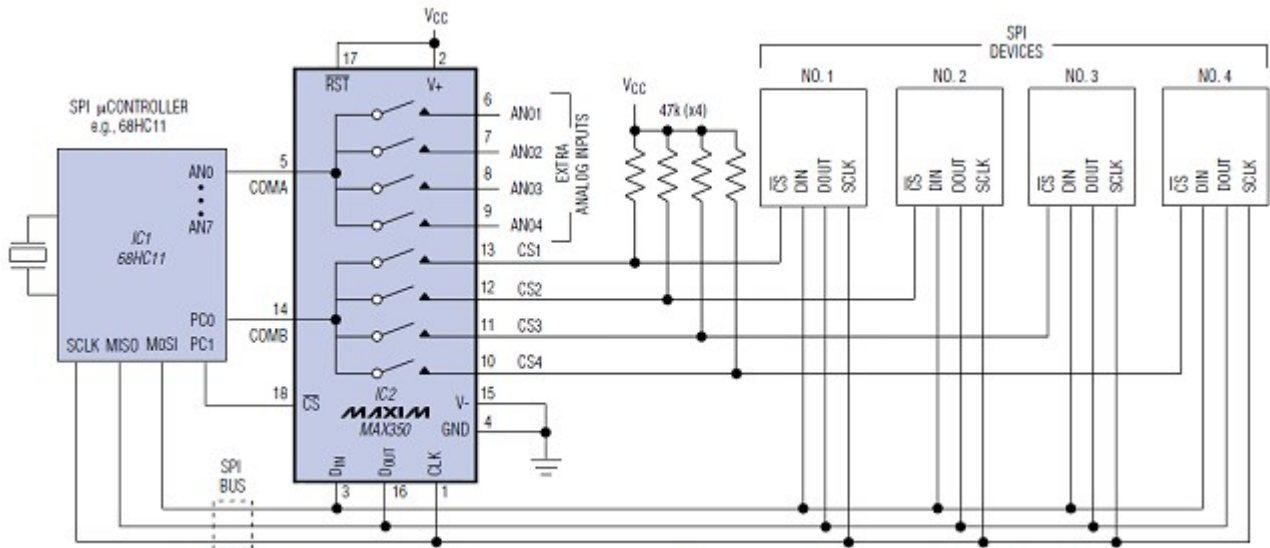
Cette disposition permet de réduire le nombre de lignes /SS, mais en contre partie il faudra un "buffer" plus grand dans le maître (ou une gestion du soft plus élaborée).

Avantages et Inconvénients 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'acquiescement (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	

