



Ecole Supérieure
des Techniques Industrielles
et des Textiles

52, allée Lakanal - BP 209
59654 Villeeneuve d'Ascq Cedex
Tél 33 (0)3.20.79.90.10
Fax 33 (0)3.20.79.90.11

Microcontrôleur

Motorola

68 HC 11

SOMMAIRE

1. INTRODUCTION.....	3
2. DEFINITION	4
3. STRUCTURE DES MICROCONTROLEURS 68HC11.....	5
4. LES ACCUMULATEURS INTERNES	15
5. MODELE DE PROGRAMMATION	17
6. PLAN MEMOIRE DU MCU.....	18
7. GESTION DES E/S PARALLELES.....	19
8. LA LIAISON SERIE ASYNCHRONE (SCI).....	21
9. LA LIAISON SERIE SYNCHRONE (SPI).....	28
10. LE TIMER.....	31
11. LE CONVERTISSEUR ANALOGIQUE/DIGITAL.....	39
12. ANNEXES	41

1. INTRODUCTION

1.1 Définition

Un microcontrôleur se présente sous la forme d'un circuit électronique intégré réunissant un certain nombre d'éléments construit autour d'un microprocesseur. Il effectue entre autres :

- opérations arithmétiques (addition, multiplication...) et logiques (ET, OU, ...),
- l'exécution d'instructions stockées dans sa mémoire programme,
- le dialogue avec des éléments électroniques extérieurs (échanges de données numériques et analogiques, génération de signaux, ...)

1.2 Modèles

Il existe une multitude de modèles de microcontrôleurs. Ce vaste choix permet de trouver le type de modèle le plus adapté à ce que l'on veut faire. Les microcontrôleurs ne sont généralement pas très chers, comparé aux processeurs dédiés au calcul très rapide (type PENTIUM). Les plus courants commencent à 10 €, tandis que les plus chers atteignent plus de 150 €. Les microcontrôleurs se distinguent par une grande caractéristique : la taille de sa mémoire de calcul. Quelques microcontrôleurs disponibles sur le marché sont :

ATMEL : AT90S8515 ; MICROSHIP : 16C71 ; MOTOROLA : 68HC5, **68HC11**, 68HC16, 68340... ; SEAGATE-THOMSOM : ST6, ST9... ; INTEL : 8031, 8051...

1.3 Utilisation

Les systèmes intelligents automatiques, intégrant une technologie à base de microcontrôleur, sont dans tous les domaines de la vie quotidienne :

- **Télécommunications** : carte FAX/MODEM, minitel, téléphones portables, interfaces homme machine, gestion d'écrans graphiques...
- **Industriels** : automates programmables, contrôle/régulation de processus, supervision...
- **Commercial** : électroménager, domotique...
- **Automobile** : ABS, calculateur d'injection, tableau de bord, contrôle des sièges...
- **Militaire et spatial** : sonde, lanceurs de fusées, missile, robots...
- **Loisirs** : concours robotique ...

Le microcontrôleur est dimensionné pour « l'embarqué ». Il est beaucoup plus facile à mettre en œuvre qu'un microprocesseur. Il est conçu pour se suffire à lui-même ; un microprocesseur a besoin de périphériques pour pouvoir fonctionner contrairement au microcontrôleur qui lui possède sa propre mémoire et le programme pour exécuter les tâches. De part le nombre important d'entrées et sorties, le microcontrôleur trouve également son rôle dans le « contrôle commande des processus ».

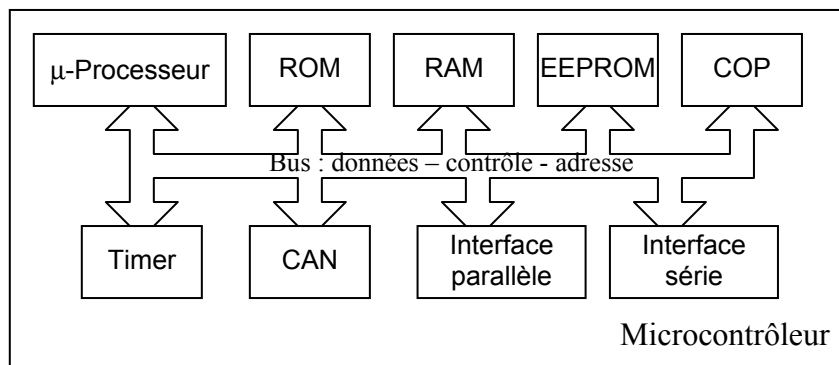
Avantages	Inconvénients
<ul style="list-style-type: none">- Encombrement réduit,- Circuit imprimé peu complexe,- Faible consommation,- Coût réduit.	<ul style="list-style-type: none">- Système de développement onéreux,- Programmation nécessitant un matériel adapté

2. DEFINITION

Un microcontrôleur (MCU¹) est un composant électronique qui intègre :

- Un microprocesseur (MPU²) qui effectue des opérations logiques et arithmétiques,
- Un ensemble de périphériques :
 - de la mémoire de données : RAM³ et EEPROM⁴
 - de la mémoire programme : ROM⁵, EPROM⁶
 - des interfaces parallèles pour la connexion des E/S,
 - des interfaces séries (synchrone et asynchrone) pour le dialogue avec d'autres composants
 - des convertisseurs digital/analogique pour le traitement de signaux analogiques
 - des timers pour générer ou mesurer des signaux analogiques

Le schéma fonctionnel d'un microcontrôleur peut être représenté par l'architecture suivante :



¹ **MCU** : Micro Controller Unit : Unité effectuant les opérations logiques et arithmétiques du calculateur, inclut les sous-ensembles de contrôle des périphériques. Peut fonctionner en monopuce sans élément annexe.

² **MPU** : Micro Processor Unit : Unité centrale effectuant toutes les opérations logiques et arithmétiques nécessaire au calculateur. Ne comporte aucun élément annexe et nécessite des circuits périphériques pour fonctionner (RAM, EPROM, ...).

³ **RAM** : Random Access Memory : Mémoire à accès aléatoire en lecture écriture. Le contenu d'une RAM est perdu lors de la coupure de la tension d'alimentation.

⁴ **EEPROM** : Electrically Erasable Programmable Read Only Memory : Mémoire pouvant être effacée électriquement et reprogrammable. Utilisation en lecture seule par l'unité centrale. Le contenu est conservé après coupure de l'alimentation.

⁵ **ROM** : Read Only Memory : Mémoire à lecture uniquement. La programmation est effectuée lors de la fabrication. Le contenu d'une ROM est conservé après coupure de la tension d'alimentation et ne peut être effacé.

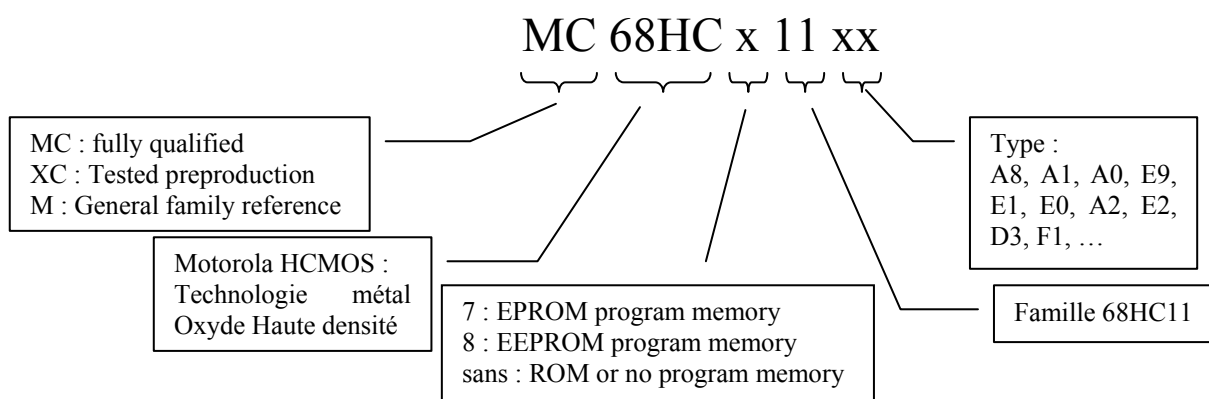
⁶ **EPROM** : Erasable Programmable Read Only Memory : Mémoire effaçable et reprogrammable. Utilisation en lecture seule par l'unité centrale. Effacement par rayon UV. Le contenu est conservé après coupure de l'alimentation.

3. STRUCTURE DES MICROCONTROLEURS 68HC11

3.1 Description générale

La famille des microcontrôleurs 68HC11 est construite autour d'un MCU fonctionnant sur un bus de données de 8 bits. Les circuits peuvent fonctionner à des fréquences d'horloge pouvant atteindre 16 MHz. Le système de mémoire associé au MPU est variable selon le type de produit et comprend une zone RAM, une ROM ou EPROM ainsi qu'une EEPROM. Le 68HC11 adresse jusqu'à 64 ko de mémoire. Le jeu d'instructions est dérivé de ces ancêtres (6801, 6805, 6809).

□ Référence des microcontrôleurs



□ Produits de la famille 68HC11

Type de MCU	EPROM (en ko)	ROM (en ko)	EEPROM	RAM	Commentaires
MC68HC11A8	-	8	512	256	Produit de base de la famille HC11.
MC68HC11A1	-	-	512	256	Identique au A8 avec la ROM désactivées.
MC68HC11A0	-	-	-	256	Identique au A8 avec la ROM et EEPROM désactivées.
MC68HC811A8	-	-	8ko + 512	256	Version A8 avec émulation EEPROM.
MC68HC11E9	-	12	512	512	Quatre entrées digitales Input Capture.
MC68HC11E1	-	-	512	512	Identique au E9 avec la ROM désactivée.
MC68HC11E0	-	-	-	512	Identique au E9 avec ROM et EEPROM désactivées.
MC68HC11E2	-	-	2k	256	Pas de ROM.
MC68HC711E9	12	-	512	512	Version programmable du E9.
MC68HC11D3	-	4	-	192	Version 40 broches.
MC68HC711D3	4	-	-	192	Version programmable du D3
MC68HC11F1	-	-	512	1 ko	Haute performance, bus Adresses et données démultiplexés, CS, boîtier 68 broches.
MC68HC11K4	-	24	640	768	Espace adressable > 1 Mo, PWM et CS Inclus, boîtier 84 broches.
MC68HC711K4	24	-	640	768	Version programmable du K4.
MC68HC11L6	-	16	512	512	Similaire au E9 avec ROM plus importante, I/O supplémentaires, boîtier 64 et 68 broches.
MC68HC711L6	16	-	512	512	Version programmable du L6.

□ Périphériques intégrés

La plupart des fonctions périphériques sont intégrées dans le MCU :

- un convertisseur analogique numérique, multiplexé sur 8 canaux, permet le traitement des grandeurs analogiques sur 8 bits,
- le circuit d'interface asynchrone (SCI⁷) de type RS232 : communication extérieure
- un port de communication synchrone (SPI⁸) : communications avec périphériques
- un compteur 16 bits pour la génération d'impulsions ainsi que la mesure de temps avec la précision donnée par le quartz de l'horloge de base.

