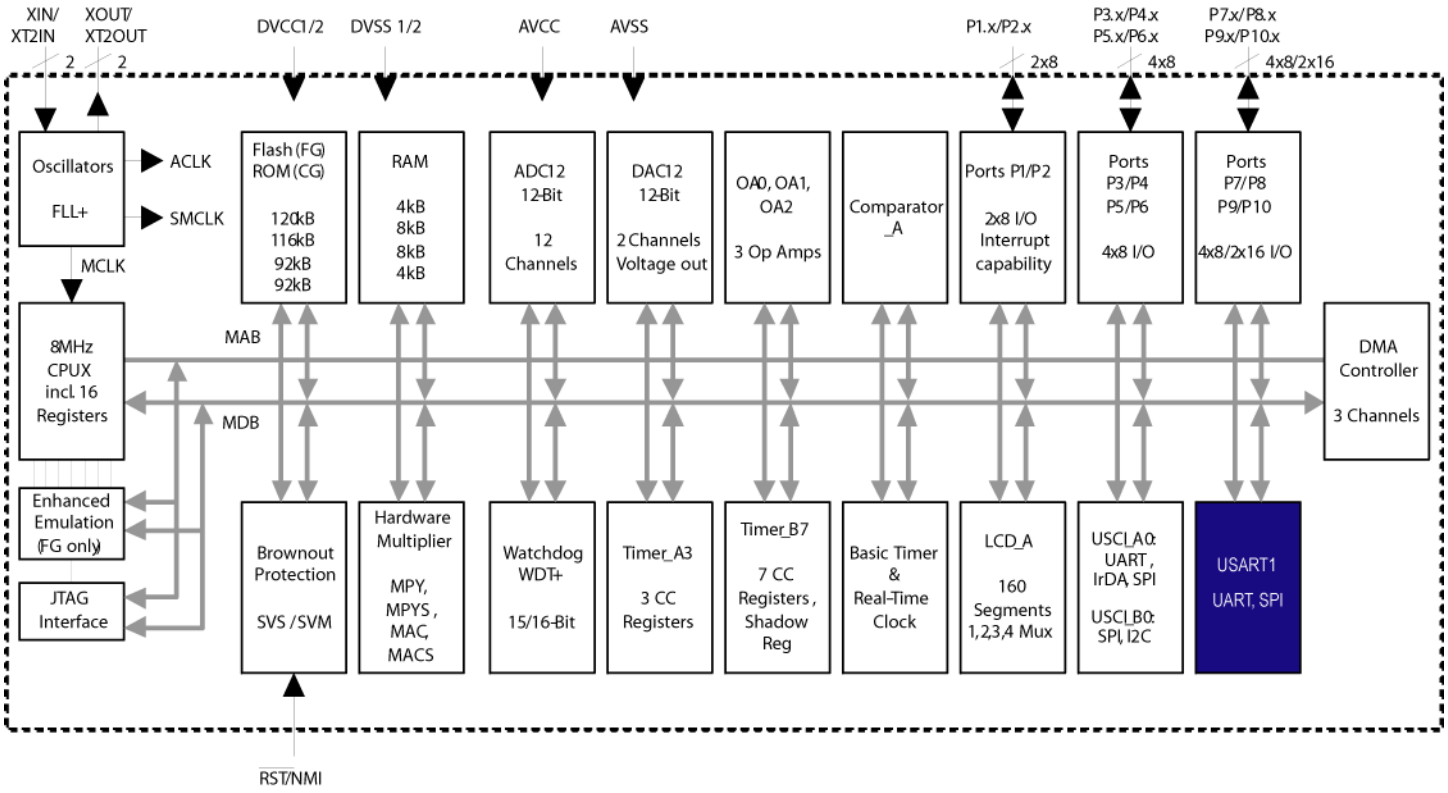
USART en mode SPI



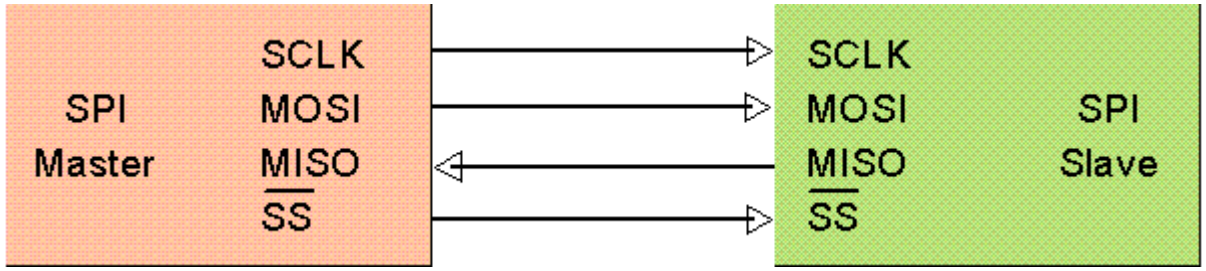
USART

- USART est une abréviation signifiant
  Universal Synchronous & Asynchronous Receiver Transmitter.

- Beaucoup de circuits intégrés disposent désormais d'UART qui peuvent communiquer de manière synchrone; de tels périphériques portent le nom d'USARTs.

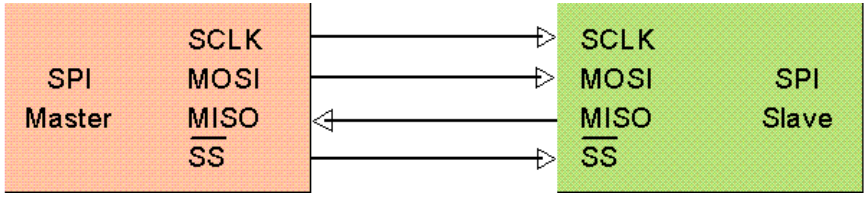
Communication série

- Une liaison SPI (pour Serial Peripheral Interface) est un bus de donnée série synchrone baptisé ainsi par Motorola, et qui opère en **Full Duplex** (voir article Wikipédia).
- Les circuits communiquent selon un schéma maître-esclaves, où le maître s'occupe totalement de la communication.
- Plusieurs esclaves peuvent coexister sur un bus, la sélection du destinataire se fait par une ligne dédiée entre le maître et l'esclave appelée chip select.



USART en mode SPI (Serial Peripheral Interface)

En mode synchrone, l'USART connecte le MSP430 a un périphérique externe via 3 ou 4 broches :



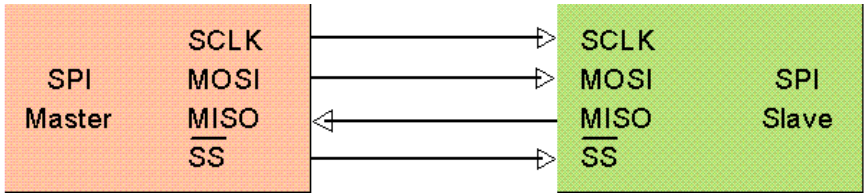
- SIMO ou MOSI: **S**lave **I**nput – **M**aster **O**utput
- SOMI ou MISO : **S**lave **O**utput – **M**aster **I**nput
- SCLK ou UCLK: **U**sart SPI **CL**oc**K**
- STE ou SS : **S**lave **T**ransmit **E**nable (aussi **S**lave **S**elect)

Le mode SPI est sélectionné lorsque le bit SYNC est forcé à 1 et le bit I2C forcé à 0.

Les caractéristiques du mode SPI comprennent :

- des données de longueur de 7bits ou 8-bits
- interface nécessitant 3 ou 4 lignes
- mode en Master (maître) ou en Slave (esclave)
- registre à décalage de transmission et de réception indépendant
- buffer de ligne de transmission et de réception séparés
- polarité et contrôle de phase de l'horloge UCLK
- fréquence programmable pour l'horloge UCLK en mode Master
- source d'interruption indépendante pour la transmission et la réception

USART en mode SPI (Serial Peripheral Interface)

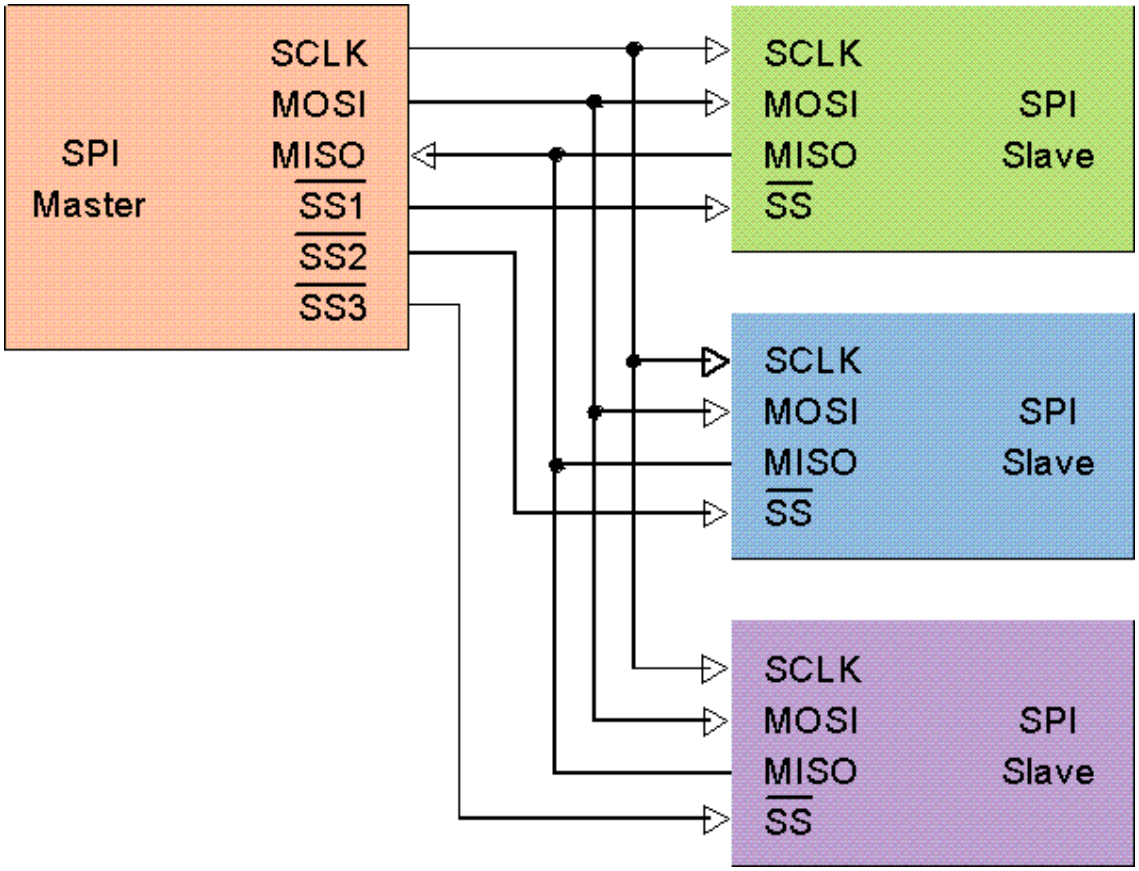


Il existe d'autres noms qui sont souvent utilisés.

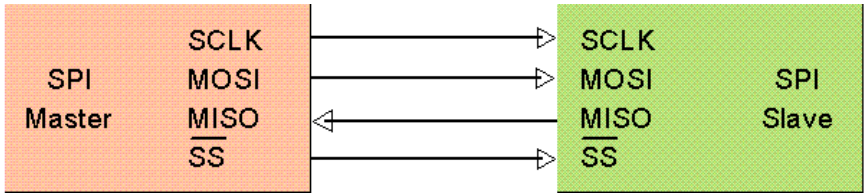
* SCK — Horloge (généré par le maître)
* SDI,DI,SI — Serial Data IN
* SDO,DO,SO — Serial Data OUT
* nCS, CS, nSS, STE — SS

Dans le cas de la convention de nommage SDI/SDO, le SDO du maître doit-être reliées au SDI de l'esclave et vice-versa.
Pour éviter les confusions au moment du câblage, il est donc souvent recommandé d'utiliser les dénominations MISO-MOSI (ou SOMI-SIMO) qui évite une certaine ambiguïté.