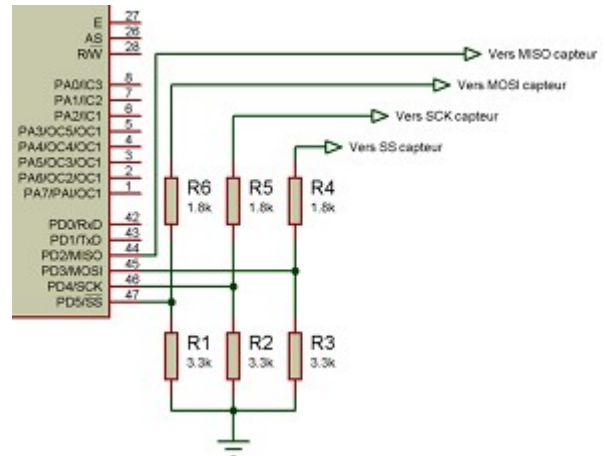
3.5. Bus SPI et niveaux électriques

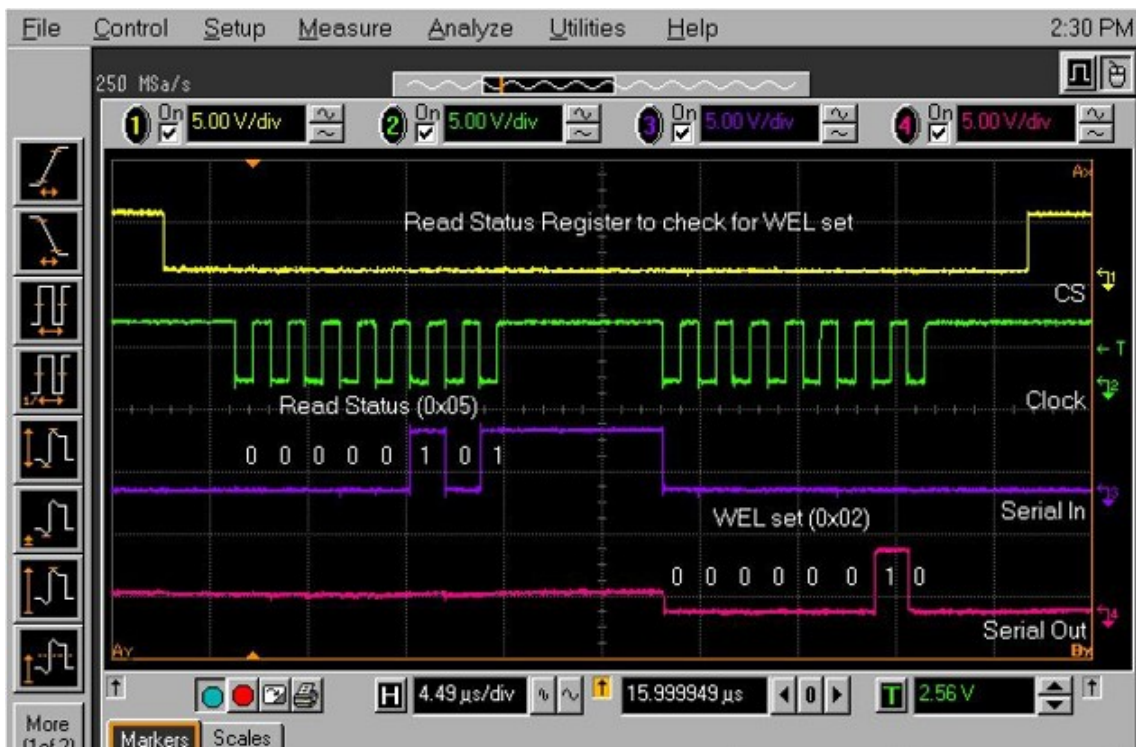
Les signaux échangés sont de types TTL ou CMOS. Il pourra-t-être envisagé dans certains cas de placer des résistances de Pull-up de 47kΩ (voir ci-dessous : utilisation d'un multiplexeur MAXIM MAX350).



Un multiplexeur double à 4 canaux accroît le nombre de canaux et d'entrées de sélection pouvant être utilisés par ce microcontrôleur.

Si on utilise des composants de tension d'alimentation différente par exemple un μp en 5v et un capteur en 3.3v, il convient de procéder à une adaptation du niveau de tension des broches MOSI, SS, SCLK (la MISO n'est pas affectée car compatible avec le μp) par diviseur de tension à base de résistance ¼ de watt.