□ Protection

La famille des 68HC11 dispose de protections internes qui permettent de garder le contrôle du MPU en cas de défaillance logicielle ou du système. La fonction COP (chien de garde) assure la surveillance de l'exécution correcte du programme, l'exécution des codes illégaux étant détectée par une interruption non masquable $\overline{\text{XIRQ}}$. En outre, le MCU effectue un reset automatique dans le cas où l'horloge du système tomberait à une fréquence inférieure à un seuil défini par l'utilisateur.

□ Données

Les MCU de cette famille supportent les types de données suivantes :

- **Bit**,
- **entiers** signés et non signés sur 8 et 16 bits,
- **fractions** non signés sur 16 bits
- **adresse** sur 16 bits

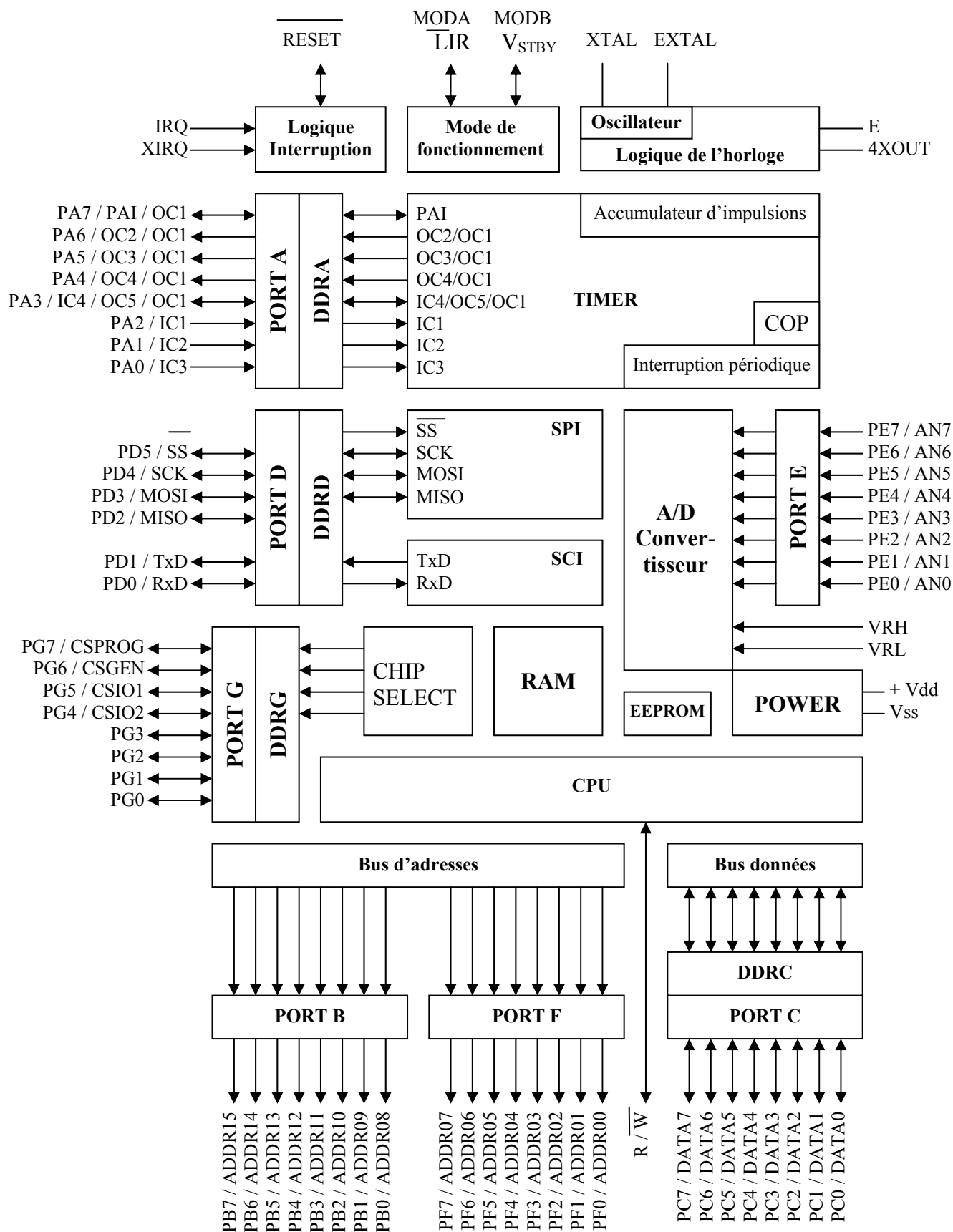
Un octet peut être situé à n'importe quelle location mémoire dans l'espace adressable du MCU. Un mot est composé de deux octets consécutifs de telle sorte que l'octet le plus significatif se trouve à la valeur basse de l'adresse

□ Schéma bloc du MC68HC11F1 (voir page suivante)

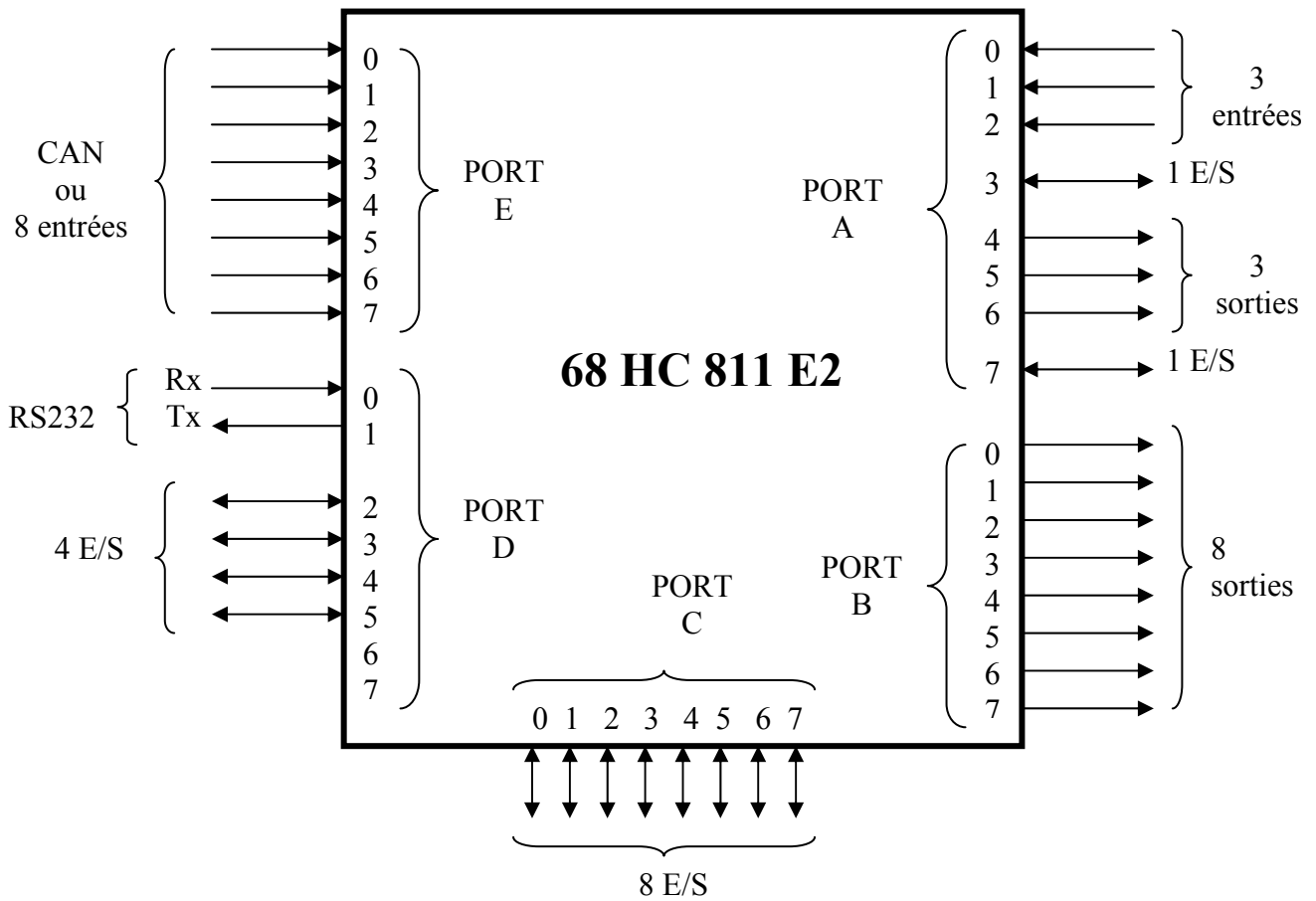
Les blocs de ce schéma sont validés ou inhibés selon le type de produit choisi, le système pouvant être étendu par le type de périphériques RAM/EPROM utilisé.

⁷ SCI : Serial Communication Interface : Interface de communication série

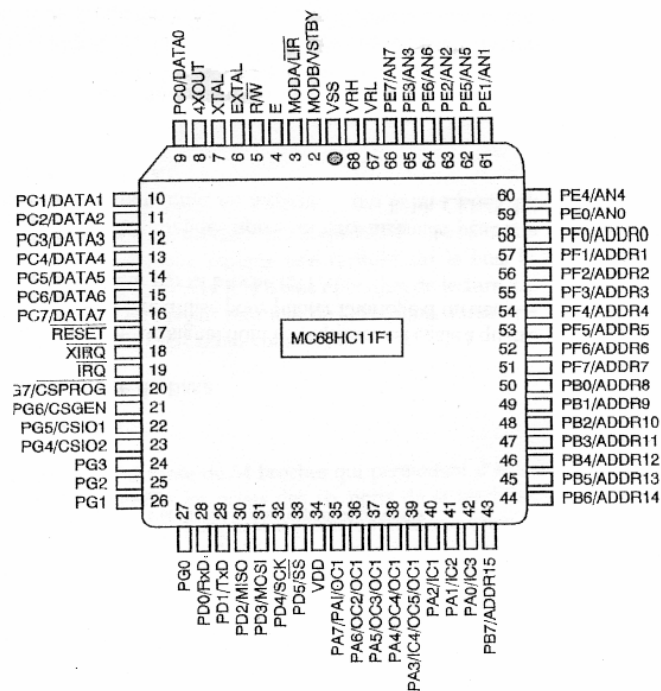
⁸ SPI : Serial Peripheral Interface : Interface série pour les périphériques du MCU



