

Port C:

Le port C est un port d'usage général et bidirectionnel dont le sens de chaque ligne est programmé grâce au registre DDRC.

La lecture ou écriture des données de ou vers le port C fait par contre appel à deux registres : PORTC et PORTCL.

En mode Etendu et mode Test, les bus de données et d'adresse sont multiplexés sur ce port et le registre PORTC est traité en tant que case mémoire externe

Port C:

Le registre DDRC: (\$1007) Data Direction Register for Port C

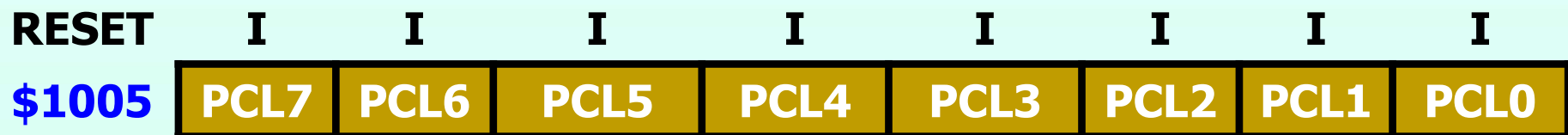
- 0: Entrée
- 1: Sortie

Le registre PORTC: Port C Data

RESET	I	I	I	I	I	I	I	I	I
\$1003	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	
Exp/test	ADDR7	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0	
	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0	

Port C:

*Le registre **PORCL**: Port C Latched*



Le registre **PORTCL** est utilisé lorsque le Port C fonctionne en mode Handshake. Lorsqu'un front actif est détecté sur l'entrée **STRA** (strobe input), les données du Port C sont mémorisées dans ce registre.

Port D:

Le port D est un port d'entrées sorties 8 bits à usage général partagé avec les interfaces séries asynchrones (SCI) et synchrones (SPI).

Tant que ces interfaces ne sont utilisées, le sens de travail des lignes du port est programmé par les bits correspondants du registre DDRD .

La sortie de données se fait par écriture de registre PORTD et c'est dans ce même registre que l'on vient lire l'état des broches qui sont positionnées en entrées.

Port D:

*Le registre **DDRD**: (\$1009) Data Direction Register for Port D*

DDRD5 à DDRD0: définit le sens de transfert

0: Entrée

1: Sortie

DDRD7 à DDRD6: ne sont pas utilisés

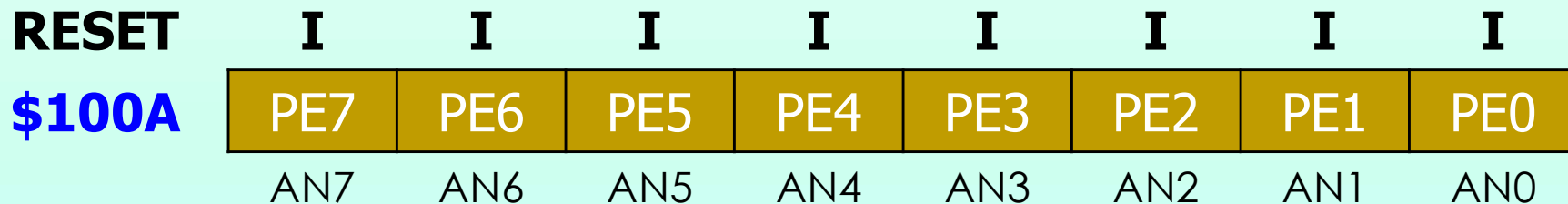
*Le registre **PORTD**: Port D Data*

RESET	-	-	I	I	I	I	I	I
\$1008	0	0	PD5	PD4	PD3	PD2	PD1	PD0
	-	-	PD5	PD4	PD3	PD2	PD1	PD0
			/SS	SCK	MOSI	MISO	TX	RX

Port E:

Le port E est un port d'entrées 8 bits à usage général partagé avec le convertisseur analogique-numérique.

*Le registre **PORTE**: Port E Data*



Liaison série asynchrone : (Asynchronous Serial Communications Interface)

Cette interface permet au microcontrôleur d'échanger des données avec des circuits périphériques utilisant une liaison série asynchrone (ordinateurs, tables traçantes, modem, instruments de musique...).