Liaison SPI avec un maître et trois esclaves



Fonctionnement



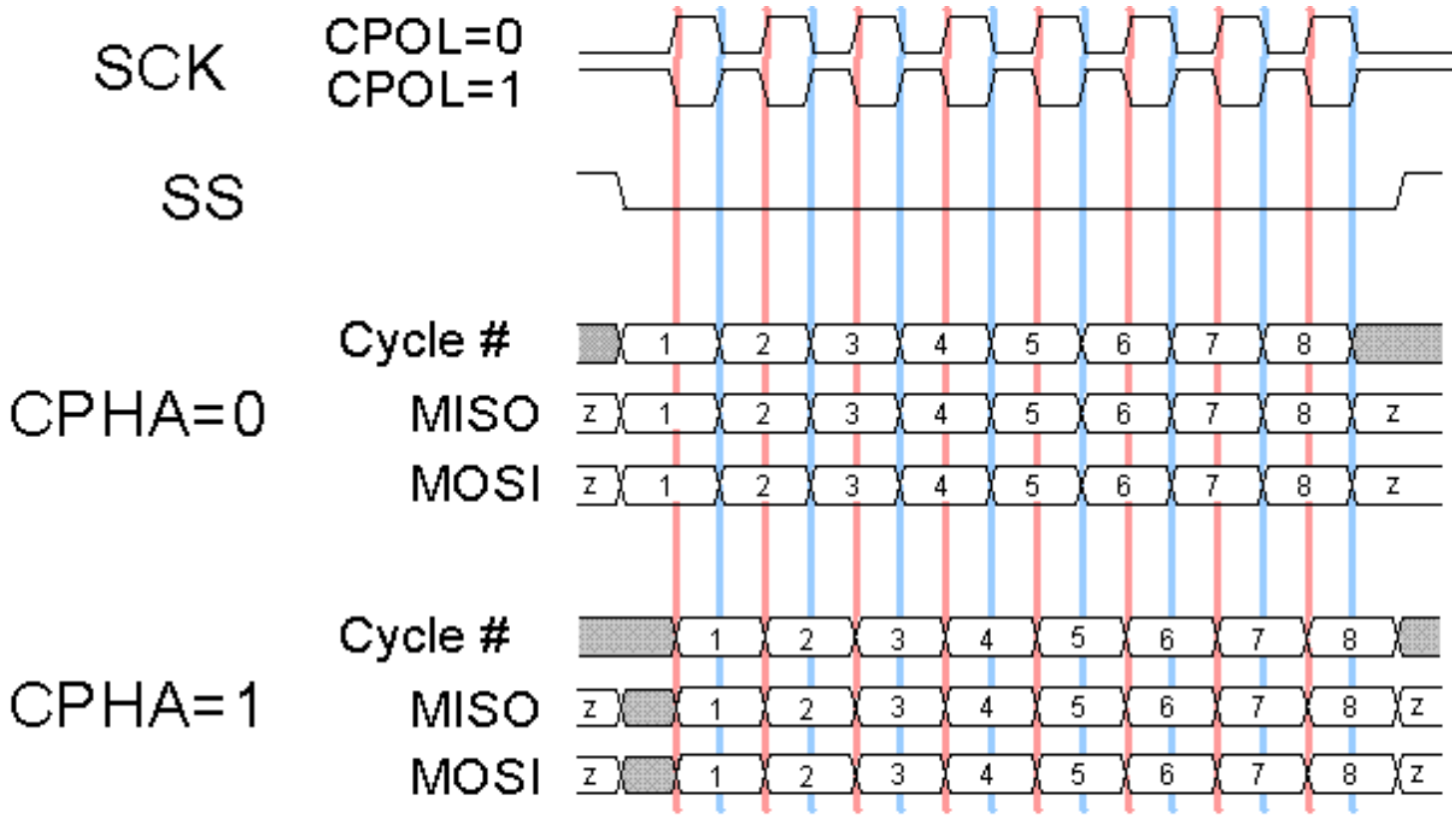
Une transmission SPI typique est une communication simultanée entre un maître et un esclave.

* Le maître génère l'horloge et sélectionne l'esclave avec qui il veut communiquer
* L'esclave répond aux requêtes du maître

A chaque coup d'horloge le maître et l'esclave s'échangent un bit.
Après huit coups d'horloges le maître a transmis un octet à l'esclave et vice-versa.
La vitesse de l'horloge est réglée selon des caractéristiques propres aux périphériques.

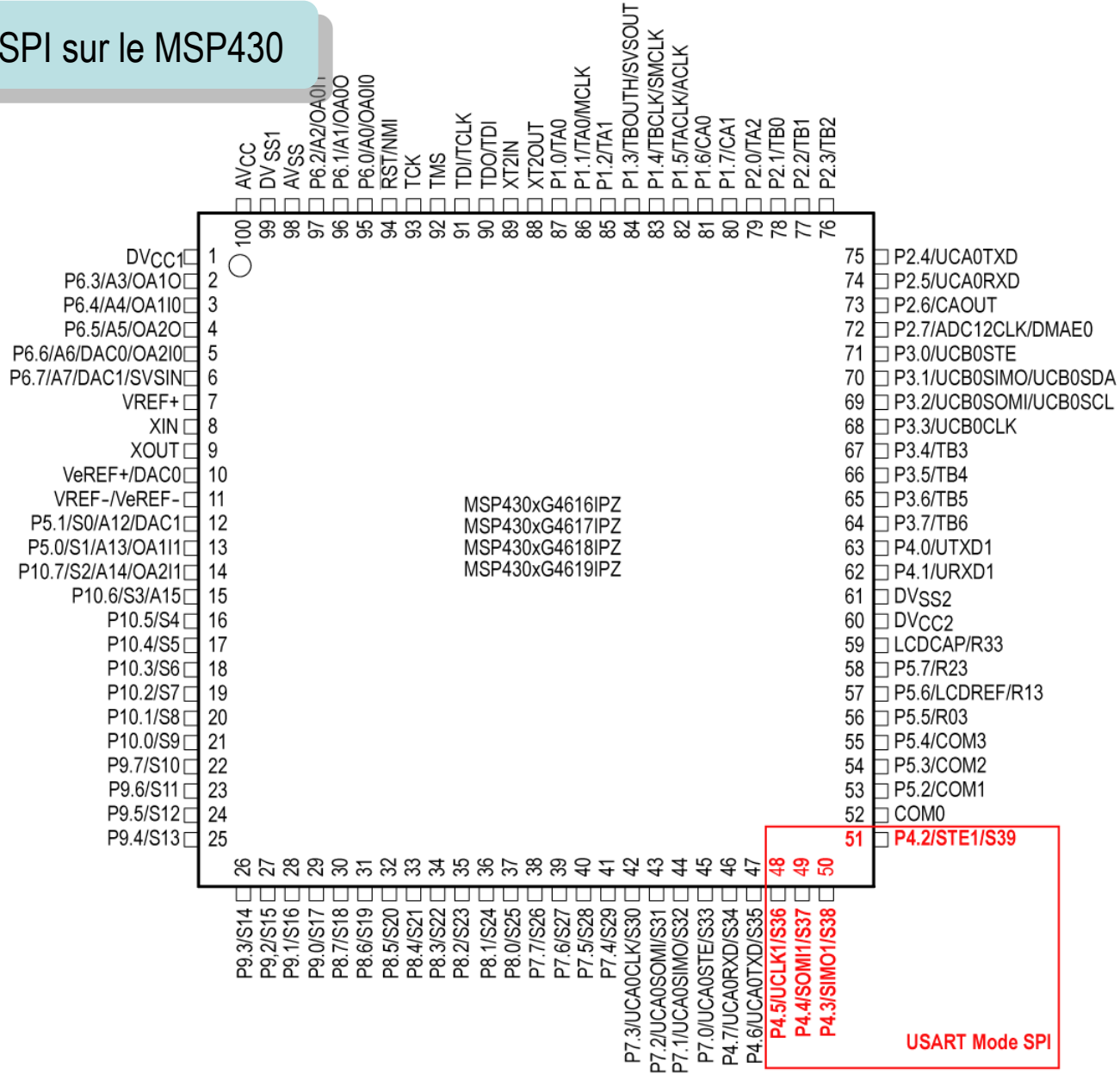
Fonctionnement

Le maître doit aussi configurer la polarité et la phase des données transmises.



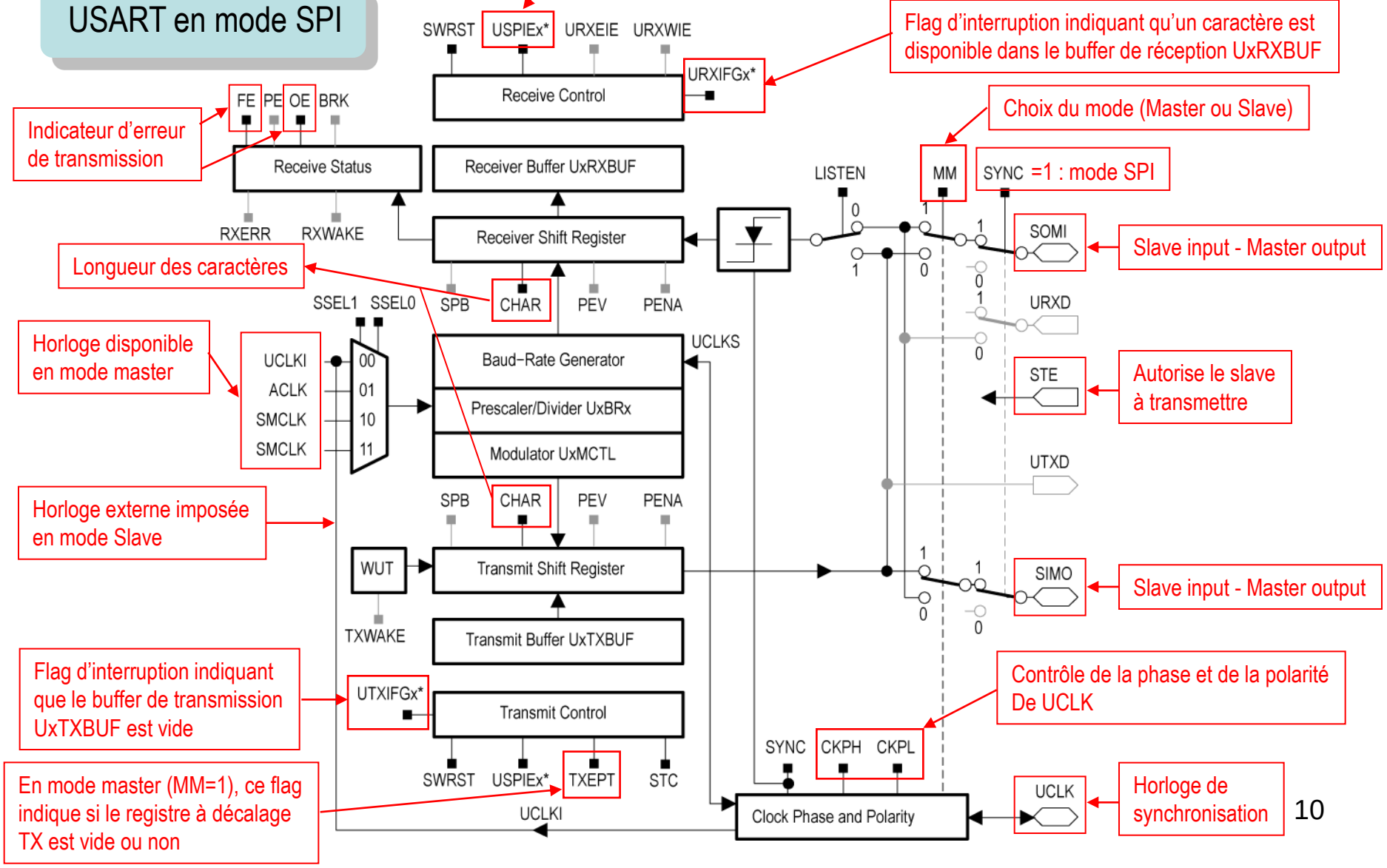
Périphérique : USART - SPI

USART en mode SPI sur le MSP430



Périphérique : USART - SPI

USART en mode SPI



USART en mode SPI

En mode SPI les données sont transmises et reçues de manière sérielle en utilisant une horloge partagée générée par le Master.
Une ligne additionnelle STE, contrôlée par le Master permet d'activer l'esclave avec lequel il doit y avoir un échange de données

SIMO : **S**lave **I**nput, **M**aster **O**utput
- en mode Master : SIMO est la ligne de sortie des données.
- en mode Slave : SIMO est la ligne d'entrée des données.

SOMI : **S**lave **O**utput, **M**aster **I**nput
- en mode Master : SOMI est la ligne d'entrée des données.
- en mode Slave : SOMI est la ligne de sortie des données.

UCLK : **U**SART SPI **CL**oc**K**
- en mode Master : UCLK est une sortie.
- en mode Slave : UCLK est une entrée.