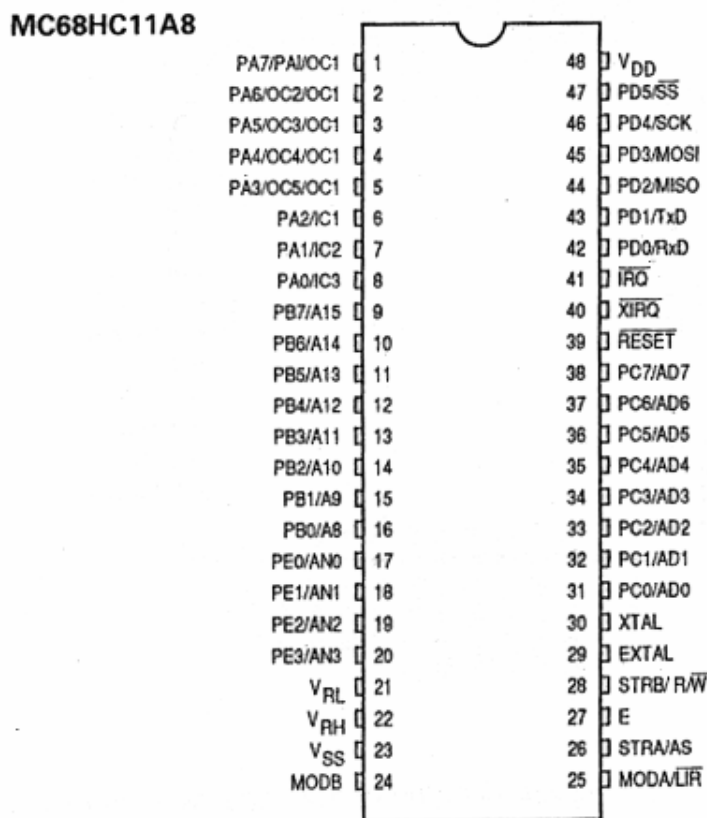
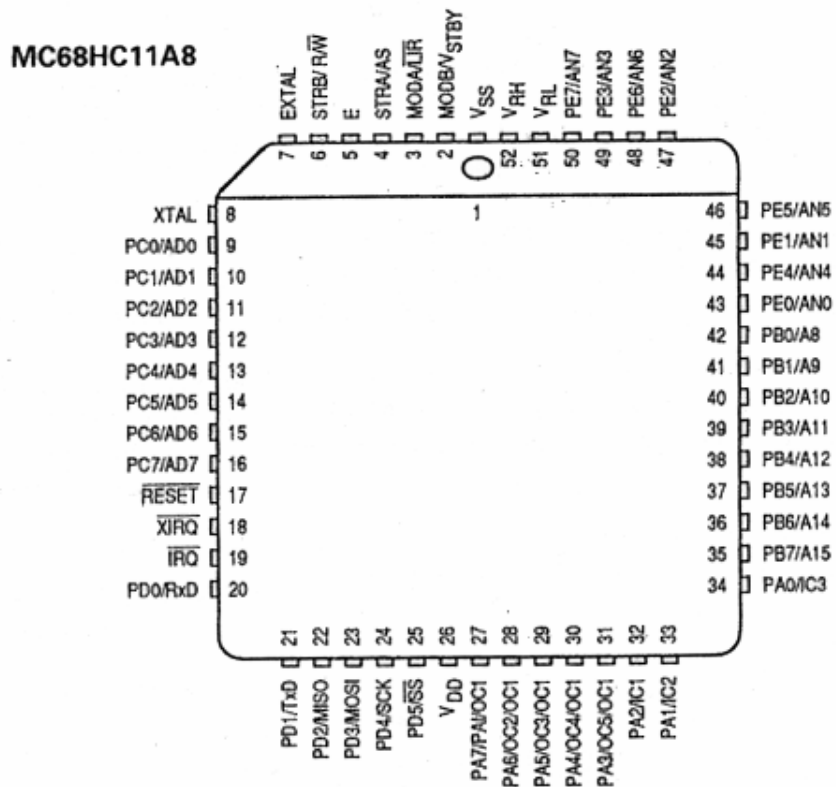
□ Ports du 68HC811E2



□ Brochage du MC68HC11F1 en boîtier PLCC 68 broches



□ Brochage du MC68HC11A8



3.2 Signaux du MC68HC11F1

- **VSS et VDD : tension d'alimentation**

La tension d'alimentation ($5V \pm 5\%$) est appliquée entre les entrées VDD et VSS. VDD doit être connectée au +5V et VSS est connectée à la masse.

- **XTAL et EXTAL : contrôle du quartz ou entrée d'horloge (cf. brochage en annexe)**

Pour cadencer le fonctionnement du 68HC11, une horloge est nécessaire. Les broches XTAL et EXTAL permettent la connexion d'un quartz ou d'un oscillateur externe.

→ La broche XTAL est à un niveau indéterminé lorsque l'on utilise une horloge externe de fréquence désirée sur la broche EXTAL.

→ Les broches XTAL & EXTAL sont utilisées pour piloter un quartz oscillant à la fréquence désirée.

- **E : sortie d'horloge**

Le signal présent sur la broche E est généré en interne par division par 4 de l'horloge du quartz. Ce signal est utilisé comme référence lors des transferts sur les bus. Lorsque le signal E est à l'état bas, le processeur effectue des tâches internes. Dans le cas inverse, les données sont accessibles sur le bus concerné.

Remarque : Toutes les horloges, incluant E, sont arrêtées lorsque le microprocesseur est en mode STOP.

- **4XOUT : horloge de base**

Cette sortie fournit un signal dont la fréquence est égale à 4 fois E et qui peut être utilisé pour piloter l'horloge d'un deuxième microprocesseur par sa broche EXTAL.

La sortie 4XOUT est validée après un reset mais elle peut être inhibée par le programme, en mettant à zéro le bit CLK4X dans le registre OPT2.

- **RESET : signal d'initialisation**

Cette broche est bidirectionnelle. Un de ses rôles est d'effectuer l'initialisation du MCU dans un état logique définie lors de l'application d'un niveau bas pendant plus d'un cycle d'horloge. Il y a également initialisation du compteur programme PC avec le contenu du vecteur de reset (\$FFFE et \$FFFF) qui contient l'adresse de début du programme.

- **$\overline{\text{IRQ}}$ & $\overline{\text{XIRQ}}$: interruptions**

Le MCU dispose de 2 entrées d'interruptions externes :

- L'entrée $\overline{\text{IRQ}}$ donne un accès pour établir une demande d'interruption de programme asynchrone. Le déclenchement de l'interruption peut être validé soit sur un front descendant, soit sur un niveau bas, par le bit 5 (IRQE) dans le registre OPTION (\$1039). Cette entrée correspond à un niveau faible de priorité. Ce niveau est masquable par programmation logicielle.
- L'entrée $\overline{\text{XIRQ}}$, active sur un niveau bas, correspond à un niveau de forte priorité. Ce niveau est non masquable de manière logicielle. Cette entrée permet de déclencher une demande d'interruption non masquable après un reset.

Remarque : Pour déterminer le sous-programme d'interruption à exécuter, le MCU utilise un vecteur d'interruption qui contient l'adresse du début du sous programme. $\overline{\text{IRQ}}$: FFF2-FFF3 et $\overline{\text{XIRQ}}$: FFF4-FFF5

- **$\text{MODA}/\overline{\text{LIR}}$ et MODB/VSTBY : Mode de fonctionnement**

Ces deux broches définissent le mode de fonctionnement du 68HC11 lors d'un reset.

MODB	MODA	Mode de fonctionnement	
1	0	Mode normal	Microcontrôleur ou monopuce
	1		Microprocesseur ou étendu
0	0	Mode spécial	Bootstrap
	1		Test

Après le reset, la broche $\text{MODA}/\overline{\text{LIR}}$ active la fonction $\overline{\text{LIR}}$ qui permet de synchroniser un circuit externe sur le début d'exécution d'une instruction. Dans ce mode, le MCU génère une série de cycles de l'horloge E durant l'exécution d'une instruction. La broche MODB/VSTBY permet de connecter une alimentation indépendante (généralement une pile ou une batterie) pour maintenir le contenu de la mémoire RAM interne au MC68HC11F1 lorsque l'alimentation +Vdd est coupée. (cf. annexe)

- **$\text{R}/\overline{\text{W}}$: sens de transfert des données**

Lorsque que le HC11 est activé en mode étendu ou test, le signal $\text{R}/\overline{\text{W}}$ indique le sens du transfert des données sur le bus de données externe. Un zéro logique indique une écriture sur le bus, le niveau logique 1 étant actif lors d'une opération de lecture. Le signal $\text{R}/\overline{\text{W}}$ reste au niveau zéro lors de l'exécution d'une instruction portant sur deux octets consécutifs.

- **VRH et VRL : tension de référence pour la conversion A→N**

Ces deux broches fournissent la tension de référence pour le convertisseur analogique numérique interne au HC11. VRH doit être inférieur à VDD.

- **RxD / TxD : Utilisation de l'interface série asynchrone SCI**

Lorsque l'on utilise l'interface série asynchrone SCI, la broche RxD réceptionne les données et la broche TxD permet d'émettre les données.

- **MISO / MOSI / \overline{SS} / SCK : Utilisation de l'interface série synchrone SPI**

Lorsque l'on utilise l'interface série synchrone SPI :

- MISO est une entrée en mode maître et une sortie en mode esclave,
- MOSI à l'inverse, est une sortie en mode maître et une entrée en mode esclave.
- \overline{SS} (Slave Select) permet de sélectionner les différents esclaves. Lorsque ce signal est à l'état haut, l'esclave est inactif. Il ignore les signaux d'horloge et il maintient la broche de sortie MISO en haute impédance. Lorsque ce signal est à l'état bas, l'esclave est actif et accroche les données de MOSI à la cadence de l'horloge SCK tout en renvoyant le contenu de son registre SPDR sur la ligne MISO (échange de registre SPDR entre le maître et l'esclave).
- SCK est le signal d'horloge fournie par le maître. Cette broche devient une entrée sur les périphériques SPI esclaves.

3.3 Fonctions des ports du MPU

Le MC68HC11F1 dispose de 54 broches (E, S ou E/S) organisées en 7 ports parallèles. Les ports du MCU remplissent plusieurs fonctions selon le mode de fonctionnement opté pour le MPU. Les ports utilisant des broches bidirectionnelles ont un registre associé pour la définition de la direction des données circulant sur les broches. Ces registres (DDR_x) ont un bit pour chacune des lignes concernées.

Les fonctions E/S des ports B, C et F ne sont disponibles qu'en mode monopuce et bootstrap. Les broches des ports A, C, D et G sont bidirectionnelles, alors que les broches des ports B et F sont uniquement des sorties. Le port E, qui gère les entrées du convertisseur analogique numérique, fonctionne uniquement en entrée.