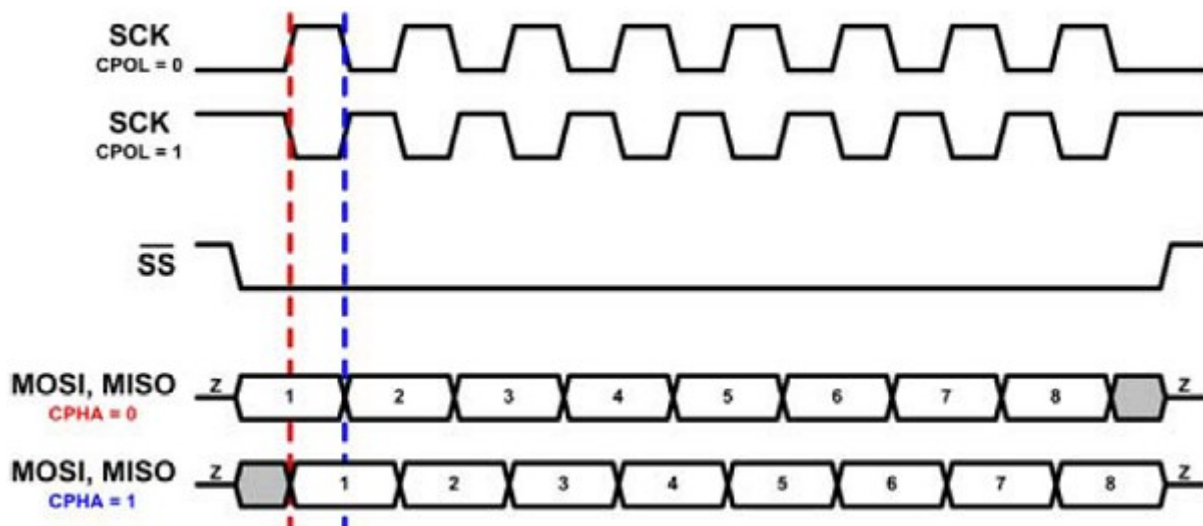




Oscillogrammes des signaux électriques SPI

4. Chronogramme

Toutes les liaisons SPI respectent le chronogramme général ci-dessous, mais quatre variantes existent au niveau de la prise en compte des données qui sont échangées.



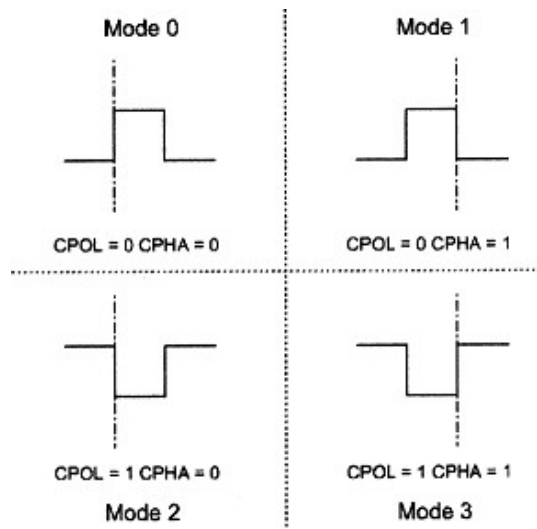
Chronogramme général d'un échange sur une liaison SPI

Ces variantes sont parfois appelées modes SPI et sont alors numérotées de 0 à 3. En présence d'un circuit à interface SPI, il vous suffira de comparer sa fiche technique avec le chronogramme pour savoir dans quel « mode » il fonctionne.

Ces quatre modes sont définis par la polarité de l'horloge, appelée CPOL (Clock POLarity), qui

peut-être égale à 0 ou à 1, et par la phase de cette même horloge, appelée CPHA (Clock PHAse), par rapport aux données, qui peut être aussi égale à 0 ou à 1. Cela conduit aux quatre possibilités, qui indiquent sur quel front du signal d'horloge les données sont échantillonnées et peuvent donc être considérées comme valides.

Par défaut, le mode le plus courant est appelé mode 0, c'est-à-dire celui où CPOL et CPHA sont tous les deux nuls.



Les quatre modes de prise en compte des données de la liaison SPI

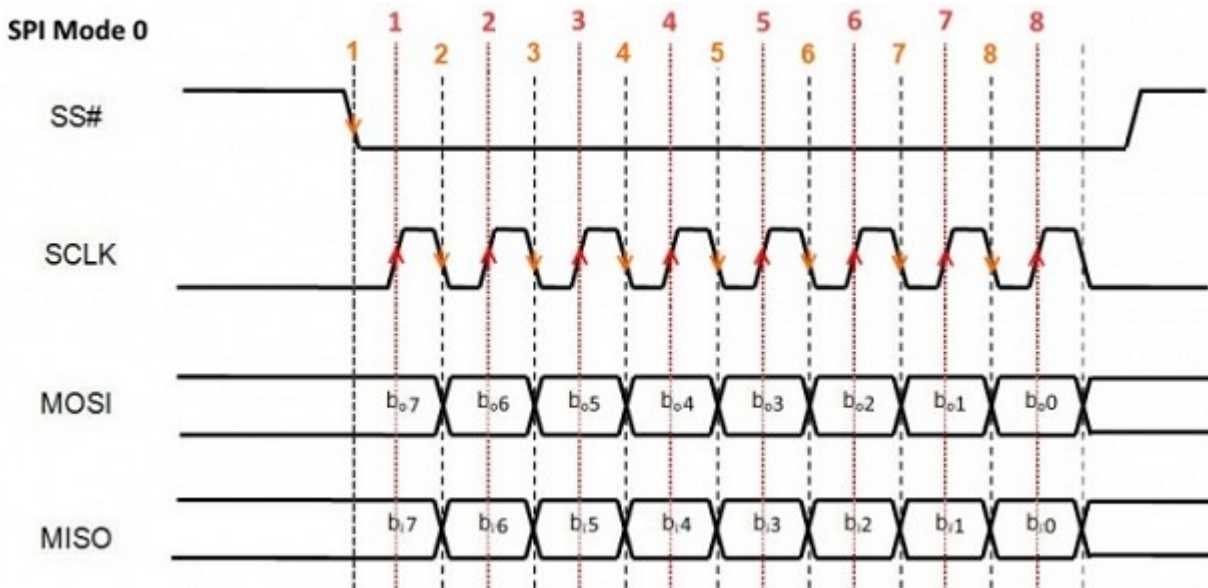
5. Exemple de mise en œuvre d'une communication