STE Slave **T**ransmit **E**nable (utilisé en mode 4 lignes pour plusieurs Master sur le même bus
- 4-lignes mode Master :
  - Si STE=1, SIMO et UCLK travaillent normalement.
  - Si STE=0, SIMO et UCLK sont configurés entrées.
- 4-lignes mode Slave :
  - Si STE=1, les fonctions RX/TX du Slave sont désactivées et SOMI est configuré en entrée.
  - Si STE=0, les fonctions RX/TX du Slave sont activées et SOMI travaille normalement.

USART : initialisation et reset

Le reset de l'USART est activé par la ligne PUC (Power Up Clear) ou par le flag SWRST.
Après une activation du PUC, le flag SWRST est automatiquement mis à 1, ce qui maintient l'USART en mode RESET.
Les flag URXIEx (Interrupt Enable RX), UTXIEx (Interrupt Enable TX), URXIFGx (Interrupt Flag RX) OE (Overrun Error) et FE () sont forcés à 0 alors que UTXIFGx (Interrupt Flag TX) est forcé à 1.
Le flag USPIEx (SPI Enable) n'est pas modifié par l'état de SWRST.
La désactivation du flag SWRST (mise à 0) libère l'USART

La procédure d'initialisation des USART est la suivante :

(1) ⇨ Forcer à 1 le flag SWRST par software (BIS.B #SWRST,&UxCTL)
(2) ⇨ Initialiser tous les registres des USART en maintenant le flag SWRST=1 (registre UxCTL)
(3) ⇨ Activer le périphérique USART via les registres MEx (USPIEx)
(4) ⇨ Forcer à 0 le flag SWRST par software (BIC.B #SWRST,&UxCTL)
(5) ⇨ Démasquer les sources d'interruption via les registres IEx (URXIEx and/or UTXIEx)

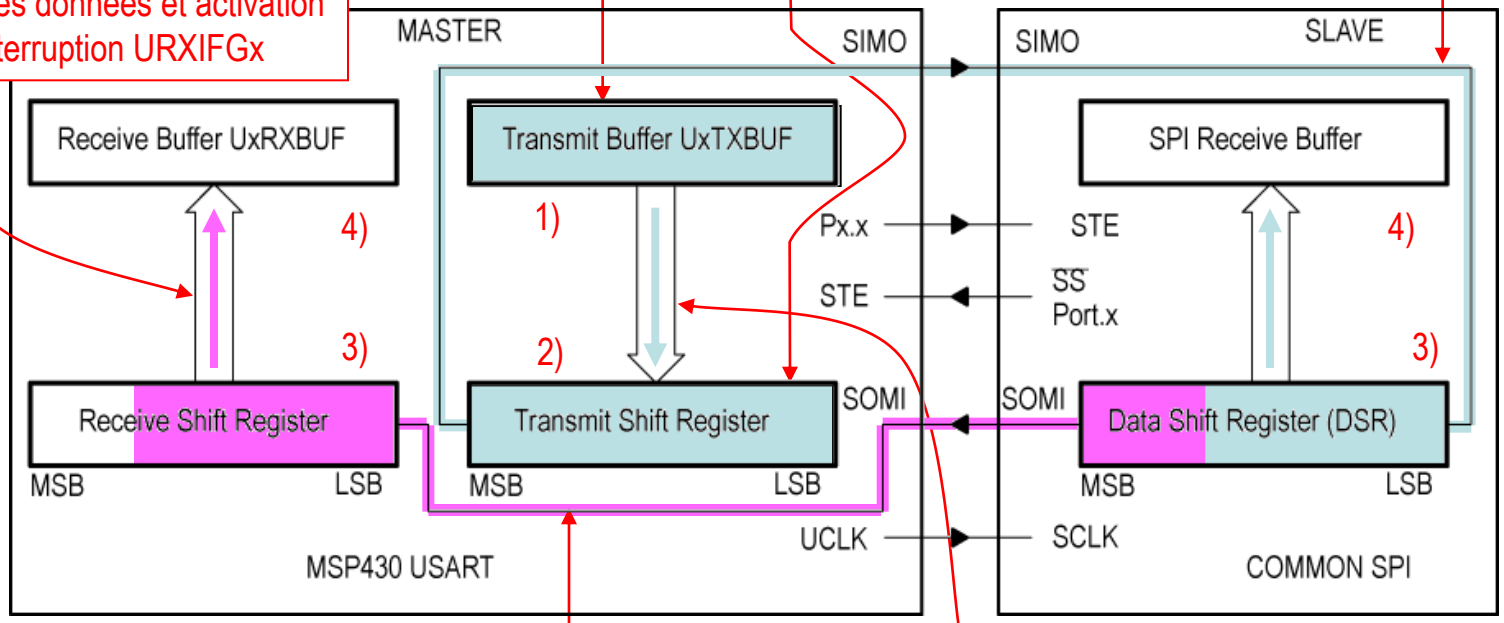
Configuration en mode Master (référence MSP430)

Données transmises du Master au Slave sur le flanc de référence de l'horloge UCLK

Les données (8 bits) sont placées dans le buffer UxTXBUF

Les données (8 bits) sont transférée dans le registre à décalage TX lorsqu'il est vide

Transfert des données et activation du flag d'interruption URXIFGx



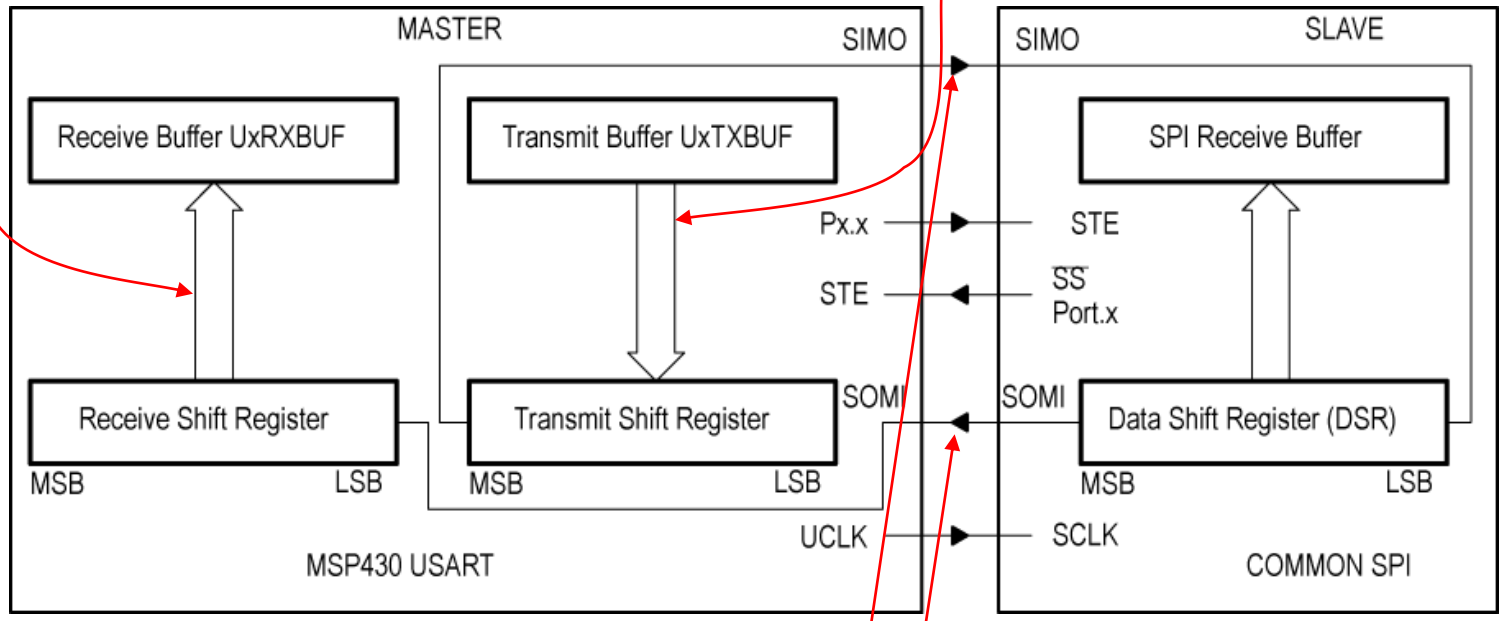
Données transmises du Slave au Master sur le flanc opposé au flanc de référence de l'horloge UCLK

Données transférées si le registre est vide et activation du flag d'interruption UTXIFGx pour indiquer qu'une donnée peut être transmise. Le buffer UxTXBUF est prêt pour recevoir une nouvelle donnée

Configuration en mode Master (référence MSP430)

Le flag d'interruption URXIFGx indique une opération de transfert complète

Le flag d'interruption UTXIFGx n'indique pas une opération de transfert complète



En mode Master une transmission complète est indiquée par la mise à 1 du flag TXEPT (transmitter empty). Pour recevoir un caractère dans l'USART en mode Master, une donnée doit être écrite dans le buffer de transmission UxTXBUF puisque les deux lignes (SIMO et MOSI) travaillent simultanément.

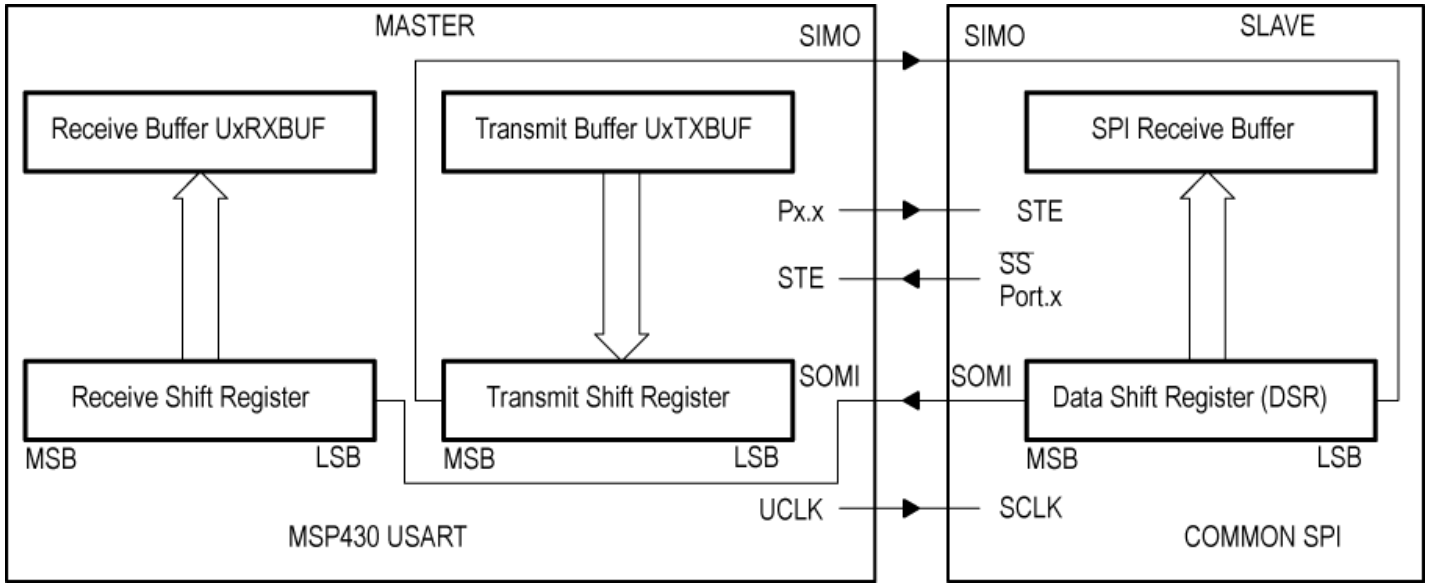
Mode SPI Master à 4 lignes

En mode SPI Master à 4 lignes, STE est utilisé pour prévenir des conflits entre Masters

Le Master travaille normalement lorsque la ligne STE est à l'état haut.