Port	Broches	Bit	Mode monopuce et bootstrap	Mode étendu et test
A	8: E/S	0	PA0/IC3	
		1	PA1/IC2	
		2	PA2/IC1	
		3	PA3/OC5/IC4/OC1	
		4	PA4/OC4/OC1	
		5	PA5/OC3/OC1	
		6	PA6/OC2/OC1	
		7	PA7/PAI/OC1	
B	8: S	[7:0]	PB[7:0]	ADDR[15:8]
C	8: E/S	[7:0]	PC[7:0]	DATA[7:0]
D	6: E/S	0	PD0/RxD	
		1	PD1/TxD	
		2	PD2/MISO	
		3	PD3/MOSI	
		4	PD4/SCK	
		5	PD5/SS	
E	8: E	[7:0]	PE[7:0]/AN[7:0]	
F	8: S	[7:0]	PF[7:0]	ADDR[7:0]
G	8: E/S	0	PG0	
		1	PG1	
		2	PG2	
		3	PG3	
		4	PG4	PG4/CSIO2
		5	PG5	PG5/CSIO2
		6	PG6	PG6/CSGEN
		7	PG7	PG7/CSPROG

Port	E	S	E/S	Fonctions partagées
A			8	Compteur
B		8		Octet haut du bus d'adresse
C			8	Bus de données
D			6	Interface série (SCI & SPI)
E	8			Entrée du convertisseur A/D
F		8		Octet bas du bus d'adresse
G			8	Sorties des Chip Select

- **Port A (\$1000) : 3 entrées (PA0 à PA2), 3 sorties (PA4 à PA6) et 2 E/S (PA3 et PA7)**

Le port A est un port d'usage général qui permet des E/S sur 8 bits. Il dispose d'un registre PACTL (\$1026) qui permet de définir le sens de transfert des 2 broches E/S. Les fonctions des broches du port A sont partagées avec le compteur Timer interne du HC11. L'état du port A peut être lu à tout moment. Lorsque le port est utilisé en sortie, l'information est maintenue par des verrous internes (*latches*) au HC11.

- **Port B (\$1004) : 8 sorties (PB0 à PB7)**

Le port B (8 bits) est utilisable uniquement en sortie. En mode monopuce, les broches du port B sont validées en sortie. En mode étendu, les broches du port B sont affectées aux adresses hautes ADDR[15:8] du bus d'adresses du MCU. L'état du port B peut être lu à tout moment. Lorsque le port est utilisé en sortie, l'information est maintenue par des verrous internes au HC11.

- **Port C (\$1003) : 8 E/S (PC0 à PC7)**

Le port C (8 bits) est un port d'usage général. Il dispose d'un registre DDRC (\$1007) permettant la programmation du sens de transfert des données sur chacune de ses broches. En mode monopuce, les 8 bits sont affectés aux E/S. En mode étendue, le port C est configuré comme bus de données DATA[7:0]. L'état du port C peut être lu à tout moment.

- **Port D (\$1008) : 6 E/S (PD0 à PD5)**

Le port D est codé sur 6 bits à usage général. Il dispose d'un registre DDRD (\$1009) permettant la programmation du sens de transfert des données sur chacune de ses broches. Les six bits peuvent être affectés à des fonctions de communication via les protocoles SCI ou SPI. Le port D peut être lu à tout moment. Lorsque le port est utilisé en sortie, l'information maintenue par des verrous internes.

- **Port E (\$100A) : 8 entrées (PE0 à PE7)**

Le port E (8 bits) est utilisable en entrée uniquement dont les broches sont partagées avec le multiplexeur du convertisseur ADC interne au HC11. Les broches non utilisées par la fonction conversion AD (selon l'initialisation établie par l'utilisateur) peuvent être utilisées en entrées digitales. Cependant, le port E ne doit pas être lu lorsqu'une séquence de conversion analogique digitale est en cours.

- **Port F (\$1005) : 8 sorties (PF0 à PF7)**

Le port F (8 bits) est utilisable uniquement en sortie. En mode monopuce, les broches sont des entrées à usage général. En mode étendue, le port F est utilisé pour générer les adresses basses ADDR[7:0] du HC11. Le port F peut être lu à tout moment. Lorsque le port est utilisé en sortie, l'information est maintenue par des verrous internes. Après un reset en mode étendu, le port F est affecté au bus adresse ADDR[7:0].

- **Port G (\$1002) : 8 E/S (PG0 à PG7)**

Le port G est un port qui permet les E/S sur 8 bits. Il dispose d'un registre DDRG (\$1003) permettant la programmation du sens de transfert des données sur chacune de ses broches. Le port G peut être lu à tout moment. Lorsque le port est utilisé en sortie, l'information est maintenue par des verrous internes. En mode étendu, les quatre bits [7:4] du port G peuvent être configurés en validation d'adresse (*Chip Select*).

4. LES ACCUMULATEURS INTERNES

Les registres internes du CPU⁹ font partie du MCU et ne peuvent être adressés comme une location mémoire dans le plan du microprocesseur.

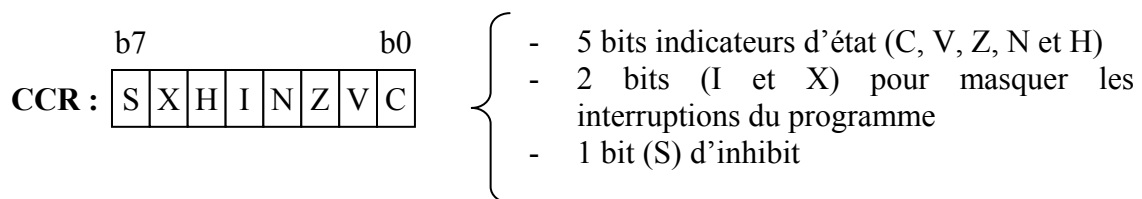
Les registres internes de la famille des MC68HC11 se composent de :

- 3 registres de 8 bits :
 - 2 accumulateurs A et B
 - 1 registre d'état CCR (Condition Code Register)

- 5 registres de 16 bits :
 - 2 registres d'index (IX et IY)
 - 1 registre de pile (SP)
 - 1 registre de compteur de programme (PC)
 - 1 pseudo accumulateur D (concaténation de A et B)

□ Les 7 registres du CPU utilisables par le programmeur

7	A	0	7	B	0	Registres 8 bits, usage général
15	D				0	Registre 16 bits, concaténation de A et B
15	IX				0	Registre 16 bits d'index X, usage général
15	IY				0	Registre 16 bits d'index Y, usage général
15	SP				0	Pointeur de pile 16 bits. Indexe l'adresse d'accès à la pile
15	PC				0	Compteur programme
			7	CCR	0	Registre d'état CCR (Condition Code Register) 8 bits : donne le résultat des opérations logiques ou arithmétiques.



⁹ CPU : Central Processor Unit : Unité centrale effectuant les opérations logiques et arithmétiques nécessaires au calculateur.

- **Accumulateurs A, B et D**

Les accumulateurs A et B sont des registres codés sur 8 bits, qui contiennent les opérandes ainsi que les résultats d'opérations arithmétiques ou de manipulations de données. Dans le cas de certaines instructions, ces deux registres sont traités comme un seul accumulateur sur 16 bits : le registre D.

- **Registres d'index X (IX) et d'index Y (IY)**

- Le registre d'index X fournit une valeur d'indexage codée sur 16 bits qui peut être ajoutée à un décalage sur 8 bits, selon les instructions spécifiques, pour définir l'adresse effective d'un périphérique. Le registre X peut être utilisé comme compteur où comme zone de stockage temporaire dans le cours d'un programme.
- Le registre Y effectue les mêmes fonctions d'indexage que celles définies pour le registre X. Cependant, les instructions faisant appel au registre Y nécessitent un octet supplémentaire pour le codage de l'instruction, ainsi qu'un cycle supplémentaire pour l'exécution.

- **Registre de pile (SP)**

Le 68HC11 dispose d'une gestion automatique de la pile (*stack*) qui peut être situé dans n'importe quelle zone de l'espace mémoire adressable par le microprocesseur. La taille de la pile peut prendre toute dimension souhaitée dans la limite de la mémoire RAM disponible. L'adresse de base de la pile doit être définie dès le début du programme. La pile est configurée comme une structure sur 8 bits qui croit vers le bas à chaque écriture.

Lorsqu'un sous-programme est appelé par une instruction, l'adresse de l'instruction est sauvegardée sur la pile, l'octet le moins significatif en premier. L'adresse de retour est extraite de la pile et chargée dans le compteur de programme PC. Le programme reprend l'exécution à cette adresse.

Lorsqu'une demande d'interruption est identifiée, l'instruction en cours se termine normalement, l'adresse de retour (valeur courante du PC) est sauvegardée sur la pile, l'octet le moins significatif en premier suivi de tous les registres du CPU. L'exécution du programme se poursuit à partir de l'adresse définie dans le vecteur de l'interruption concernée. À la fin du traitement de la routine d'interruption, le programme reprend son cours à l'adresse extraite de la pile.

Il est important de s'assurer que la mémoire RAM disponible est suffisante pour couvrir tous les accès à la pile, en particulier lorsque le programme utilise des sous-routines imbriquées.

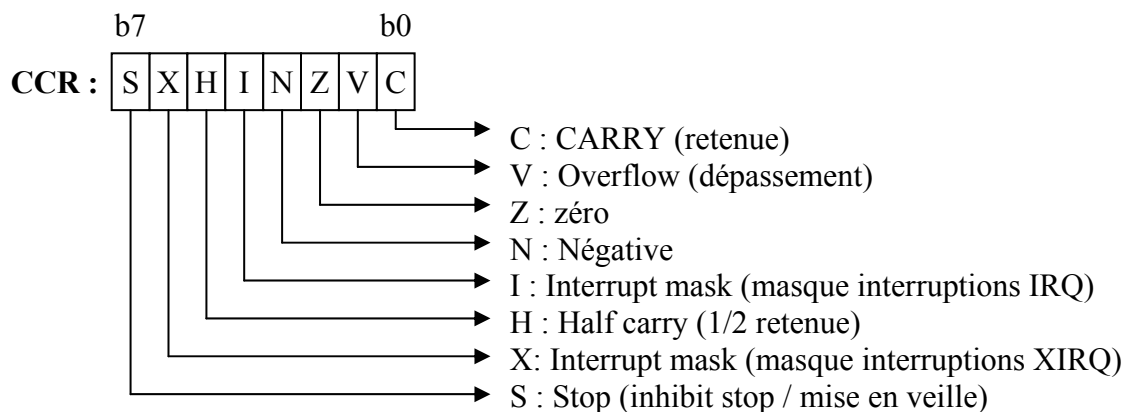
Le MC68HC11 dispose d'instructions permettant de sauvegarder et d'extraire des octets sur la pile.

- **Le compteur de programme (PC)**

Le compteur de programme est un registre codé sur 16 bits qui contient l'adresse de la prochaine instruction devant être exécutée. Après un reset, le PC est initialisé à partir de l'un des six vecteurs possibles, selon le mode de fonctionnement du MCU.

- **Le registre d'état (CCR)**

Le registre d'état contient 5 bits indicateurs d'état (C, V, Z, N et H), 2 bits (I et X) pour masquer les interruptions du programme et 1 bit (S) d'inhibit



5. MODELE DE PROGRAMMATION

La famille des microcontrôleurs HC11 utilise des instructions codées sur 8 bits sur une des 4 pages de codes opératoires possibles (max d'instructions possibles = 256×4). Chaque instruction identifie une fonction particulière ainsi que le mode d'adressage utilisé. Une instruction complète comprend un prébyte, un code opératoire et jusqu'à 3 opérandes. De ce fait, une instruction occupe de 1 à 5 octets dans la mémoire du système. Il existe 6 modes d'adressage des MCU de la famille des MC68HC11xx : immédiat, direct, étendu, indexé, inhérent et relatif (cf. annexe)

On dispose de langages de programmation de haut niveau tel le Pascal, le C ou C++. Néanmoins, l'électronique d'un système informatique ne sait interpréter que le seul code binaire. Il est donc obligatoire de convertir le programme écrit dans un langage « humain » en un code utilisable par l'ordinateur : c'est le rôle des assembleurs.