Les circuits internes de la SCI permettent :

- **des échanges asynchrones bidirectionnels en full duplex**
- **de choisir la vitesse de transmission**
- **de choisir le format de transmission**
- **de choisir le mode d'activation de la partie réception (classique ou réveil)**
- **de gérer les erreurs de transmission**
- **d'indiquer la fin de transmission**

Liaison série asynchrone :

La liaison est assurée par deux broches RxD – Receive Data- (PD0) et TxD – Transmit Data- (PD1) du port D.

Après un RESET, la SCI est inhibée et le port D est configuré comme un port d'usage général.

- La SCI peut être activée en réception et/ou en émission (half ou full duplex).
- Lorsque la SCI est activée en réception, PD0 devient l'entrée de réception des données RXD.
- Lorsque la SCI est activée en émission, PD1 devient la sortie de transmission des données TXD.

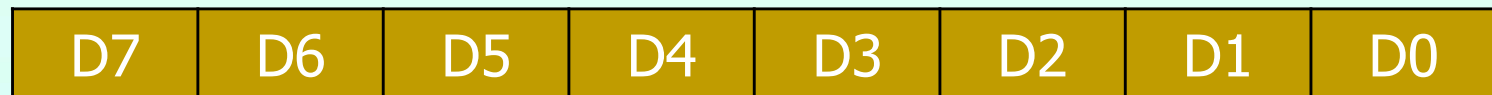
Liaison série asynchrone : *Registres associés*

Le registre SCDR :

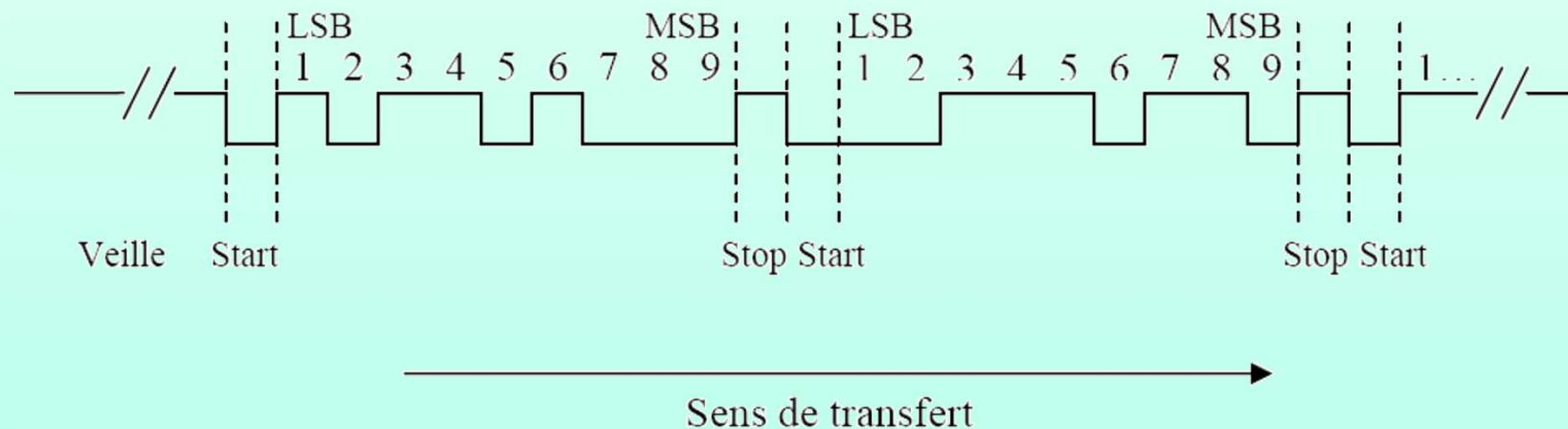
Ce registre, accessible en lecture - écriture, est un registre 8 bits dans lequel on vient écrire la donnée à transmettre ou dans lequel on vient lire la donnée reçue.

RESET

\$102F



Lors d'une transmission, c'est le bit de poids faible **D0** qui apparaît en **premier** sur la ligne après le bit de **start**.



Liaison série asynchrone : *Registres associés*

Le registre BAUD :

Le récepteur et l'émetteur travaillent à la même vitesse, dérivée de l'horloge interne du microcontrôleur.