Lorsque la ligne STE est à l'état bas :

- ⇨ Les lignes SIMO et UCLK sont configurés en entrée et ne contrôle plus le bus
- ⇨ Le flag d'erreur FE est forcé à 1 indiquant une violation de l'intégrité de la communications par une mauvaise manipulation de l'utilisateur

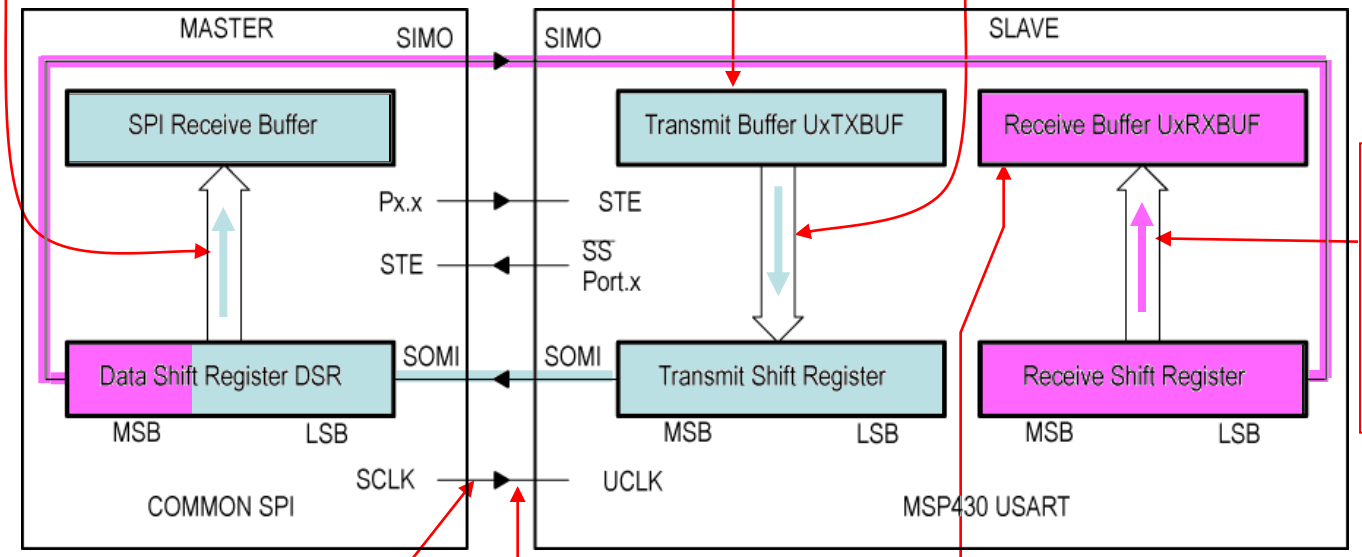


Configuration en mode Slave (référence MSP430)

Transfert des données se fait par la ligne SIMO

Les données sont écrites dans le Buffer UxTXBUF

Les données (8 bits) sont transférées dans le registre à décalage TX lorsqu'il est vide



Les données passent dans le Buffer UxRXBUF lorsque les 8 bits ont été transférés.
Le flag d'interruption URXIFGx est forcé à 1

UCLK est utilisé comme entrée par le Master, il est donc crée par le Slave

Le flag d'erreur OE (overrun) est forcé à 1 si la donnée précédente Placée dans le buffer UxRXBUF n'a pas été lue

La vitesse de transfert des données est déterminée la fréquence de l'horloge UCLK et non par le générateur interne de Baudrate

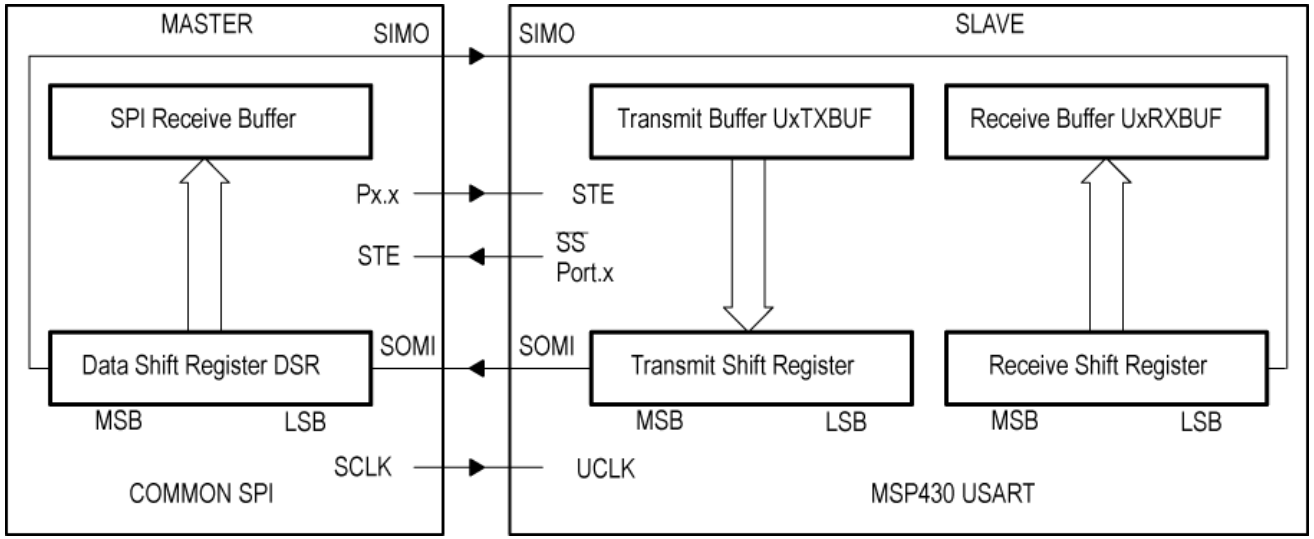
Mode SPI Slave à 4 lignes

En mode Slave 4 lignes, STE, produit par le Master est utilisé par le Slave pour activer les opérations de transmission et de réception.

Lorsque la ligne STE est à l'état bas, le Slave travaille normalement

Lorsque la ligne STE est à l'état haut :

- ⇨ les opérations de réception en cours sur la ligne SIMO sont stoppées
- ⇨ SOMI est configuré en entrée



Activation / désactivation du SPI

Le flag USPIEx active / désactive la transmission / réception de l'USART en mode SPI

Lorsque USPIEx=0, l'USART désactive les opérations de transfert une fois le transfert en cours terminé, ou immédiatement si aucun transfert n'a lieu

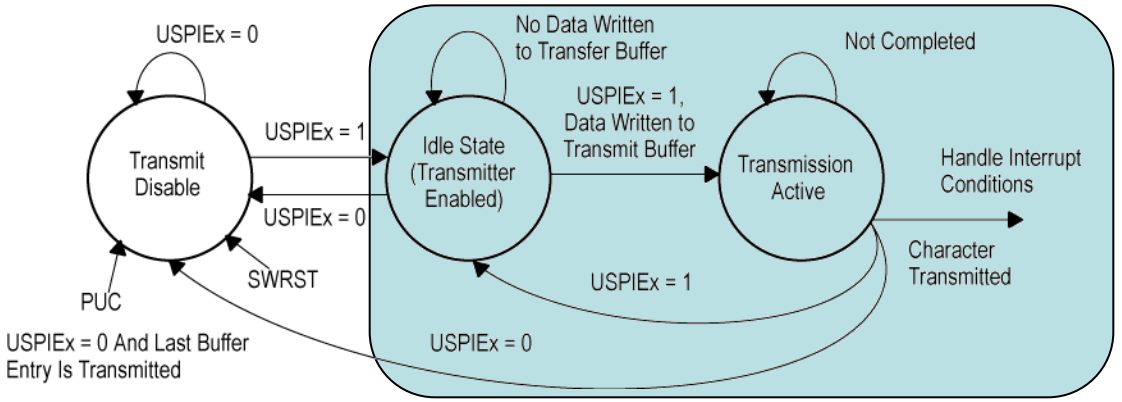
Lorsque le flag PUC (Power Up Clear) ou un reset software (flag SWRST) est activé, USART est immédiatement stoppé sans attendre la fin d'un éventuel transfert en cours.

Transmission active

Lorsque le flag USPIEx = 0, l'écriture dans le buffer UxTXBUF n'active aucune transmission de données.
Les données écrites dans le buffer UxTXBUF commence à être transmises lorsque le flag USPIEx = 1 et que l'horloge de référence BRCLK est active.

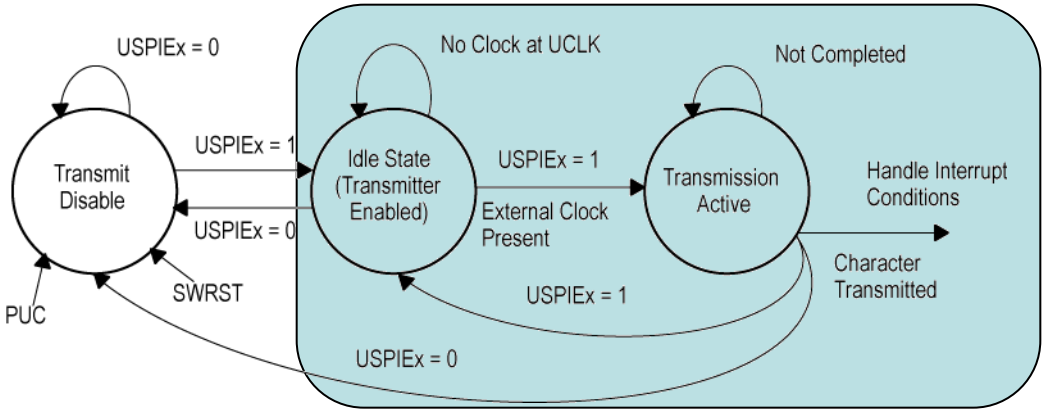
Mode Master :

- activation de la transmission



Mode Slave :

- activation de la transmission

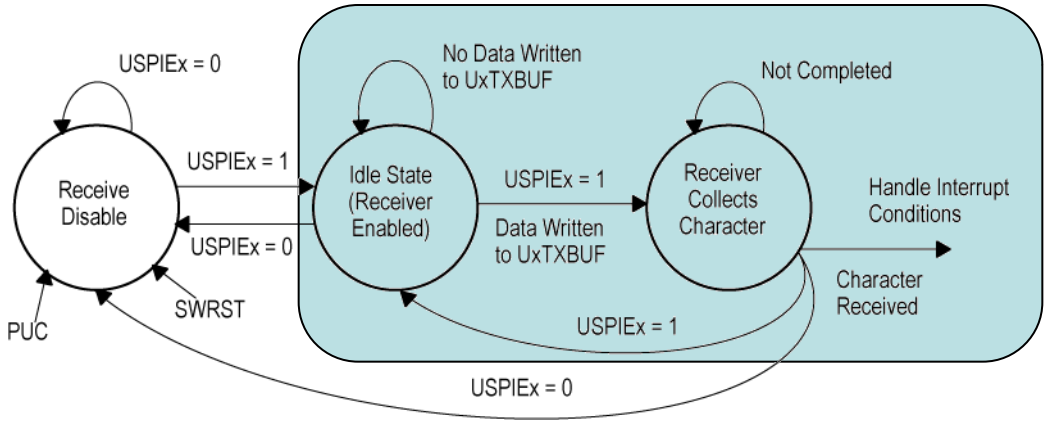


Réception active

Lorsque USPIEx = 0, l'horloge UCLK est désactivée et le registre de décalage de réception ne fonctionne plus.

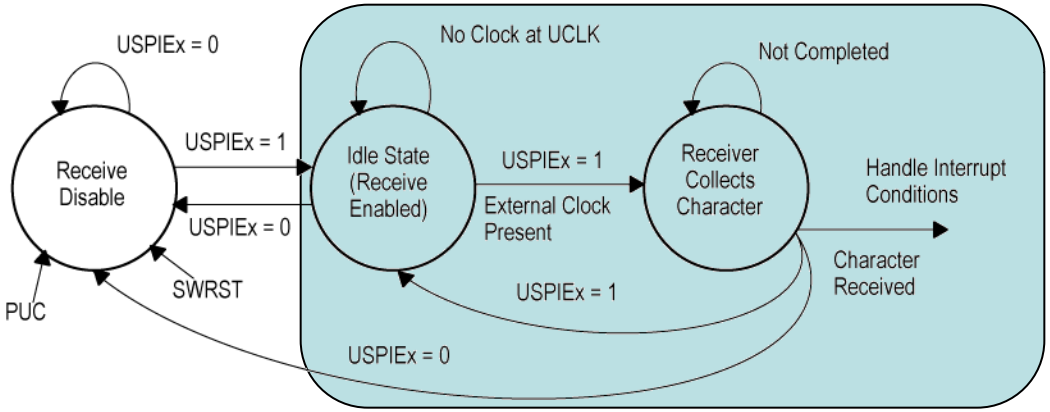
Mode Master :

- activation de la réception



Mode Slave :