Dans le cadre des MCU, il est souvent plus efficace, du point de vue dimension du code binaire généré et de la vitesse d'exécution, d'écrire le programme dans la symbolique définie pour un type de MCU donné. Cette méthode est difficile d'emploi si le programme doit gérer du texte : le Pascal ou le C++ s'imposent comme solution.

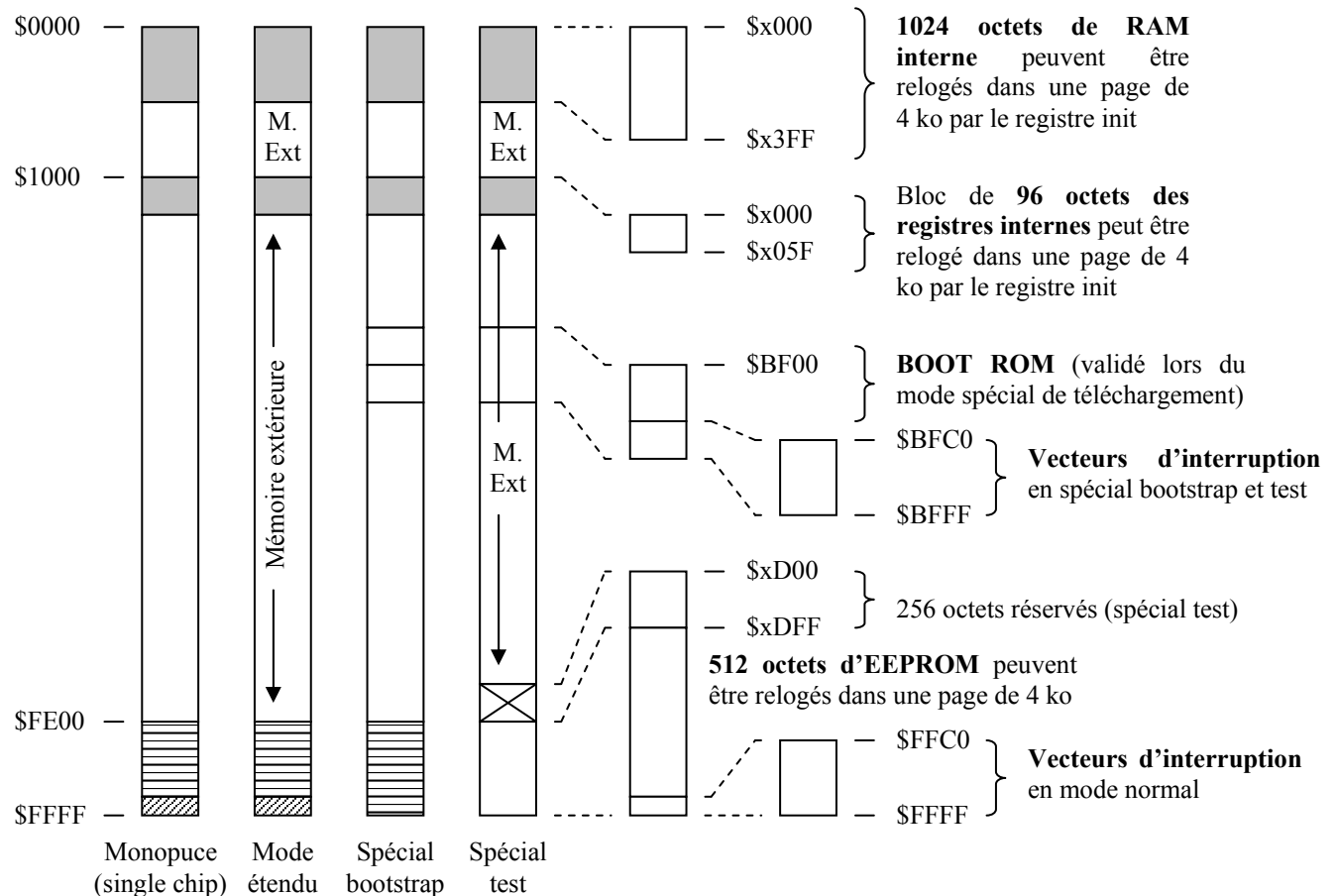
Par ailleurs, il existe des compilateurs croisés (*cross compiler*) qui permettent de générer le code binaire utilisable par le MCU à partir d'un langage de haut niveau comme le C. (cf. TP).

6. PLAN MEMOIRE DU MCU

Les locations des ressources intégrées dans la puce du HC11F1 sont identiques dans les modes de fonctionnement monopuce ou étendu.

Le bloc de 96 octets affectés aux registres commence à l'adresse \$1000 par défaut après un reset. Ce bloc peut être déplacé au début de n'importe quelle page de 4 ko (\$x000) en écrivant la valeur correcte dans le registre INIT après un reset.

□ Plan mémoire du MC68HC11F1 :



- Le MC68HC11F1 comporte 1024 octets de RAM statique qui peut être utilisée pour un programme exécutable, pour le stockage de variables et de données temporaires lors du déroulement du programme.
- La RAM peut être relogée dans n'importe quelle page de 4 ko dans l'espace des 64 ko adressable par le MCU en écrivant la valeur appropriée dans le registre INIT. La RAM est située, par défaut, à l'adresse \$0000 après un reset, permettant ainsi l'accès aux octets compris entre \$0000 et \$00FF en mode direct.
- Les 512 octets de l'EEPROM sont localisés à l'adresse \$FE00 à \$FFFF après un reset dès lors que l'EEPROM est validée dans le plan mémoire par le registre CONFIG.
- La ROM, contenant le programme de téléchargement (Bootloader ROM) est validée dans la zone \$BF00 à \$BFFF lors du fonctionnement du MCU en mode *bootstrap*.
- Le MC68HC11F1 comprend 96 octets (registres de 8 bits) dédiés au contrôle de la machine. (cf. doc 68HC11)

7. GESTION DES E/S PARALLELES

Le MC68HC11F1 disposent de 54 lignes d'entrées / sorties, organisées en 7 ports parallèles (A, B, C, D, E, F et G) selon le mode de fonctionnement choisi. La fonction du port est dépendante du mode de fonctionnement choisi pour le MCU.

Les ports utilisant des broches bidirectionnelles (A, C, D et G) ont un registre associé (DDRA, DDRC, DDRD et DDRG) pour la définition de la direction des données circulant sur les broches. Ces registres (DDRx) ont un bit pour chacune des lignes du port concerné.

Configuration des ports séries :

Port	Broches d'entrées	Broches de sorties	Broches bidirectionnelles	Fonctions partagées
Port A	-	-	8	Compteur
Port B	-	8	-	Octet haut du bus d'adresse
Port C	-	-	8	Bus de données
Port D	-	-	6	Interface SPI et SCI
Port E	8	-	-	Entrées du convertisseur A/D
Port F	-	8	-	Octet bas du bus d'adresse
Port G	-	-	8	Sortie des chip select

Remarque :

- Les tableaux suivants donnent la description des ports parallèles. Après un reset, lorsque les états logiques sont connus, ils sont symbolisés par 0 ou 1 sinon on utilise X pour l'état indéterminé.
- DDRX[7:0] : direction des données pour le port X (0 = entrée , 1 : sortie)

□ Port A , données du port A, \$1000 :

Bits :	7	6	5	4	3	2	1	0
Nom :	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
Reset :	X	X	X	X	X	X	X	X
Fonctions alternées :	PAI	OC2	OC3	OC4	IC4/OC5	IC1	IC2	IC3
Autres :	OC1	OC1	OC1	OC1	OC1			

□ PACTL , Pulse Accumulator Control Register, \$1026 :

Bits :	7	6	5	4	3	2	1	0
Nom :	DDRA7	PAEN	PAMOD	PEDGE	DDRA3	I4/O5	RTR1	RTR0
Reset :	0	0	0	0	0	0	0	0

□ **Port B , données du port B, \$1004 :**

Bits :	7	6	5	4	3	2	1	0
Nom :	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
Reset :	0	0	0	0	0	0	0	0
Monopuce ou Boot :	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
Etendue ou test :	ADDR15	ADDR14	ADDR13	ADDR12	ADDR11	ADDR10	ADDR9	ADDR8

□ **Port C , données du port C, \$1003 :**

Bits :	7	6	5	4	3	2	1	0
Nom :	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Reset :	X	X	X	X	X	X	X	X
Monopuce ou Boot :	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Etendue ou test :	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0

□ **DDRC , registre de direction de données pour le port C, \$1007 :**

Bits :	7	6	5	4	3	2	1	0
Nom :	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0
Reset :	0	0	0	0	0	0	0	0

□ **Port D , données du port D, \$1008 :**

Bits :	7	6	5	4	3	2	1	0
Nom :	-	-	PD5	PD4	PD3	PD2	PD1	PD0
Reset :	0	0	X	X	X	X	X	X
Fonctions alternées :	-	-	\overline{SS}	SCK	MOSI	MISO	TxD	RxD

□ **DDRD , registre de direction de données pour le port D, \$1009 :**

Bits :	7	6	5	4	3	2	1	0
Nom :	-	-	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0
Reset :	0	0	0	0	0	0	0	0

□ **Port E , données du port E, \$100A :**

Bits :	7	6	5	4	3	2	1	0
Nom :	PE7	-	PE5	PE4	PE3	PE2	PE1	PE0
Reset :	X	0	X	X	X	X	X	X
Fonctions alternées :	AN7	-	AN5	AN4	AN3	AN2	AN1	AN0

□ **Port F , données du port F, \$1005 :**

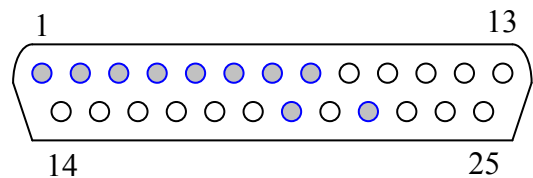
Bits :	7	6	5	4	3	2	1	0
Nom :	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
Reset :	0	0	0	0	0	0	0	0
Monopuce ou Boot :	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
Etendue ou test :	ADDR7	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0

8. LA LIAISON SERIE ASYNCHRONE (SCI)

L'interface de communication série est constituée d'un récepteur transmetteur asynchrone universel (UART) qui est l'un des deux sous systèmes E/S série interne au MC68HC11F1. Il est basé sur le format standard avec un bit de start, 8 ou 9 bits de données et un bit de stop. Plusieurs vitesses de communication (baud) sont disponibles et programmables par l'utilisateur. Le récepteur et le transmetteur du système SCI sont indépendants mais utilisent le même format ainsi que la même vitesse de communication.

Le signal est transmis par deux voies de communications (Txd et Rxd) associées à un signal commun (SG : Signal ground) et quatre signaux de contrôle (qui ne sont pas tous utilisés systématiquement).