Le registre 8 bits BAUD, accessible en lecture – écriture permet de choisir cette vitesse.

RESET	0	0	0	0	0	0	0	0
\$102B	0	0	SCP1	SCP1	0	SCR2	SCR1	SCR0

Liaison série asynchrone : *Registres associés*

Le registre BAUD :



Débit de transmission	SCP1	SCP0
$D_M=125\ 000$ bauds	0	0
$D_M=41\ 666$ bauds	0	1
$D_M=31\ 250$ bauds	1	0
$D_M=9\ 600$ bauds	1	1

Liaison série asynchrone : *Registres associés*

Le registre BAUD :

\$102B	0	0	SCP1	SCP1	0	SCR2	SCR1	SCR0
						↓	↓	↓
Débit de transmission en bauds								$D_M=9\ 600$ bauds
$D=D_M/1$	0	0	0	9600 bauds				
$D=D_M/2$	0	0	1	4800 bauds				
$D=D_M/4$	0	1	0	2400 bauds				
$D=D_M/8$	0	1	1	1200 bauds				
$D=D_M/16$	1	0	0	600 bauds				
$D=D_M/32$	1	0	1	300 bauds				
$D=D_M/64$	1	1	0	150 bauds				
$D=D_M/128$	1	1	1	75 bauds				

Liaison série asynchrone : *Registres associés*

Le registre SCSR :

Ce registre 8 bits, accessible uniquement en lecture, permet de surveiller le bon déroulement des opérations de la SCI.

RESET	1	1	0	0	0	0	0	0
\$102E	TDRE	TC	RDRF	IDLE	OR	NF	FE	0

	<i>Registre d'émission vide (Transmit Data register Empty Bit)</i>
0	Registre occupé
1	Registre vide (RAZ par la lecture du SCSR puis lecture du SCDR)

Liaison série asynchrone : *Registres associés*

Le registre SCSR :

\$102E

TDRE	TC	RDRF	IDLE	OR	NF	FE	0
------	----	------	------	----	----	----	---

<i>Indicateur de fin de transmission (Transmit Complete Bit)</i>	
0	Transmission en cours (<i>Transmetteur occupé</i>)
1	Transmission terminée (<i>Transmetteur en attente</i>) (RAZ par lecture du SCSR puis lecture du SCDR)

Liaison série asynchrone : *Registres associés*

Le registre SCSR :

\$102E



*Indicateur d'état du registre de réception de données
(Receive Data Register Full Bit)*

0	Registre de données SCDR vide (<i>Réception incomplète</i>)
1	Registre de données SCDR plein (<i>Réception terminée</i>) (RAZ par lecture du SCSR puis lecture du SCDR)

Liaison série asynchrone : *Registres associés*

Le registre SCSR :

\$102E

TDRE	TC	RDRF	IDLE	OR	NF	FE	0
------	----	------	------	----	----	----	---

Indicateur de détection d'un état de veille (Idle-Line Detect Bit)

0	Ligne RxD en activité
1	Ligne RxD en attente au niveau 1 (RAZ par lecture du SCSR puis lecture du SCDR)

Liaison série asynchrone : *Registres associés*

Le registre SCSR :

\$102E

TDRE	TC	RDRF	IDLE	OR	NF	FE	0
------	----	------	------	----	----	----	---

<i>Indicateur de dépassement (Overrun Error Bit)</i>	
0	Pas de surcharge
1	Surcharge du registre de réception (RAZ par lecture du SCSR puis lecture du SCDR)

Liaison série asynchrone : *Registres associés*

Le registre SCSR :

\$102E



Indicateur de bruit de transmission (Noise Flag)

0 Réception non bruitée

1 Réception bruitée (RAZ par lecture du SCSR puis lecture du SCDR)

Liaison série asynchrone : *Registres associés*

Le registre SCSR :

\$102E



Erreur de trame (Framing Error Bit)