Nous prendrons pour exemple l'échange d'une donnée au format d'un octet. En effet à ce stade de la mise en œuvre, il faut avoir à l'esprit que dans la plupart des cas c'est le périphérique esclave qui va imposer au maître le format de la trame, car il s'agit pour le MCU de piloter un composant externe afin d'étendre ces capacités. Il existe donc de nombreux périphériques esclave intégrant une liaison SPI, comme des mémoires, des ADC ou DAC, des RTC, etc...

Imaginons alors d'implémenter le firmware d'un périphérique SPI esclave sur la carte d'un CPLD¹. Le mode 0 étant le plus répandu, nous le choisirons avec une trame de 1 octet. Chaque octet qu'il recevra du maître sera alors copier vers 8 LED et comme la liaison est Full duplex, nous pouvons imaginer de définir 8 entrées dont les valeurs seront transmises par l'esclave.

Pour commencer, observons le chronogramme des signaux de la liaisons SPI, avec en rouge l'échantillonnage des données sur les fronts montant du signal SCLK et en orange l'état transitoire des données sur les fronts descendants:

¹ Complex Programmable Logic Devic : circuit intégré logique qui peut être reprogrammé après sa fabrication



On peut décomposer la communication en trois phases:

1. Début de la communication

Le maître du bus SPI, ici la carte PIC, met le signal SS à 0 de façon à indiquer à l'esclave que la communication commence (Slave Select est actif à l'état bas). Ce signal SS est très important et souvent négligé dans l'implémentation des algorithmes.

Coté microcontrôleur en mode maître SPI, il ne fait pas partie du module interne (uniquement dans le mode esclave) et doit-être défini par le développeur qui devra alors réserver autant de ligne SS qu'il y aura d'esclave avec lesquels communiquer.

Coté CPLD la détection du front descendant de ce signal va permettre d'initialiser les signaux et registres internes avant le premier coup d'horloge, le bit 7 doit alors imposer sont état sur la ligne MISO.

2. Acquisition et transfert des bits

Le maître fera ensuite basculé huit fois le signal d'horloge.

Sur les fronts montants, l'esclave enregistre les données provenant du signal MOSI et le maître mémorise sur la ligne MISO la valeur de chaque bit provenant de l'esclave, c'est le Full duplexe.

Les données sur les lignes MOSI/MISO sont ensuite mise à jour sur les fronts descendants du signal SCLK.

3. Fin de la communication

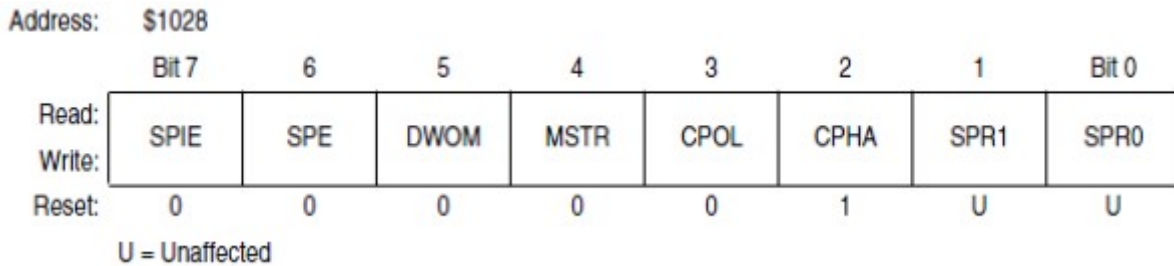
Le maître tire la ligne SS à l'état haut pour indiquer que le transfert est terminé.