- activation de la réception

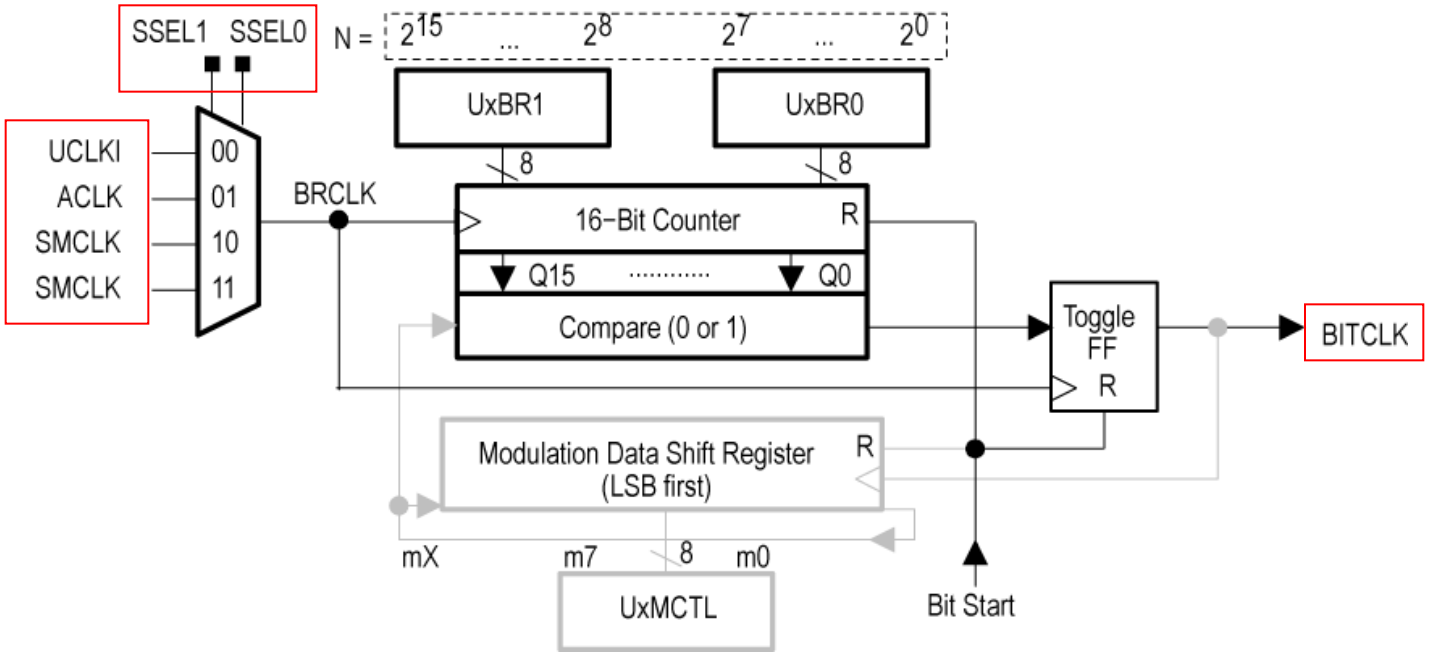


Contrôle de l'horloge de référence : générateur de Baud rate

L'horloge UCLK en mode SPI est fournie par le Master.
Lorsque MM=1, BITCLK est fourni par le générateur de baud rate de l'USART sur la broche UCLK.
Lorsque MM=0,

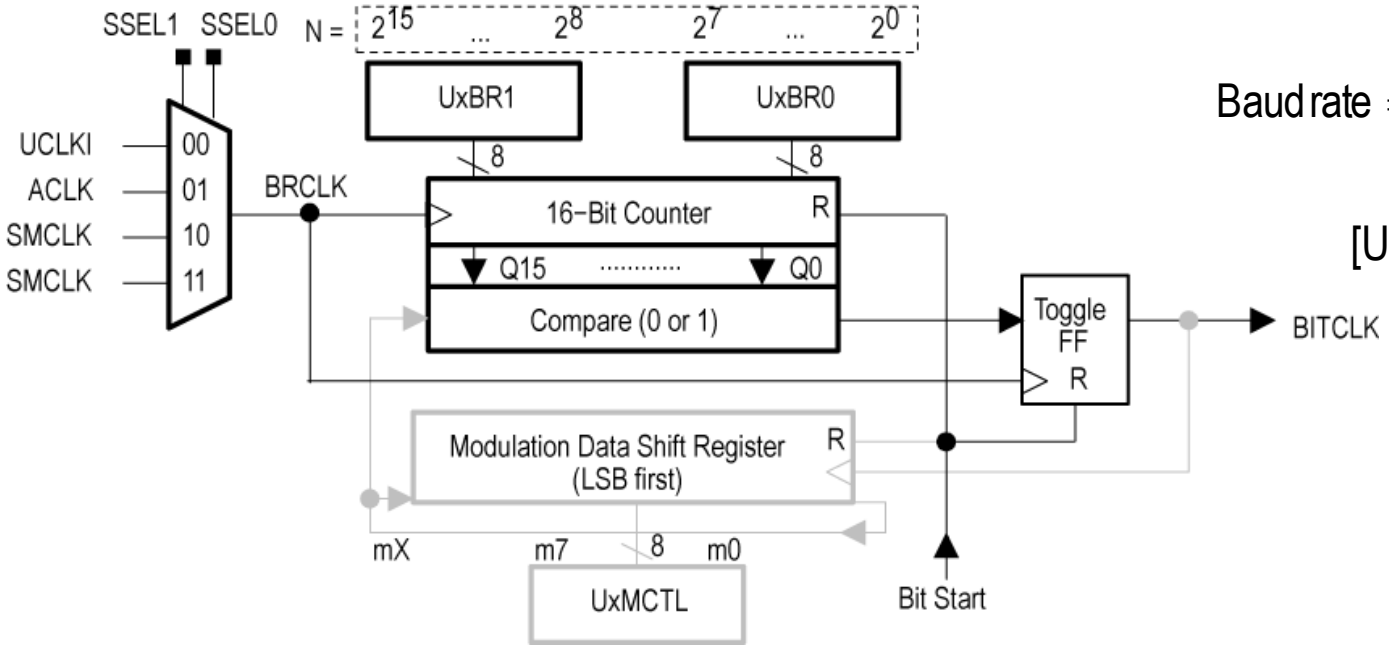
L'horloge de l'USART est fourni par le Master sur la broche UCLK. Dans ce cas le générateur de baud rate n'est pas utilisé et les bits de sélections de l'horloge SSELx ne sont pas pris en compte

Le récepteur et l'émetteur SPI opèrent en parallèle et avec la même horloge de référence pour le transfert de données



Contrôle de l'horloge de référence : générateur de Baud rate

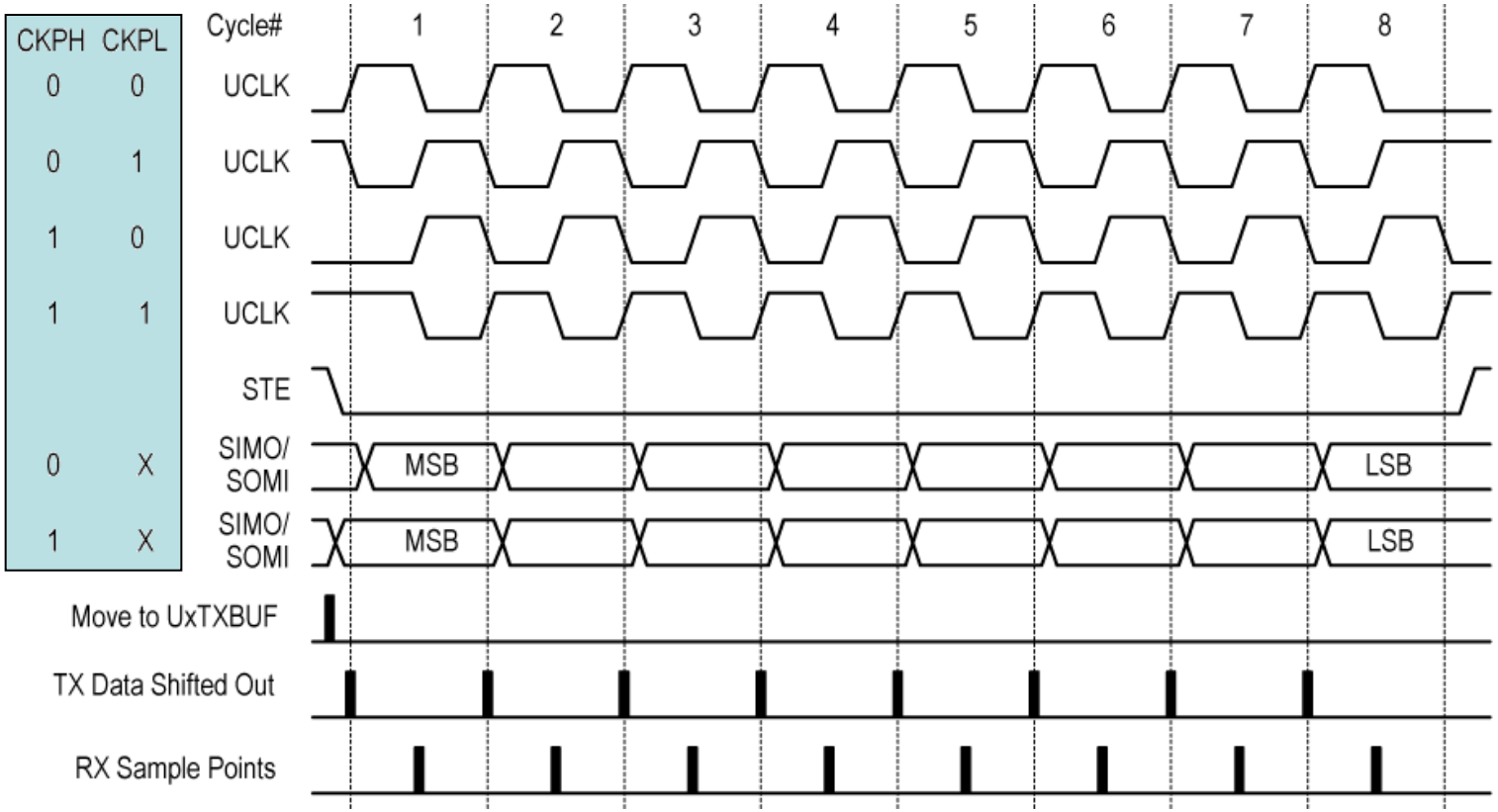
La valeur de 16 bits placée dans les registres [UxBR1 UxBR0] correspond au facteur de division de l'horloge source BRCLK.
Le baud rate maximum généré en mode Master vaut BRCLK/2.
Le baud rate maximum généré en mode Slave vaut BRCLK
Le modulateur du générateur de baud rate de l'USART n'est pas utilisé en mode SPI. Il est recommandé de lui donner la valeur 0x00.



$$\text{Baud rate} = \frac{\text{BRCLK}}{\underbrace{\text{UxBR}}_{\text{[UxBR1 UxBR0]}}}$$

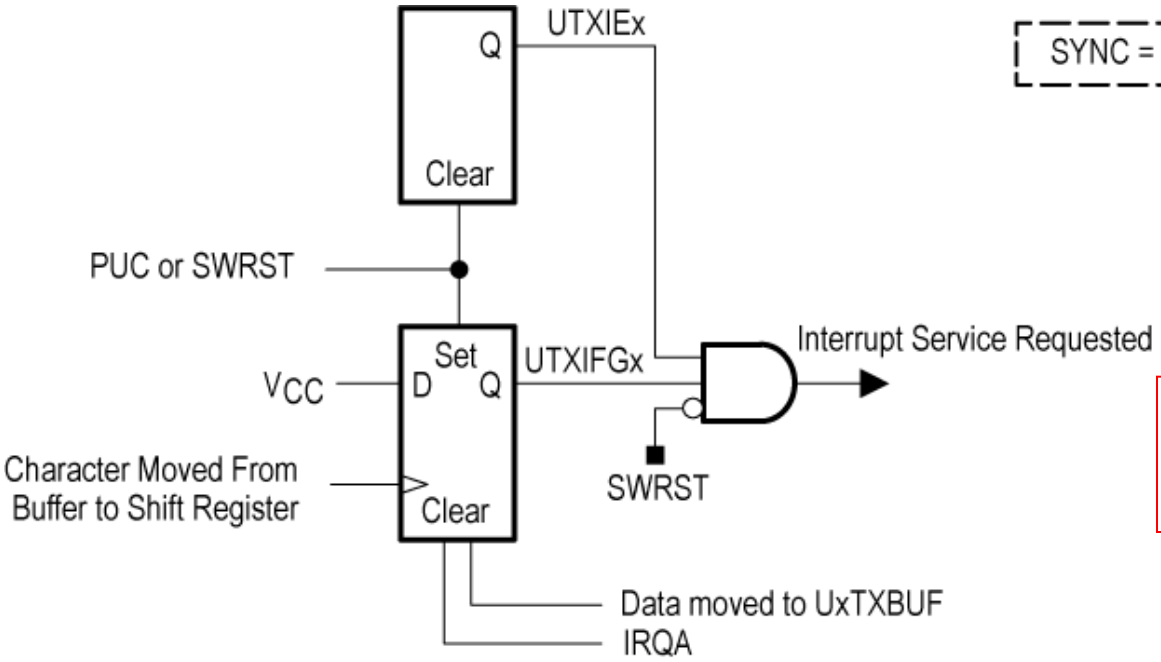
Horloge de référence : polarité et phase

La polarité et la phase de l'horloge UCLK est configurable via les bits de contrôle CKPH et CKPL.