- **Brochage normalisé du connecteur DB25 pour une liaison série :**



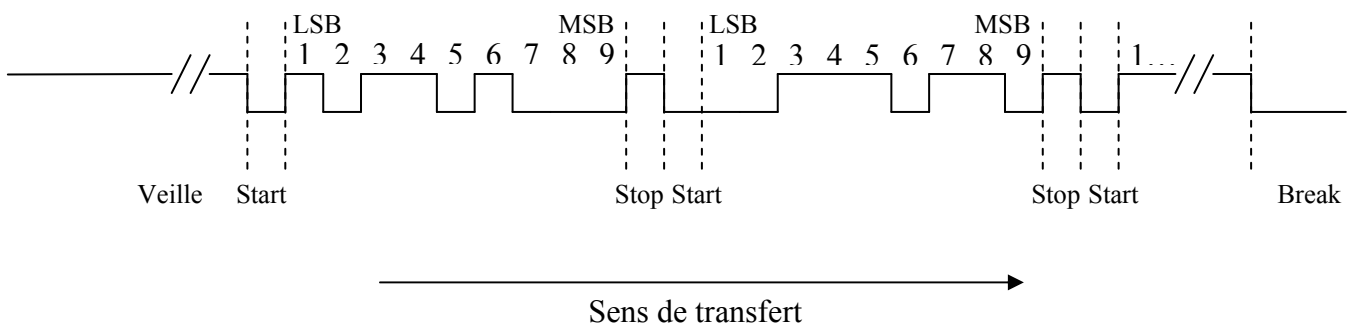
Broche DB25	Nom	Abréviation	Fonction
1	Protective Ground	FG	Effectue la connexion entre le châssis des équipements concernés
2	Transmit Data	TxD	Donnée transmise dans le sens calculateur → terminal (périphérique)
3	Receive Data	RxD	Donnée transmise dans le sens terminal (périphérique) → calculateur
4	Request To Send (Demande pour émettre)	RTS	Full duplex = valide les circuits de transmission Half duplex = établit la liaison en mode transmission et inhibe les circuits de réception
5	Clear To Send (prêt à émettre)	CTS	Réponse au signal RTS : le niveau 1 indique le fonctionnement du circuit de transmission
6	Data Set Ready	DSR	Liaison téléphonique : la ligne est en mode veille et peut donc être utilisée
7	Signal Ground	SG	Commun
8	Data Carrier Detect	DCD	Détection d'un signal valide sur la ligne
20	Data Terminal Ready	DTR	Le terminal est prêt pour la réception de données
22	Ring Indicator	RI	Indicateur de sonnerie

Le MC68HC11F1 ne dispose pas d'entrées dédiées aux lignes de contrôle. Le port série se limite aux lignes RxD et TxD suffisantes pour des liaisons locales entre le MCU et un périphérique proche. Dans le cas contraire, il faut prendre les précautions nécessaires pour s'assurer que les données transmises ne seraient pas perturbées lors de la communication. On utilisera des circuits spécifiques (cf. doc 68HC11).

□ Format des données

Le format des données sur la liaison série est défini comme suit :

- Une mise à 1 (état veille, IDLE) de la ligne avant émission ou réception d'un caractère.
- Un bit de start (0 logique), transmis ou reçu, qui indique le début de chaque caractère.
- L'octet est transmis ou reçu avec le bit le moins significatif (LSB) en premier. Le choix de la longueur du caractère est effectué par le bit M dans le registre SCCR1.
- Un bit de stop (1 logique) indique la fin d'une trame (une trame consiste en un bit de start, un caractère codé sur 8 ou 9 bits, et un bit de stop).
- Un break qui est défini comme la transmission ou la réception d'un zéro logique pour plusieurs trames consécutives.



□ Fonctionnement en Emission

Le transmetteur du SCI inclut un registre de donnée parallèle (SCDR) et un registre à décalage pour la transmission série. Le contenu du registre à décalage ne peut être écrit que par la biais du registre SCDR.

Cette opération permet le transfert d'un caractère sur le port série pendant le caractère suivant est en attente dans le registre SCDR. La sortie du registre à décalage est appliquée à la broche TxD tant que la transmission est en cours, ou que le bit TE du registre de contrôle SCCR2 est mis à 1.

□ Fonctionnement en réception

Lors des opérations de réception, la séquence de transmission est inversée. Le registre à décalage du port série reçoit les données et les transfère dans le registre parallèle dédié au port série (SCDR) sous la forme d'un mot complet. Cette opération permet de lire un caractère sur le registre à décalage tandis que le caractère précédent est disponible dans le registre SCDR.

□ **Faculté de réveil : Evaluation du 1er caractère de chaque message**

La faculté de réveil du SCI permet de réduire la charge logiciel du traitement du SCI dans les ensembles utilisant plusieurs récepteurs grâce à l'évaluation du 1^{er} caractère de chaque message.

Le récepteur est placé dans le mode veille en mettant à 1 le bit RWU dans le registre SCCR2. Lorsque le bit RWU est à 1, tous les indicateurs relatifs au récepteur (RDRF, IDLE, OR, NF et FF) sont inhibés. Le bit RWU est mis à un par le programme et automatiquement mis à zéro par le matériel. Dès qu'un nouveau message commence, la logique alerte les récepteurs en veille de passer en mode actif pour évaluer le premier caractère de ce nouveau message.

Il existe deux méthodes pour activer cette fonction :

➤ Réveil par la ligne d'attente (Idle line) :

Lors d'un réveil par la ligne d'attente, un récepteur en veille est activé dès que la ligne RxD passe en mode attente (début d'un message). Tous les récepteurs sont activés (RWU = 0) dès que chaque message commence. Dès qu'un récepteur a déterminé qu'un message ne lui est pas adressé, il remet le bit RWU à 1, ce qui inhibe la mise à un des indicateurs SCI jusqu'au retour au niveau IDLE de la ligne RxD en fin du message en cours. Ainsi les caractères suivants du message ne sont pas pris en compte par le récepteur.

➤ Réveil par adresse (address mark) :

Les caractères transmis dans ce mode de réveil du système SCI sont codés sur 7 ou 8 bits (si le bit M = 1 dans le registre SCCR1) et d'un niveau 1 dans le bit le plus significatif MSB qui indique que le caractère est une adresse. Le premier caractère d'un message est toujours un caractère d'adresse (MSB = 1). Tous les récepteurs du système évaluent ce caractère afin de déterminer si la suite du message leur est destinée. Dès qu'un récepteur établit qu'un message ne lui est pas destiné, il active la fonction RWU en mettant à 1 ce bit dans le registre SCCR2. Étant donné que les indicateurs de réception sont inhibés lorsque RWU = 1, le MCU est libéré du traitement du port série pour le reste du message.

□ **Détection des erreurs**

Trois conditions d'erreurs (dépassement SCRD, bruit sur un bit et erreur de trame) peuvent être activées lors de la génération d'une interruption par le système SCI. Le registre d'état du système (SCSR) dispose de trois bits (OR, NF et FE) pour identifier ce type d'erreurs dans la transmission. (cf. doc 68HC11)

□ **Registres du port série**

La liaison série asynchrone (SCI) dispose de 5 registres adressables :

- ❖ 3 registres de contrôle :
 - SCCR1 : registre de contrôle n° 1 du SCI
 - SCCR2 : registre de contrôle n° 2 du SCI
 - BAUD : registre de définition Baud du SCI
- ❖ 1 registre contenant les données du SCI:
 - SCDR : registre de données du SCI : contient la donnée transmise ou émise
- ❖ 1 registre d'état du SCI:
 - SCSR : registre de statut du SCI : condition pour la génération des demandes d'interruptions du système SCI.

➤ **SCDR** , registre des données du SCI, \$102F :

Bits :	7	6	5	4	3	2	1	0
Nom :	R7/T7	R6/T6	R5/T5	R4/T4	R3/T3	R2/T2	R1/T1	R0/T0
Reset :	X	X	X	X	X	X	X	X

R : réception T : transmission

➤ **SCCR1**, registre de contrôle n°1 du SCI, \$102C :

Bits :	7	6	5	4	3	2	1	0
Nom :	R8	T8	-	M	WAKE	-	-	-
Reset :	X	X	0	0	0	0	0	0

Les bits du registre SCCR1 permettent de définir la longueur du mot transmis sur le port série ainsi que la méthode devant être utilisée pour effectuer l'activation du port.

R8	bit 8 dans la réception des données. Dans le cas où le bit M serait à 1, R8 contient le 9 ^{ème} bit dans le caractère reçu
T8	bit 8 dans la transmission des données. Dans le cas où le bit M serait à 1, T8 contient le 9 ^{ème} bit dans le caractère transmis
M	Sélection du format de transmission : 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
WAKE	Sélection du mode d'activation (réveil) du port série 0 = activation par reconnaissance de l'état IDLE (attente) 1 = activation par reconnaissance d'adresse (LSB est à 1)

➤ **SCCR2**, registre de contrôle n° 2 du SCI, \$102D :

Bits :	7	6	5	4	3	2	1	0
Nom :	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
Reset :	0	0	0	0	0	0	0	0

Les bits du registre SCCR2 permettent de définir les bits de contrôle pour valider ou inhiber les fonctions SCI individuelles. (cf. registre de statut SCSR)

TIE	Validation de la demande d'interruption pour transmission 0 = Interruption TDRE inhibées 1 = Demande d'interruption SCI validée lorsque l'indicateur TDRE de SCSR est à 1 (registre d'émission vide)
TCIE	Validation de la demande d'interruption à la fin d'une transmission 0 = Interruption TC inhibée 1 = Interruption TC validée
RIE	Validation de demande d'interruption par la réception 0 = Interruption RDRF et OR inhibées 1 = Demande d'interruption SCI validée lorsque l'indicateur RDRF ou OR est à 1 (registre de réception est plein)
ILIE	Validation de la demande d'interruption IDLE 0 = Interruption IDLE inhibée 1 = Demande d'interruption SCI validée lorsque l'indicateur IDLE est à 1 (la ligne est libre)
TE	Validation du transmetteur 0 = transmetteur inhibé 1 = transmetteur validé (la transmission est autorisée)
RE	Validation du récepteur 0 = récepteur inhibé 1 = récepteur validé (la réception est autorisée)
RWU	Contrôle du mode veille 0 = fonctionnement normal du récepteur 1 = mode veille validé et demandes d'interruptions du récepteur inhibées
SBK	Transmission d'un break 0 = générateur de break arrêté 1 = générateur de break activé

Remarque :

Le vecteur d'interruption du SCI se trouve en \$FFD6 - \$FFD7

➤ **SCSR**, registre de statut SCI, \$102E :

Bits :	7	6	5	4	3	2	1	0
Nom :	TDRE	TC	RDRF	IDLE	OR	NF	FE	-
Reset :	1	1	0	0	0	0	0	0

Ce registre fournit les conditions pour la génération des demandes d'interruptions du système SCI

TDRE (1)	Indicateur de registre de transmission vide. Cet indicateur est mis à 1 lorsque le registre de données SCDR est vide. 0 = registre de donnée SCDR occupé 1 = registre de donnée SCDR vide
TC (1)	Indicateur de fin de transmission (le registre à décalage est vide). 0 = transmetteur occupé (transmission en cours) 1 = transmetteur en attente (transmission terminée)
RDRF (1)	Indicateur d'état du registre de réception de donnée. Cet indicateur est mis à 1 lorsqu'un caractère, reçu par le port série, est disponible en lecture dans le registre de données SCDR. 0 = registre de donnée SCDR vide 1 = registre de donnée SCDR plein
IDLE (1)	Indicateur de détection d'un état de veille. Cet indicateur est mis à 1 lorsque la ligne RxD est en attente. 0 = la ligne RxD est active 1 = la ligne RxD est en attente
OR (1)	Indicateur de dépassement (Over Run Flag) Est mis à 1 si un nouveau caractère est reçu avant que le caractère précédent ne soit lu dans le registre SCDR. 0 = pas de dépassement 1 = dépassement détecté
NF (1)	Indicateur de bruit (Noise Error Flag). Est mis à 1 si la logique interne du HC11 détecte un bit perturbé par du bruit lors de la communication. (Cf. doc 68HC11)
FE (1)	Erreur de trame (Framing Error) Est mis à 1 lorsque le niveau logique détecté est à 0 au lieu du bit de stop attendu.
Bit 0 (1)	Non implémenté

(1) : La mise à zéro de ce bit s'effectue en lisant le registre de statut SCSR suivi d'une écriture dans le registre de données SCDR.