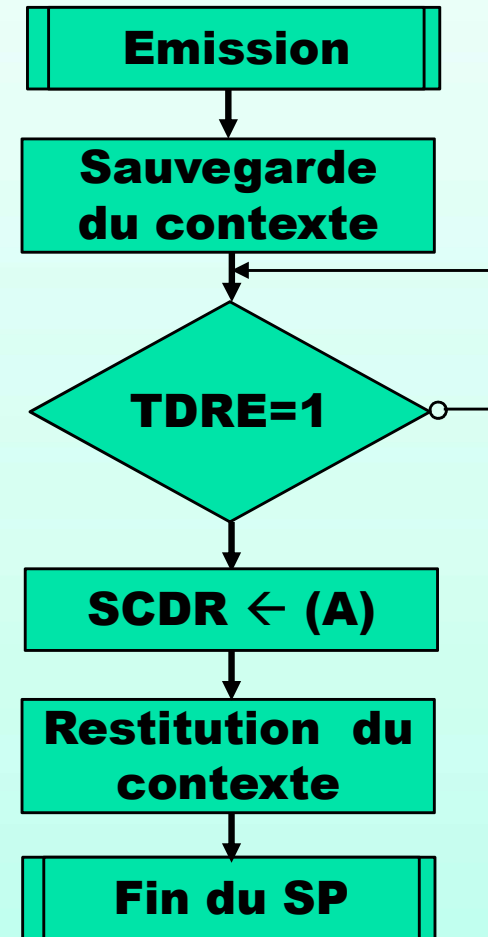
0	Format correct
1	Format incorrect (RAZ par lecture du SCSR puis lecture du SCDR)

Liaison série asynchrone : *Fonctionnement de l'interface*

Fonctionnement en émission:

Envoi

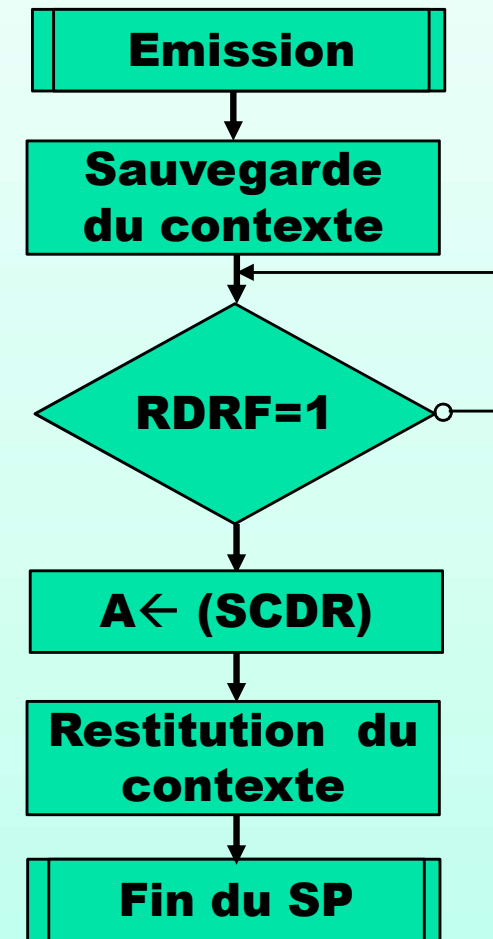
```
PSHX
LDX #SCSR
BRCLR 0,X,$80 Envoi
STAA SCDR
PULX
RTS
```



Liaison série asynchrone : *Fonctionnement de l'interface*

Fonctionnement en émission:

Attente PSHX
 LDX #SCSR
 BRCLR 0,X,\$80 attente
 LDAA SCDR
 PULX
 RTS



Liaison série asynchrone : *Registres associés*

le registre SCCR1 :

Ce registre huit bits, accessible en lecture – écriture, permet de définir le format de transmission et la procédure de réveil de réception.

RESET	-	-	0	0	0	0	0	0
\$102C	R8	T8	0	M	WAKE	0	0	0

Liaison série asynchrone : *Registres associés*

le registre SCCR1 :

\$102C

R8	T8	0	M	WAKE	0	0	0
----	----	---	---	------	---	---	---

	<i>9 bits</i>
R8	Réception sur 9 bits
T8	Emission sur 9 bits

Liaison série asynchrone : *Registres associés*

le registre SCCR1 :

\$102C



Format de la trame

0	1 bit de start + 8 bits de données + 1 bit de stop
1	1 bit de start + 9 bits de données + 1 bit de stop

Liaison série asynchrone : *Registres associés*

le registre SCCR1 :

\$102C



Mode de réveil du récepteur

0	RxD à 1 pendant au moins 1 trame de 10 ou 11 bits
1	MSB du mot à 1 ⇔ adresse

Liaison série asynchrone : *Registres associés*

le registre SCCR2:

Ce registre 8 bits, accessible en lecture – écriture, permet de valider ou inhiber certaines fonctions de l'interface.