6. Les différents autres noms utilisés avec un bus SPI

- SCK :

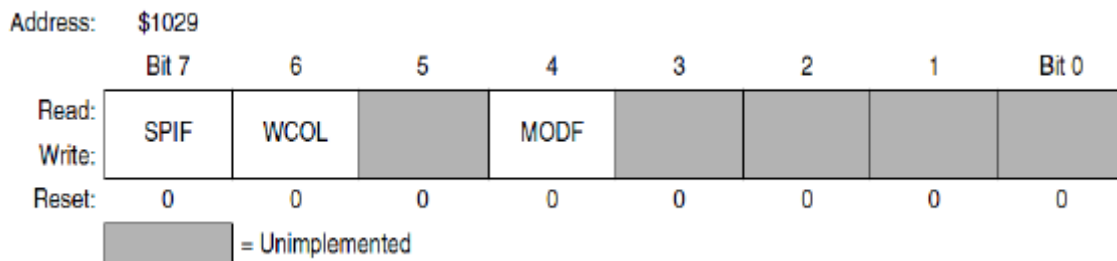
Il s'agit de l'horloge de synchronisation des échanges, elle est générée par le maître et est commune à tous les éléments du bus. Cela a pour avantage d'éviter que chaque composant possède son propre quartz avec les problèmes de précision que cela impose.

7.2. Le registre SPCR: Serial Peripheral Control Register



SPIE	Validation de l'interruption en fin de transfert = 1 interruption validée (si le SPIF du SPSR passe à 1 on a une interruption) = 0 interruption inhibée (si le SPIF du SPSR passe à 1 on n'a pas interruption, il faut le scruter)
SPE	Validation SPI = 1 le port D est en SPI = 0 le port D est en I/O normal
DWOM	Port D Wire or Mode Option : =1 les sorties du port D sont en type collecteur ouvert = 0 elles sont de types CMOS
MSTR	Master Mode Select: = 1 le 68HC est en Master (SS à 1) = 0 le 68HC est en Slave (SS à 0)
CPOL	Clock POLarity = 1 le SCK est à 1 au repos = 0 le SCK est à 0 au repos
CPHA	Clock PHAse = 1 le SCK est valide sur front montant = 0 le SCK est valide sur front descendant
SPR1 & 0	= 00 l'horloge interne est divisée par 2 = 01 l'horloge interne est divisée par 4 = 10 l'horloge interne est divisée par 8 = 11 l'horloge interne est divisée par 16 Pas d'effet en mode slave

7.3. Le registre SPSR: Serial Peripheral Status Register



SPIF	Drapeau de fin de transfert SPI / Flag SPI (si SPIE=1) = 1 le transfert est terminé = 0 le transfert n'est pas terminé
WCOL	Write COLlision = 1 collision en écriture détectée

	= 0 pas de collision
MODF	Mode Fault: drapeau qui indique un défaut de mode = 1 (le 68HC est en master avec un SS=0) = 0

7.4. Le registre SPDR: Serial Peripheral Data Register

Address: \$102A

	Bit 7	6	5	4	3	2	1	Bit 0
Read:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Write:	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reset:	Indeterminate after reset							

C'est le registre où s'effectue le transfert des données.

7.5. Le système de détection d'erreur SPI

Seules 2 types d'erreurs sont détectés:

- Erreur de mode: détection d'un niveau bas sur SS alors que le 68HC est déclaré en Maître, il ne peut y avoir 2 maîtres sur le même bus SPI
- Erreur de collision: détection d'une écriture dans le SPDR alors que le transfert n'est pas terminé.