SPI : sources d'interruptions : en transmission

Le flag d'interruption UTXIFGx est forcé à 1 par le transmetteur pour indiquer que le buffer UxTXBUF est prêt pour recevoir un nouveau caractère.
Une interruption est générée si les bits UTXIEx (Interrupt TX enable) et GIE (General Interrupt Enable) sont à 1. UTXIFGx est automatiquement remis à 0 lorsque la routine d'interruption a été exécutée ou si un caractère est écrit dans le buffer de transmission UxTXBUF.
UTXIFGx (flag d'interruption) est forcé à 1 et UTXIEx est forcé à 0 (masquage de l'interruption) après une activation de la ligne PUC ou lors d'un reset software (SWRST=1)



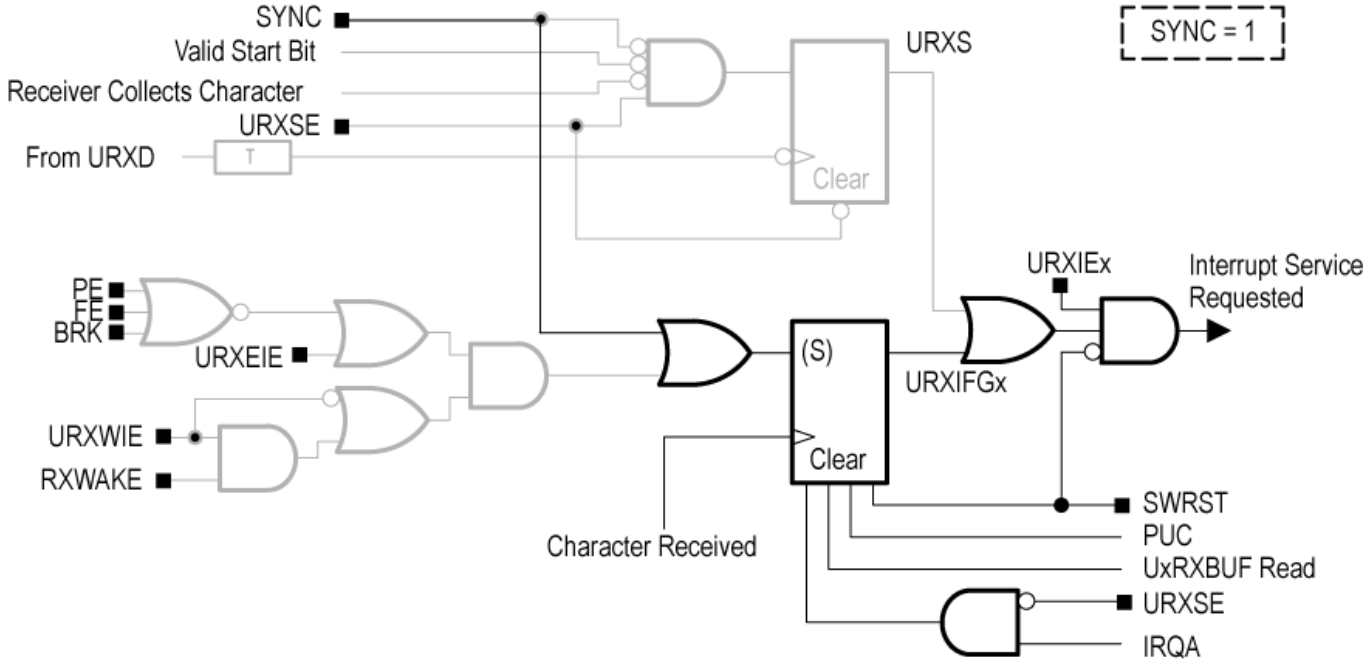
Ecrire un caractère dans le buffer UxTXBUF lorsque UTXIFGx = 0 et USPIEx = 1 peut conduire à une erreur de transmission

SPI : sources d'interruptions : en réception

Le flag d'interruption URXIFGx est forcé à 1 chaque fois qu'un caractère est reçu et chargé dans le buffer UxRXBUF

Une interruption est générée si les bits URXIEx (Interrupt RX enable) et GIE (General Interrupt Enable) sont à 1. URXIFGx est automatiquement remis à 0 lorsque la routine d'interruption a été exécutée ou si le buffer de réception UxTXBUF est lu

URXIFGx (flag d'interruption) est forcé à 1 et URXIEx est forcé à 0 (masquage de l'interruption) après une activation de la ligne PUC ou lors d'un reset software (SWRST=1)



Registre de contrôle de l'USART : UxCTL

7	6	5	4	3	2	1	0
Unused	Unused	I2C†	CHAR	LISTEN	SYNC	MM	SWRST
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-1

IC2 : UxCTL[5]
pas implanté sur le MSP430FG4617

CHAR : UxCTL[4] longueur des caractères
0 : 7 bits
1 : 8 bits

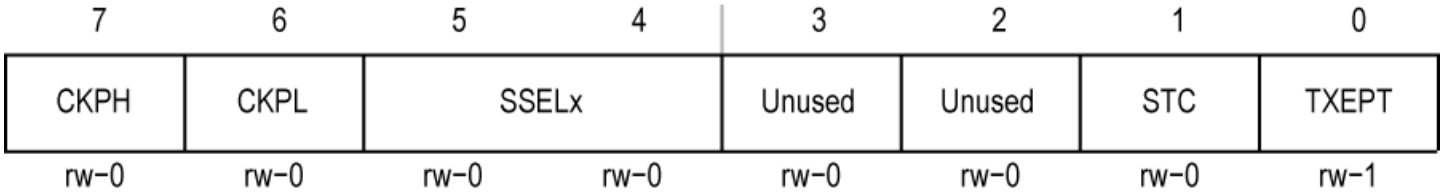
LISTEN : UxCTL[3] echo
0 : désactivé
1 : le signal transmis est bouclé en interne dans le récepteur

SYNC : UxCTL[2]
0 : mode UART
1 : mode SPI

MM : UxCTL[1]
0 : USART en mode Slave
1 : USART en mode Master

SWRST : UxCTL[0]
0 : USART activé
1 : USART forcé dans l'état RESET

Registre de contrôle de transmission de l'USART : UxTCTL



CKPH : UxTCTL[7] sélection de la phase de l'horloge UCLK
0 : les données changent sur le premier flanc de UCLK et sont capturés sur le flanc suivant
1 : les données sont capturées sur le premier flanc de UCLK et changent sur le flanc suivant

CKPL : UxTCTL[6] sélection de la polarité de l'horloge UCLK
0 : l'état inactif est l'état bas
1 : l'état inactif est l'état haut

SSELx : UxTCTL[5 4] contrôle de la transmission en mode Slave
00 : UCLK est externe (mode Slave)
01 : ACLK (mode Master)
10 : SMCLK (mode Master)
11 : SMCLK (mode Master)

STC : UxTCTL[1] contrôle de la transmission en mode Slave
0 : Mode SPI à 4 lignes (STE actif)
1 : Mode SPI à 3 lignes (STE inactif)

TXEPT : UxTCTL[0] buffer de transmission vide (pas utilisé en mode Slave)
0 : Transmission active et / ou données non encore transmises dans le buffer UxTXBUF
1 : Le buffer UxTXBUF et le registre de transmission sont vides

Registre de contrôle de réception de l'USART : UxRCTL

7	6	5	4	3	2	1	0
FE	Unused	OE	Unused	Unused	Unused	Unused	Unused
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0

FE : UxRCTL[7] Erreur de trame (conflit de bus lorsque MM=1 et STC=0). Utilisé en mode Master uniquement
0 : pas de détection de conflit
1 : Détection d'un flanc négatif sur STE

OE : UxRCTL[5] Erreur due au transfert d'un caractère dans le buffer UxRXBUF avant lecture du précédent
OE est automatiquement mis à 0 lorsque le buffer UxRXBUF est lu, ou SWRST est mis à 1
0 : pas de détection d'erreur
1 : détection d'une surécriture

Registre de contrôle de modulation de l'USART : UxMCTL

7	6	5	4	3	2	1	0
m7	m6	m5	m4	m3	m2	m1	m0
rw	rw	rw	rw	rw	rw	rw	rw

UxMCTL : UXMCTL[7 0] la modulation n'est pas utilisée en mode SPI. Ce registre doit être mis à 0x00

Registre de contrôle du Baud rate de l'USART : UxBR0 et UxBR1

7	6	5	4	3	2	1	0
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
rw	rw	rw	rw	rw	rw	rw	rw

7	6	5	4	3	2	1	0
2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8
rw	rw	rw	rw	rw	rw	rw	rw

UxBRx : [UxBR1[7 0] UxBR0[7 0]] division de l'horloge de référence pour obtenir le baud rate spécifié
En mode SPI la valeur de UxBRx doit être supérieure ou égale à 2

Buffer de réception de l'USART : UxRXBUF

7	6	5	4	3	2	1	0
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
r	r	r	r	r	r	r	r

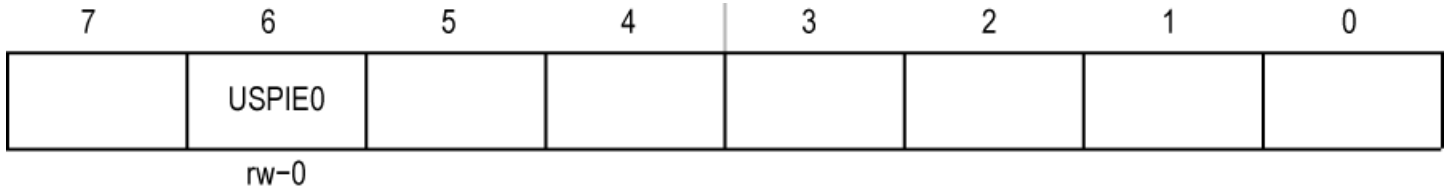
UXRXBUF : UxRXBUF[7 0] Buffer de réception contenant le dernier caractère reçu du registre à décalage RX.
La lecture du buffer UxRXBUF force à 0 les flags OE et URXIFGx.
En configuration 7 bits, les bits sont justifiés à droite, le MSB est toujours forcé à 0.

Buffer de transmission de l'USART : UxTXBUF

7	6	5	4	3	2	1	0
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
rw	rw	rw	rw	rw	rw	rw	rw

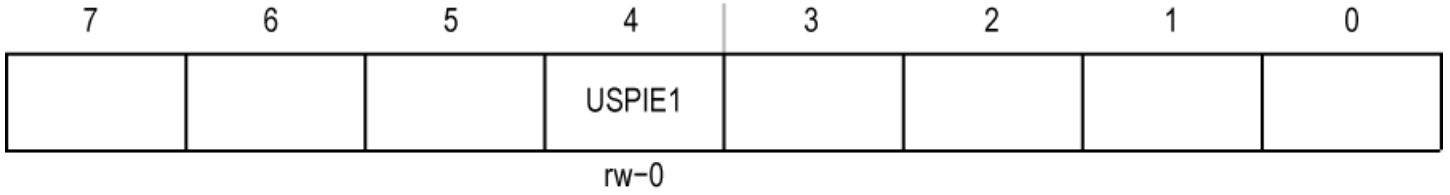
UXTXBUF : UxTXBUF[7 0] Buffer de transmission contenant le caractère courant devant être transmis.
L'écriture dans le buffer UxTXBUF force à 0 le flag d'interruption UTXIFGx.
En configuration 7 bits, les bits doivent être justifiés à gauche.