\$102D



Autorisation des interruptions

0	Interruption inhibées
1	Interruption si TDRE =1

Liaison série asynchrone : *Registres associés*

le registre SCCR2:

\$102D



Autorisation de l'interruption « transmission complète »

0	Interruption inhibées
1	Interruption si TC =1

Liaison série asynchrone : *Registres associés*

le registre SCCR2:

\$102D



Autorisation des interruptions en réception

0	Interruption inhibées
1	Interruption si RDRF =1

Liaison série asynchrone : *Registres associés*

le registre SCCR2:

\$102D



Autorisation d'interruption « ligne en attente »

0	Interruption inhibée
1	Interruption dès que la ligne est à « MARK » (10 ou 11 bits à 1)

Liaison série asynchrone : *Registres associés*

le registre SCCR2:

\$102D



Activation de la SCI

SCI désactivée (PD0 et PD1 E/S usage général)	0	0
Half duplex en réception (PD1 usage général)	0	1
Half duplex en émission (PD0 usage général)	1	0
Full duplex	1	1

Liaison série asynchrone : *Registres associés*

le registre SCCR2:

\$102D



Mode d'activation du récepteur

UART Classique

0

Mode réveil

1

Liaison série asynchrone : *Registres associés*

le registre SCCR2:

\$102D

TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
-----	------	-----	------	----	----	-----	-----

Emission d'un BREAK

Opération normale en réception

Emission en continu d'un BREAK (10 ou 11 bits à 1)

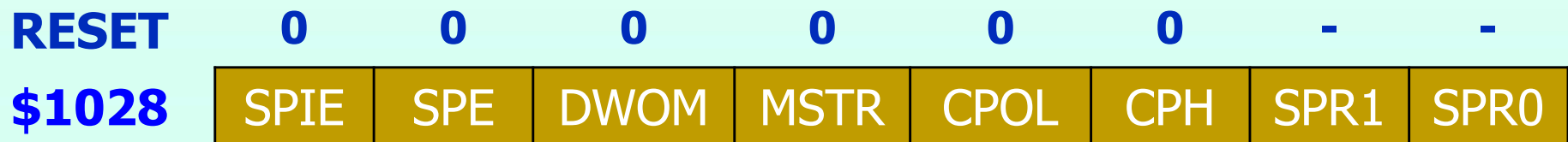
0

1

Liaison série asynchrone : *Registres associés*

Le registre SPCR:

La liaison série asynchrone, tout comme la liaison série synchrone, peut être configurée pour disposer de sorties de type « drain ouvert ». Le choix se fait en positionnant le bit DWOM du registre SPCR et influence le comportement de l'ensemble du port D.



<i>Configuration matérielle de sortie du PORT D</i>	
0	Sorties de type CMOS
1	Sorties à « drain ouvert » nécessitant des résistances de Pull up

Le microcontrôleur 68HC11:

[6]

Application:

1. Ecrire le sous-programme "Init_SCI" pour la configuration suivante:
Vitesse de 1200 Bauds, Données :8 bits, 1 bit de stop
1. Ecrire le SP pour émettre un caractère « Out_Char »
2. Ecrire le SP pour recevoir un caractère « In_Char »
3. Ecrire le programme pour recevoir une chaîne de « STRLEN » caractères

* Define the internal registers used in the SCI

* *****

```
RAM EQU $0000 ; Start of Ram
STK EQU $0041 ; User Stack RAM
PROG EQU $0100 ; Program space in RAM
EEPROM EQU $B600 ; Start of EEPROM
REGS EQU $1000 ; Internal Register Block
BAUD EQU $102B ; Baud-rate register
DDRD EQU $1009 ; Port D Data Direction register
SCCR1 EQU $102C ; SCI Control Register-1
SCCR2 EQU $102D ; SCI Control Register-2
SCSR EQU $102E ; SCI Status Register
SCDR EQU $102F ; SCI Data Register
```