Remarque : Le transmetteur SCI dispose de deux indicateurs d'état qui peuvent être lus par logiciel pour déterminer les conditions logiques. Un bit de validation d'interruption peut être utilisé pour permettre la génération d'une interruption lorsque la condition logique correspondante est remplie pour chacune des conditions d'état du port série (Cf. doc 68HC11)

➤ **BAUD**, registre de définition Baud (vitesse de transfert) du SCI, \$102B :

Bits :	7	6	5	4	3	2	1	0
Nom :	TCLR	-	SCP1	SCP0	RCKB	SCR2	SCR1	SCR0
Reset :	0	0	0	0	0	Non affecté		

Ce registre contrôle la vitesse de transfert du SCI :

TCLR	Mise à zéro du compteur de baud (test)
SCP1 SCP0	Sélection du pré-diviseur du générateur de Baud du SCI (Cf tableau suivant)
RCKB	Vérification du générateur d'horloge de baud (test)
SCR2 SCR1 SCR0	Sélection de la vitesse en baud du SCI (cf. tableau suivant)

❖ Pré-diviseur du générateur de baud : Les bits SCP1 et SCP0 déterminent la plus grande vitesse possible pour un quartz donné :

Pré-diviseur		Diviseur	Fréquence du quartz (Mhz) : 4 fois l'horloge E					
SCP1	SCP0	CLOCK	4,00	4,9152	8,00	12,00	16,00	20,00
0	0	1	62 500	76 800	12 5000	187 500	250 000	312 500
0	1	3	20 833	25 600	41 667	62 500	83 332	104 165
1	0	4	15 625	19 200	31 250	46 875	62 500	78 125
1	1	13	4 800	5 907	9 600	14 423	19 200	24 000

Taux de transfert maximal en baud

❖ Corrélation du quartz et du port RS323 par programmation du registre BAUD : les bits SCR2, SCR1 et SCR0 sélectionne un sous-multiple binaire additionnel (:1, :2, :4 jusqu'à :128) de la vitesse précédente. Le résultat de ces deux diviseurs mis en série constitue l'horloge 16X pour le récepteur :

SCR2	SCR1	SCR0	PRE DIV	4 800	19 200	76 800	9 600
0	0	0	1	4 800	19 200	76 800	9 600
0	0	1	2	2 400	9 600	38 400	4 800
0	1	0	4	1 200	4 800	19 200	2 400
0	1	1	8	600	2 400	9 600	1 200
1	0	0	16	300	1 200	4 800	600
1	0	1	32	150	600	2 400	300
1	1	0	64	75	300	1 200	150
1	1	1	128	-	150	600	75
				XTAL 4 MHZ	XTAL 16 MHZ	XTAL 4,91 MHZ	XTAL 8 MHZ

Taux de transfert en baud

9. LA LIAISON SERIE SYNCHRONE (SPI)

La liaison SPI permet une communication entre un MCU 68HC11 et :

- D'autres MCU 68HC11,
- Des périphériques séries synchrones (registre à décalage, convertisseur série, contrôleur d'afficheur 7 segments, contrôleur d'écran LCD, mémoires, horloges...)

La communication se fait par 2 signaux :

- le signal de données
- le signal d'horloge qui cadence le transfert

La liaison SPI est un échange de données entre le maître (MASTER) et l'esclave (SLAVE). Le maître est le seul apte à gérer une transmission car il génère le signal d'horloge. L'esclave subit les commandes du maître.

Le 68HC11 dispose de 4 signaux associés au port D pour gérer la liaison SPI.

□ Port D , données du port D, \$1008 :

Bits :	7	6	5	4	3	2	1	0
Nom :	-	-	PD5	PD4	PD3	PD2	PD1	PD0
Reset :	0	0	X	X	X	X	X	X
Fonctions alternées :	-	-	SS	SCK	MOSI	MISO	<i>TxD</i>	<i>RxD</i>

SS	Slave Select : permet de sélectionner les différents esclaves. Lorsque ce signal est à l'état haut, l'esclave est inactif. Il ignore les signaux d'horloge. Le liaison SPI du 68HC est désactivée Lorsque ce signal est à l'état bas, l'esclave est actif et accroche les données de MOSI à la cadence de l'horloge SCK tout en renvoyant le contenu de son registre SPDR sur la ligne MISO (échange de registre SPDR entre le maître et l'esclave).
SCK	Serial Clock : C'est le signal d'horloge fournie par le maître. Cette broche devient une entrée sur les périphériques SPI esclaves.
MOSI	Master Out Slave In : C'est une sortie en mode maître et une entrée en mode esclave
MISO	Master In Slave Out : C'est une entrée en mode maître et une sortie en mode esclave

□ DDRD , registre de direction des données pour le port D, \$1009 :

Bits :	7	6	5	4	3	2	1	0
Nom :	-	-	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0
Reset :	0	0	0	0	0	0	0	0

DDD5 à DDD0 : 0 = entrée 1 = sortie

9.1 Les trois registres de la SPI

□ **SPCR (Serial Peripheral Control Register) : registre de contrôle SPI, \$1028 :**

Bits :	7	6	5	4	3	2	1	0
Nom :	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0
Reset :	0	0	0	0	0	0	0	0

SPIE	Validation de l'interruption SPI en fin de transfert 1 = interruption validée 0 = interruption inhibée Si ce bit est à 1 et que le bit SPIF du registre SPSR passe à 1 (transmission complète en mode master ou octet reçu en mode slave), une interruption a lieu. Si ce bit est à 0, le passage du bit SPIF à 1 n'occasionne pas d'interruption. Il faut donc scruter le bit SPIF pour détecter un octet émis ou reçu
SPE	Validation SPI 1 = SPI validée 0 = SPI inhibée Si ce bit est à 0, la SPI n'est pas valide et le port D sert de port générique I/O. Si ce bit est à 1, les entrées dédiées au SPI (PD5 à PD0) sont occupées par le SPI
DWOM	PortD Wire Or Mode Option : ce bit permet de déterminer le type de sortie du port D 1 = les sorties du port D sont de type collecteur ouvert 0 = les sorties du port D sont de type CMOS (normal)
MSTR	Master Mode Select 1 = Le 68HC11 est en mode maître. La ligne \overline{SS} doit être à 1 0 = Le 68HC11 est en mode esclave. La ligne \overline{SS} doit être à 0
CPOL	Clock PO Larity : polarité du signal d'horloge au repos (hors transmission) 1 = le signal SCK est à 1 au repos 0 = le signal SCK est à 0 au repos
CPHA	Clock PHA se : front de validation du signal d'horloge 1 = le signal SCK valide la donnée sur front montant 0 = le signal SCK valide la donnée sur front descendant
SPR1 & SPR0	Configure la vitesse de transmission : fixe le diviseur de l'horloge interne (aucun effet en mode slave) 00 = horloge interne divisée par 2 01 = horloge interne divisée par 4 10 = horloge interne divisée par 16 11 = horloge interne divisée par 32

Remarque : le vecteur de l'interruption SPI est en \$FFF8-\$FFF9

□ **SPSR (Serial Peripheral Status Register) : registre de statut SPI, \$1029 :**

Bits :	7	6	5	4	3	2	1	0
Nom :	SPIF	WCOL	0	MODF	0	0	0	-
Reset :	0	0	0	0	0	0	0	0

SPIF	SPI transfer complete Flag : Drapeau de réception (SLAVE) ou transmission (MASTER) : 0 = aucun transfert terminé 1 = fin du transfert (et interruption si SPIE est positionné). La remise à 0 de ce bit est faite après une lecture de SPSR puis un accès à SPDR
WCOL	Write COLLision : Drapeau qui indique une collision en écriture 1 = collision détectée : on tente d'écrire dans SPDR alors que le transfert n'est pas terminé. 0 = aucune collision détectée
MODF	MODE Fault : Drapeau qui indique un défaut de mode. Le 68HC11 est configuré en maître alors que \overline{SS} est à 0. Le MCU se retrouve forcé en mode SLAVE (MSTR à 0) et le mode SPI est inhibé (SPE à 0)

□ **SPDR (Serial Peripheral Data Register I/O) : registre de données SPI, \$102A :**