Registre de contrôle de l'activation de l'UART0 : ME1



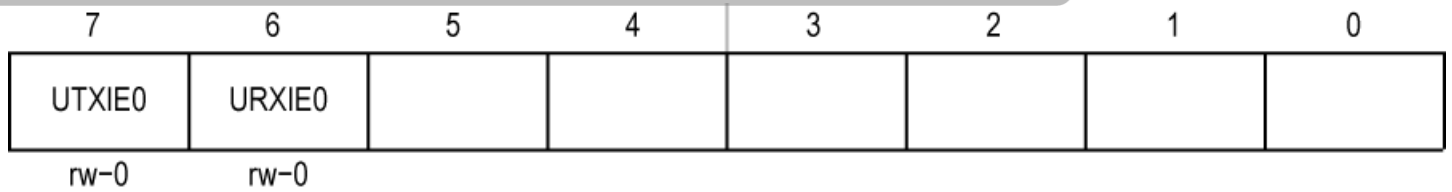
USPIE0 : ME1[6] Activation / désactivation de UART0 en mode SPI
0 : désactivation
1 : activation

Registre de contrôle de l'activation de l'UART1 : ME2



USPIE1 : ME2[6] Activation / désactivation de UART1 en mode SPI
0 : désactivation
1 : activation

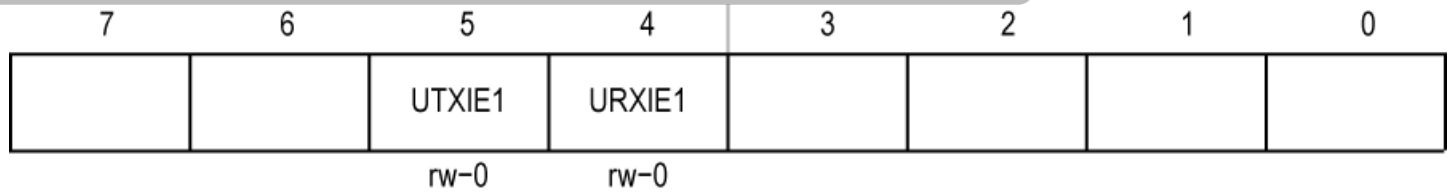
Registre de contrôle d'activation des interruptions l'UART0 : IE1



UTXIE0 : IE1[7] Interruption de transmission masquée / démasquée
0 : interruption masquée
1 : interruption démasquée

URXIE0 : IE1[6] Interruption de réception masquée / démasquée
0 : interruption masquée
1 : interruption démasquée

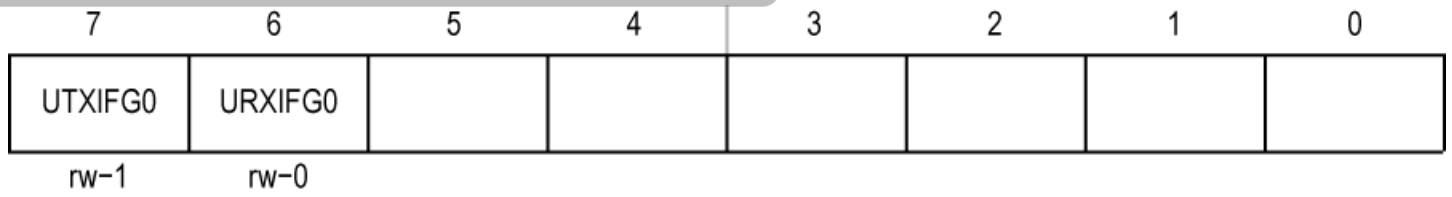
Registre de contrôle d'activation des interruptions l'UART1 : IE2



UTXIE1 : IE2[7] Interruption de transmission masquée / démasquée
0 : interruption masquée
1 : interruption démasquée

URXIE1 : IE2[6] Interruption de réception masquée / démasquée
0 : interruption masquée
1 : interruption démasquée

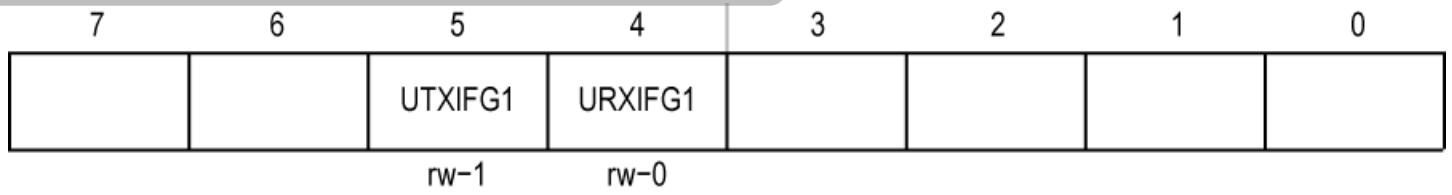
Registre des flags d' interruptions l'UART0 : IFG1



UTXIFG0 : IFG1[7] Flag d'interruption de transmission (buffer U0TXBUF vide)
0 : pas d'interruption pendante
1 : interruption pendante

URXIFG0 : IFG1[6] Flag d'interruption de réception (caractère disponible dans le buffer U0RXBUF)
0 : pas d'interruption pendante
1 : interruption pendante

Registre des flags d' interruptions l'UART1 : IFG2

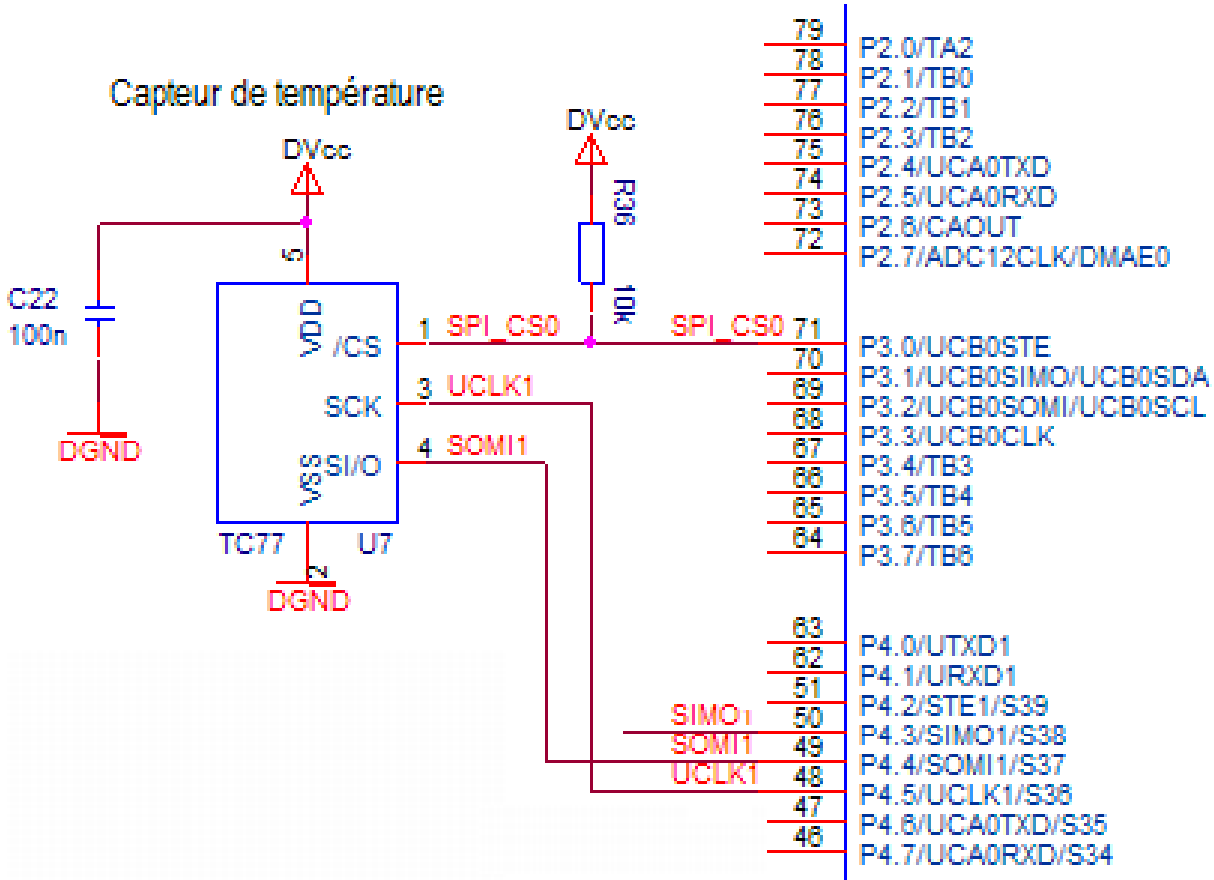


UTXIFG1 : IFG1[5] Flag d'interruption de transmission (buffer U1TXBUF vide)
0 : pas d'interruption pendante
1 : interruption pendante

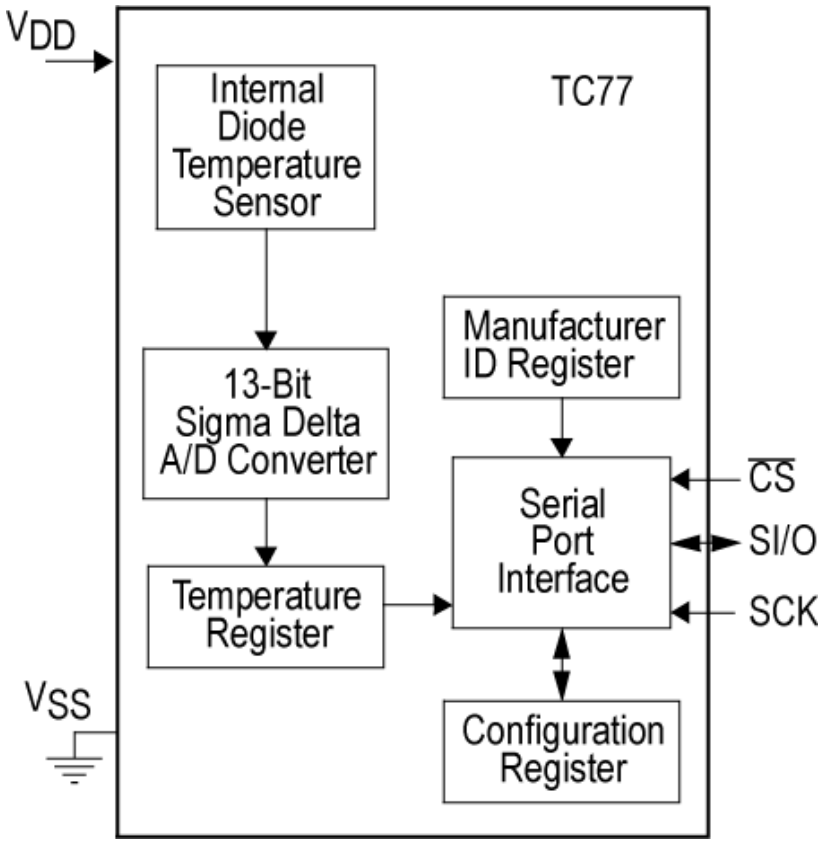
URXIFG1 : IFG1[4] Flag d'interruption de réception (caractère disponible dans le buffer U1RXBUF)
0 : pas d'interruption pendante
1 : interruption pendante