8. Constitution pour un système à microcontrôleur PIC

8.1. Le MSSP Control Register 1 (SSPCON1)

REGISTER 19-2: SSPCON1: MSSP CONTROL REGISTER 1 (SPI MODE)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV ⁽¹⁾	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 WCOL	: Write Collision Detect bit (Transmit mode only) 1 = Le registre SSPBUF register est écrit pendant que l'octet prévu est transmis (Doit être effacé par le programme) 0 = Pas de détection
bit 6 SSPOV	: Receive Overflow Indicator bit(1) SPI mode esclave: 1 = un nouvel octet est reçu pendant que le registre SSPBUF est encore en possession de l'ancien. En cas d'"overflow"(débordement) la donnée du SSPSR est perdue. Le débordement n'intervient que dans le mode Esclave. L'utilisateur doit lire le SSPBUF, seulement en cas de transmission de donnée, pour éviter le maintien de l'overflow il doit être effacé dans le soft. 0 =Pas de débordement
bit 5 SSPEN	: Master Synchronous Serial Port Enable bit 1 = validation de la configuration SCK, SDO, SDI and SS comme port série(2)

	0 = pas de validation du mode SPI, les broches sont en E/S normales(2)
bit 4 CKP	: Clock Polarity Select bit 1 = horloge au niveau haut logique au repos 0 = horloge au niveau bas logique au repos
bit 3-0 SSPM3:SSPM0	: Master Synchronous Serial Port Mode Select bits 0101 = SPI mode Esclave, clock = SCK, SS non validée, SS peut être utilisée en E/S(3) 0100 = SPI mode Esclave, clock = SCK, SS (select slave) validée(3) 0011 = SPI mode Maître, clock = TMR2 output/2(3) 0010 = SPI mode Maître, clock = FOSC/64(3) 0001 = SPI mode Maître, clock = FOSC/16(3) 0000 = SPI mode Maître, clock = FOSC/4(3)

Note 1: Dans le mode Maître, le bit de dépassement n'est pas mis à 1 à chaque transmission ou nouvelle réception mais est initialisé par une écriture dans le registre SSPBUF.

2: Si validé, cette broche doit être configuré comme une entrée ou une sortie.

8.2. Le MSSP Status Register (SSPSTAT)

REGISTER 19-1: SSPSTAT: MSSP STATUS REGISTER (SPI MODE)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE ⁽¹⁾	D/Ā	P	S	R/W	UA	BF
bit 7							bit 0

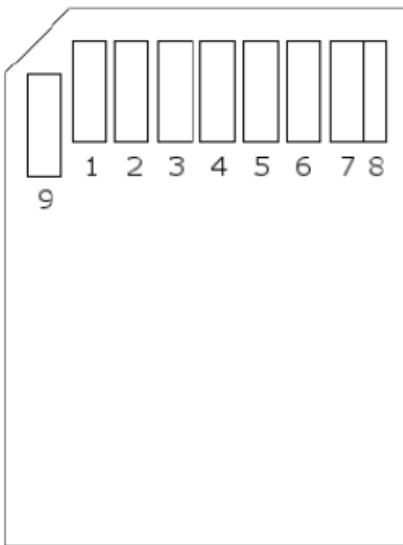
Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 SMP	: Sample bit Dans le mode Maître 1 = Echantillonnage de la donnée entrante à la fin du temps de sortie de la données sortante 0 = Echantillonnage de la donnée entrante au milieu du temps de sortie de la données sortante Dans le mode Esclave mis à 0
bit 6 CKE	: SPI Clock Select bit 1 = Transmission sur front de l'état actif vers état repos de l'horloge 0 = Transmission sur front de l'état repos vers état actif de l'horloge Dans le mode Esclave mis à 0
bit 0 BF	: Buffer Full Status bit (Receive mode only) 1 = Réception complète, SSPBUF plein 0 = Réception incomplète, SSPBUF vide

Les bits 1, 2, 3, 4 et 5 ne sont pas utilisés en mode SPI.

8.3. Le Serial Receive/Transmit Buffer Register SSPBUF

Ce registre est utilisé pour lire ou écrire les données de / vers l'extérieur du PIC.



Pin	Fonction SPI
1	Chip select
2	MOSI
3	GND
4	VCC
5	SCLK
6	GND
7	MISO
8	SD (SD only)
9	SD (SD only)

9.2. 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.

- Comparaison entre I2C et SPI : <http://www.maximintegrated.com/en/app-notes/index.mvp/id/4024>
- Référence bibliothèque Arduino SPI : <http://arduino.cc/en/Reference/SPI>