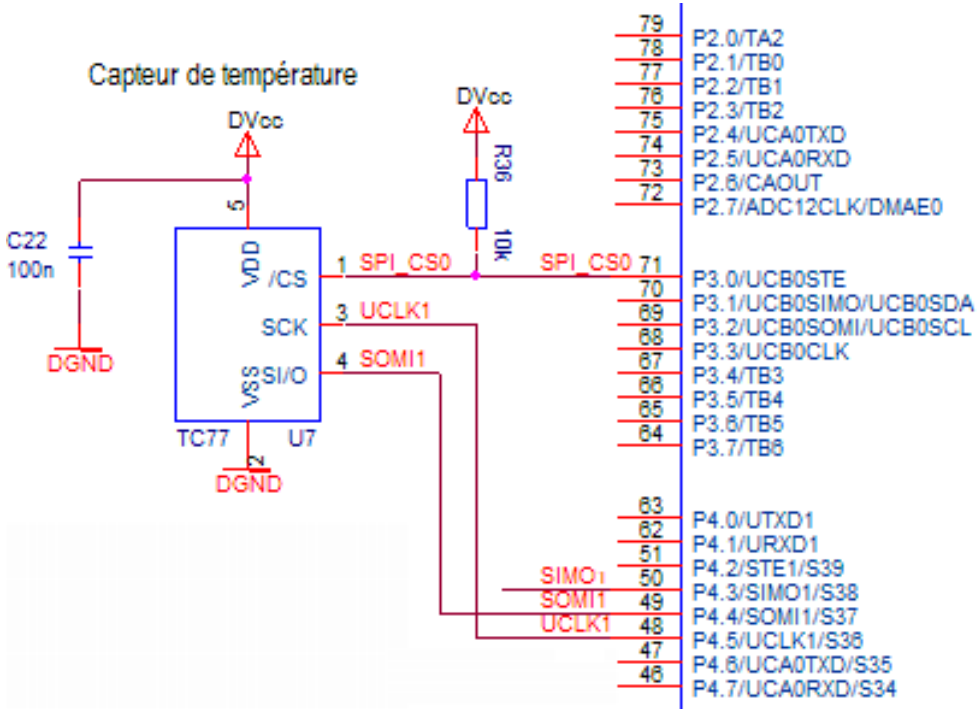
USART en mode SPI

Exemple d'application

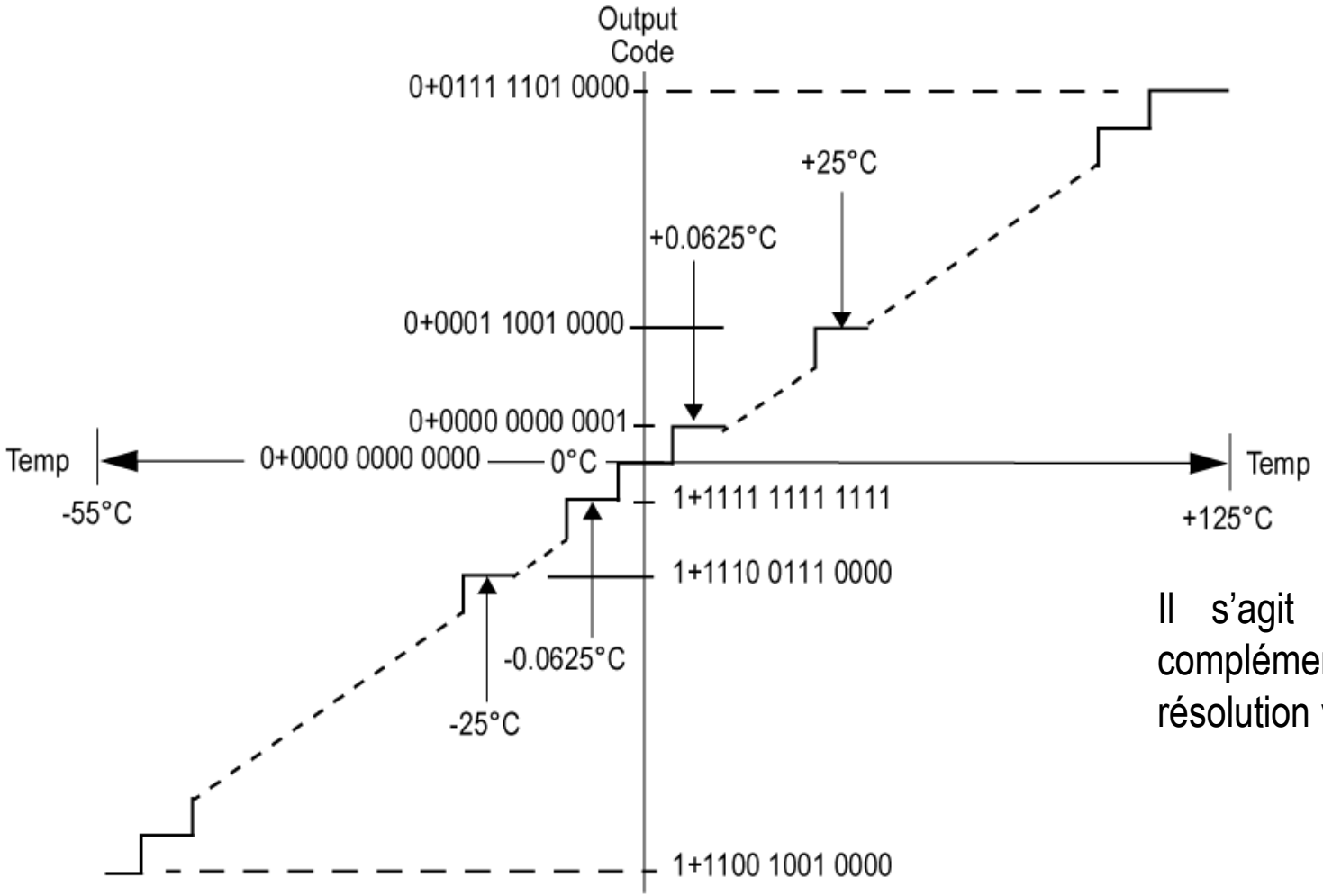


Capteur de température avec liaison SPI





Capteur de température avec liaison SPI: fonction de conversion



Il s'agit d'un code en complément à deux. La résolution vaut **0.0625°C**.

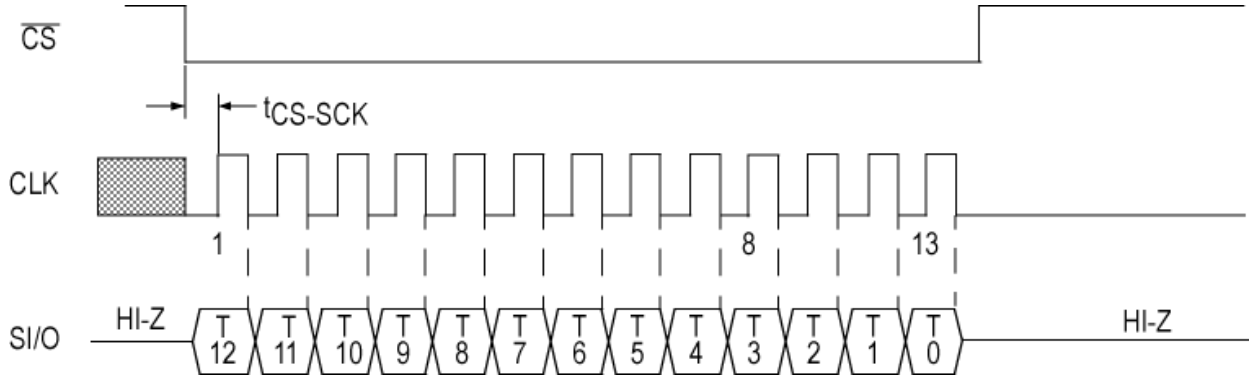
Conversion

- Les deux bits de poids faibles ne doivent pas être pris en compte (bus en haute impédance).
- Le bit 2 est toujours à 1, excepté lors de la première conversion suivant une tension d'alimentation dépassant 1.6V (superviseur de tension intégré).
- La valeur numérique utile est donc donnée sur les 13 bits de poids forts.
- Le tableau ci-dessous montre le codage de la température.

Température		Valeur numérique de sortie [LSB]				
		Binaire				Hexadécimal
+125°C		0011	1110	1000	01zz	07D0
+25°C		0000	1100	1000	01zz	0190
+0.0625°C		0000	0000	0000	11zz	0001
0°C		0000	0000	0000	01zz	0000
-0.0625°C		1111	1111	1111	11zz	1FFF
-25°C		1111	0011	1000	01zz	1E70
-55°C		1110	0100	1000	01zz	1C90

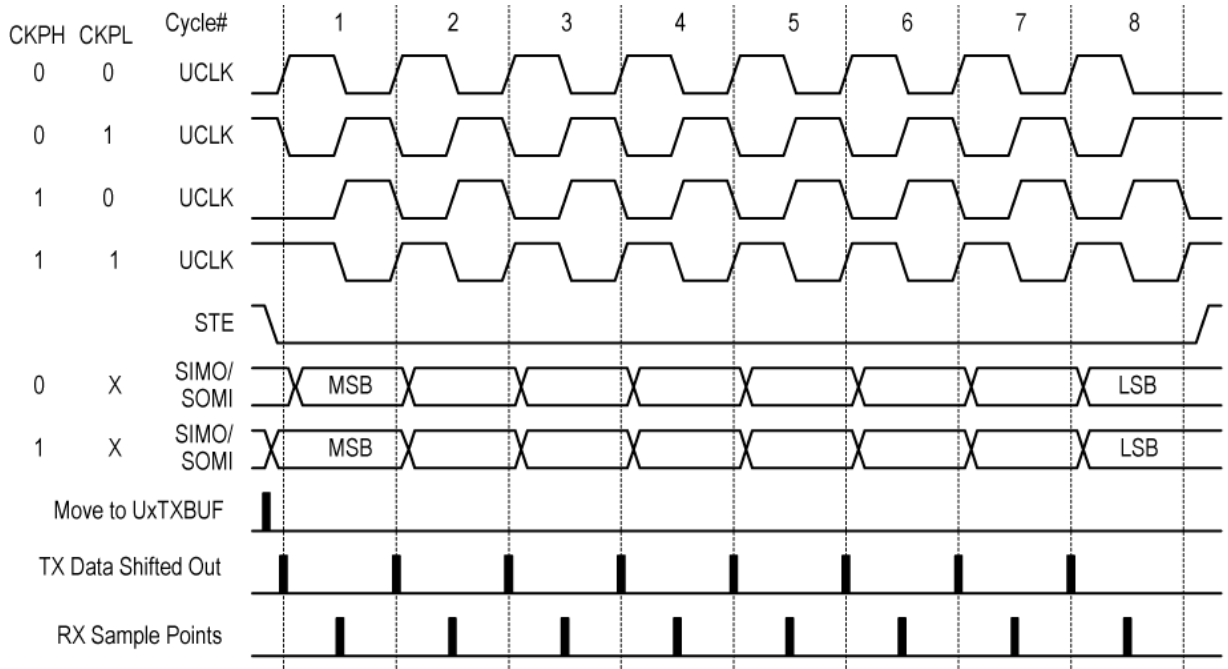
Communication

- Le transfert des données par le bus SPI peut être effectué lorsque le signal sur CS est forcé au niveau bas.
- Le bit de poids fort (MSB) est placé sur la ligne SI/0 au flanc descendant de CS, puis chaque bit est placé sur le bus au flanc descendant du CLK.
- Il faut donc 13 périodes d'horloge pour avec une transmission complète d'une donnée.
- Les données doivent être capturées sur le flanc montant de l'horloge.
- Au flanc montant du signal la communication est interrompue. Elle recommencera lors du prochain flanc descendant.



Communication

- Seule la ligne SOMI est utilisée pour les données.
- La ligne correspondant au signal est issue de la broche P3.0.
- L'horloge correspond au signal UCLK.



Initialisation de l'USART en mode SPI

```
#include "msp430xG46x.h"

// Initialisation de l'USART1 en mode SPI ************************************
void SPI_1_init(void)
{
  ME2 |= USPIE1;                    // Enable USART1 SPI mode
  U1CTL = CHAR + SYNC + MM;
  U1TCTL= SSEL0 + STC;
  U1BR0 = 0x02;
  U1BR1 = 0x00;                     // UCLK/2
  U1MCTL = 0x00;                    // no modulation

  // Configuration de la ligne CS (Chip Select) P3.0
  P3OUT |= 0x01;
  P3SEL  = 0x00;
  P3DIR |= 0x01;

  // Configuration du port P4 pour le périphérique USART1 en mode SPI
  P4DIR &= ~BIT4; // SOMI1
  P4SEL |= BIT4;
  P4DIR |= BIT5;  // UCLK1
  P4SEL |= BIT5;
}
```

Exemple de lecture du capteur

```
// Lecture de la donnée brute du capteur TC77
// 16 bits signés (3 LSBs à supprimer) unité = 0.0625 °C ************************

signed short capteurTemp_read(void)
{
  signed short data;

  P3OUT &= ~BIT0;                 // Activation du CS

  U1TXBUF = 0x00 ;                // Chargement du buffer de transmission U1TXBUF
  while (!(U1TCTL & 0x01));       // Attente de l'envoi / reception complet (8 bits)
                                  // TXEPT=1 indique que le buffer de transmission est vide
  data = U1RXBUF << 8;            // lecture des MSBs

  U1TXBUF = 0x00;                 // Chargement du buffer de transmission U1TXBUF
  while (!(U1TCTL & 0x01));       // Attente de l'envoi / reception complet (8 bits)
                                  // TXEPT=1 indique que le buffer de transmission est vide
  data = data | U1RXBUF;          // lecture des LSBs

  P3OUT |= BIT0;                  // relève enable: désactivation du CS
  return(data);                   // Retour de la valeur brute fournie par le capteur
}
```

Exemple de programme principal

```
#include "msp430xG46x.h"
#include "intrinsics.h"     // fonction intrinsèque : __no_operation()
#include "LCD.h"            // déclaration des fonctions LCD
#include <stdio.h>          // déclaration de la fonction "sprintf"

void main( void )
{
  int i;                            // variable signée de 16 bits
  unsigned int n;                   // variables non signées de 16 bits
  char str[9];                      // Tableau de 9 caractères (8 bits)

  WDTCTL = WDTPW + WDTHOLD;         // Stop watchdog timer

  LCD_init();                       // initialisation de l'affichage LCD
  SPI_1_init();  // initialisation de l'interface avec le capteur de température
  while(1)       // Boucle infinie
  {
    // Supression des 3 bits de poids faibles du mots de 16 bits issu du capteur
    n = capteurTemp_read()>>3;

    // Création d'une chaîne de caractère ASCII à afficher
    sprintf(str, "T=%4.1f C",n*0.0625);
    LCD_print(str);

    // Attente avant une nouvelle acquisition de température
    for (i=0 ; i<1000 ; i++)
      __no_operation();
  }
}
```