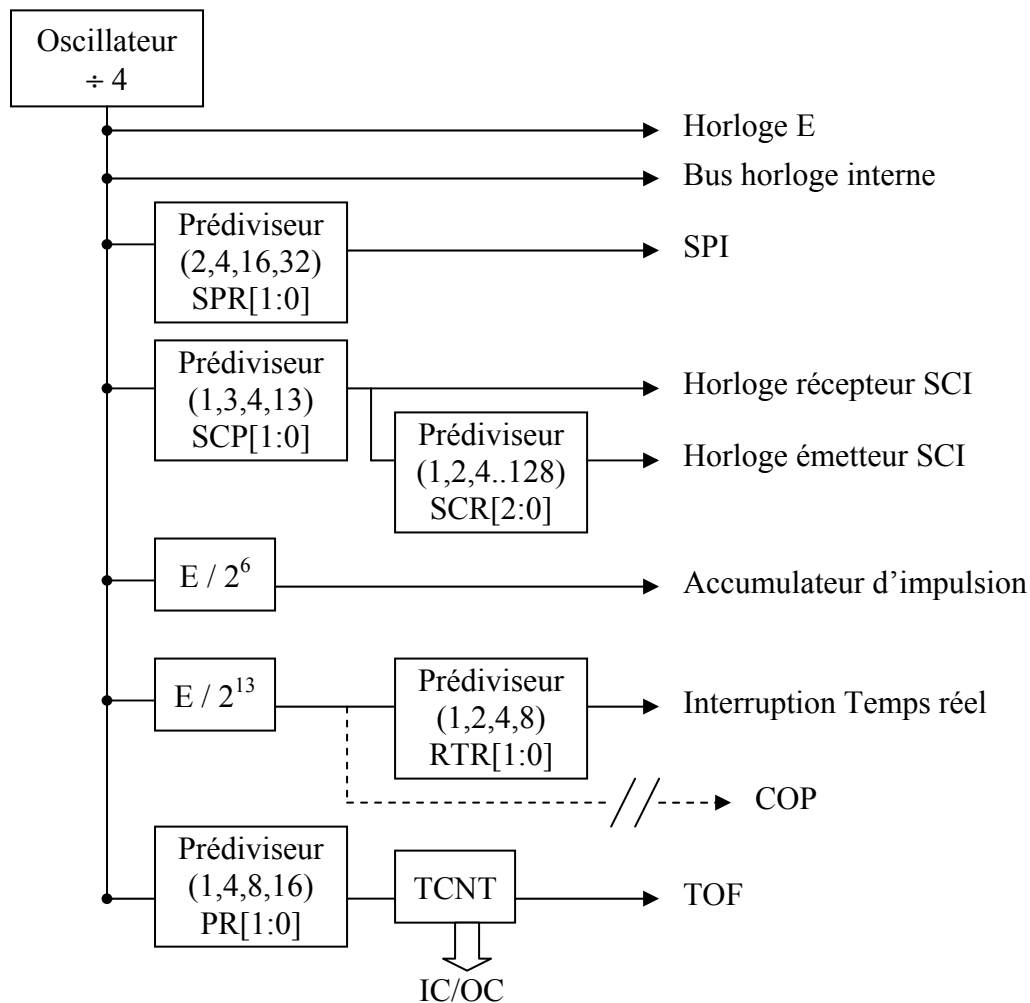
Bits :	7	6	5	4	3	2	1	0
--------	---	---	---	---	---	---	---	---

Ce registre est celui où s'effectue le transfert. On vient y écrire ou y lire les données à lire ou à écrire.

9.2 Périphériques connectables sur le bus SPI

MCM2818	EEPROM série 256 octets
MC68HC68T1	Calendrier perpétuel + 32 octets de RAM
MC14549	Convertisseur AD
MC145040	Convertisseur 10 voies AD 8 bits
MC145050	Convertisseur 10 voies AD 10 bits
MC144110	Convertisseur 6 voies DA 6 bits
MC144111	Convertisseur 4 voies DA 6 bits
MC14489	Driver d'afficheur 7 segments – 5 digits
MC145000	Driver d'afficheur LCD
MC74HC299	Registre à décalages 8 bits bidirectionnel

10. LE TIMER



Le système de compteur interne du MCU est composé de 5 chaînes de diviseur. Le diviseur de l'horloge principal comprend un compteur libre sur 16 bits (TCNT). Toutes les activités du système de comptage (*Timer*) sont référencées à ce compteur permanent. Le compteur compte sans arrêt l'horloge du 68HC11. Lorsque que le MCU sort du reset, le compteur commence à incrémenter à partir de \$0000 jusqu'à \$FFFF puis déborde et revient à \$0000 en positionnant à 1 le bit de débordement.

Sous-systèmes associés :

- IC (Input Capture) : Les captures en ENTREE permettent de mesurer des largeurs d'impulsions, de mesurer des fréquences ou de mesurer les durées entre deux évènements.
- OC (Output Compare) : Les comparaisons en SORTIE permettent de générer des signaux de rapport cyclique et de fréquence variables grâce à l'utilisation du compteur TCNT.
- PA (Pulse Accumulator) : L'accumulateur d'impulsions permet d'incrémenter un compteur (8 bits) au rythme d'impulsions asynchrones externes. Une interruption peut être générée lorsque ce compteur est plein.
- RTI (Real Time Interrupt) : L'interruption Temps Réel permet de générer une interruption à intervalle de temps régulier.

10.1 Le compteur

Il est constitué d'un compteur 16 bits, qui compte sans arrêt l'horloge du 68HC11. Il part de \$0000 jusqu'à \$FFFF puis déborde et revient à \$0000 en positionnant un bit de débordement. Il est précédé d'un prédiviseur programmable.

➤ **TMSK2**, Timer Interrupt Mask 2, \$1024 :

Bits :	7	6	5	4	3	2	1	0
Nom :	TOI	TRII	PAOVI	PAII	-	-	PR1	PR0
Reset :	0	0	0	0	0	0	0	0

PR[1:0] : Sélection du prédiviseur du Timer

PR[1:0]	Prédiviseur
0 0	1
0 1	4
1 0	8
1 1	16

Les bits PR1 et PR0 sont des bits protégés. Ils doivent impérativement être modifiés pendant les 64 cycles d'horloge qui suivent le Reset. Le bit 7 (TOI) mis à 1 autorise les interruptions produites par le passage à 1 du bit TOF du TFLG2 signalant le débordement du compteur.

Exemple : quartz = 8 Mhz → E= 2 Mhz donc T=500ns. La capacité maximale du comptage est : $65536 \times 16 \times 500\text{ns} = 524,288 \text{ ms}$ si Prédiviseur = 16.

➤ **TNCT**, Time Counter, \$100E-\$100F :

TNCT (high) \$100E :	15	14	13	12	11	10	9	8
TNCT (low) \$100F :	7	6	5	4	3	2	1	0

➤ **TFLG2**, Timer Interrupt Flag 2, \$1025 :

Bits :	7	6	5	4	3	2	1	0
Nom :	TOF	RTIF	PAOVF	PAIF	-	-	-	-
Reset :	0	0	0	0	0	0	0	0

Le bit TOF se positionne à 1 quand il y a débordement du compteur. Il reste dans cet état tant qu'il n'est pas remis à zéro par l'utilisateur. Pour cela il faut écrire un "1" au bit 7 dans ce registre TFLG2

10.2 Les registres de capture (IC) :

Quand un front arrive sur une des entrées de capture, la valeur du compteur interne est recopiée dans un des registres de 16 bits (TIC1, TIC2 ou TIC3). Une interruption est générée et un bit d'état (IC1F, IC2F ou IC3F) est mis à 1. Pour remettre à zéro ce bit, il faut écrire un 1 dans TFLG1 à la place du bit correspondant. Le choix du type de transition qui sera active sur l'entrée de capture se fait par le registre TCTL2.

Exemple :

On effectue la lecture et la sauvegarde du compteur sur les front successifs d'un signal présenté à l'entrée. On peut alors calculer la période et la largeur de ce signal. La période se calcule en moyennant le temps écoulé entre 2 fronts de même polarité (montant ou descendant) et la largeur est mesurée par le temps écoulé entre deux fronts de polarité inverse. On peut alors en déduire la cyclicité du signal.

Les bits de contrôle qui établissent les fonctions de détection d'impulsion sont contenus dans les registres PACTL, TCTL2, TMSK1 et TFLG1.

➤ **TCTL2**, Contrôle du timer 2, \$1021 :

Bits :	7	6	5	4	3	2	1	0
Nom :	EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A
Reset :	0	0	0	0	0	0	0	0

Les bits de ce registre permettent de programmer les fonctions de détection d'un signal présenté sur une broche d'entrée correspondante du Timer. Chacune des 4 fonctions de détection peut être configurée indépendamment.

EDGiB	EDGiA	Configuration
0	0	Capture d'impulsion inhibée
0	1	Capture sur front montant
1	0	Capture sur front descendant
1	1	Capture sur front quelconque

□ **Les quatre entrées de capture du port A :**

Bits :	7	6	5	4	3	2	1	0
Nom :	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
Reset :	X	X	X	X	X	X	X	X
Fonctions alternées :	PAI	OC2	OC3	OC4	IC4/OC5	IC1	IC2	IC3
Autres :	OC1	OC1	OC1	OC1	OC1			

➤ **TIC1 – TIC3**, Timer Interrupt Capture, \$1010-\$1015 :

TIC1 (high)	\$1010 :	15	14	13	12	11	10	9	8
TIC1 (low)	\$1011 :	7	6	5	4	3	2	1	0

TIC2 (high)	\$1012 :	15	14	13	12	11	10	9	8
TIC2 (low)	\$1013 :	7	6	5	4	3	2	1	0

TIC3 (high)	\$1014 :	15	14	13	12	11	10	9	8
TIC3 (low)	\$1015 :	7	6	5	4	3	2	1	0

➤ **TI4/O5**, Timer Input Capture 4/Output Compare 5, \$101E-\$101F :

TI4/O5 (high)	\$101E :	15	14	13	12	11	10	9	8
TI4/O5 (low)	\$101F :	7	6	5	4	3	2	1	0

Lorsqu'un front est détecté sur une des entrées de capture, la valeur du compteur TCNT est transférée dans le registre TIC_i correspondant.

➤ **TFLG1**, Timer Interrupt Flag 1, \$1023 :

Bits :	7	6	5	4	3	2	1	0
Nom :	OC1F	OC2F	OC3F	OC4F	I4/O5F	IC1F	IC2F	IC3F
Reset :	0	0	0	0	0	0	0	0

Les bits de ce registre indiquent l'existence d'évènements dans le système du compteur.

IC1F-IC3F (Input Capture i Flag) :

→ est mis à 1 pour chaque détection d'un front sur une entrée IC_i

➤ **TMSK1**, Timer Interrupt Mask 1, \$1022 :

Bits :	7	6	5	4	3	2	1	0
Nom :	OC1i	OC2i	OC3i	OC4i	I4/O5i	IC1i	IC2i	IC3i
Reset :	0	0	0	0	0	0	0	0

Ce registre est utilisé pour valider ou inhiber les interruptions générées par les fonctions de capture ou de comparaison du compteur.

IC1_i-IC3_i : Validation des Input Capture : Si le bit est à 1, une interruption est générée

10.3 Les registres de comparaison (OC) :

□ Les 4 sorties de comparaison du port A :

Bits :	7	6	5	4	3	2	1	0
Nom :	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
Reset :	X	X	X	X	X	X	X	X
Fonctions alternées :	PAI	OC2	OC3	OC4	IC4/OC5	IC1	IC2	IC3
Autres :	OC1	OC1	OC1	OC1	OC1			

➤ TOC1 – TOC4, Timer Output Capture, \$1016-\$101D :

TOC1 (high) \$1016 :	15	14	13	12	11	10	9	8
TOC1 (low) \$1017 :	7	6	5	4	3	2	1	0

TOC2 (high) \$1018 :	15	14	13	12	11	10	9	8
TOC2 (low) \$1019 :	7	6	5	4	3	2	1	0

TOC3 (high) \$101A :	15	14	13	12	11	10	9	8
TOC3 (low) \$101B :	7	6	5	4	3	2	1	0

TOC4 (high) \$101A :	15	14	13	12	11	10	9	8
TOC4 (low) \$101B :	7	6	5	4	3	2	1	0

Ces registres E/S, codés sur 16 bits, sont initialisés à \$FFFF après un reset.

La fonction de comparaison est établie en écrivant dans le registre concerné la valeur recherchée. Lorsque le compteur interne TCNT atteint une valeur identique, la broche OC correspondante est activée selon la logique établie par le programme.

➤ OC1M, Output Compare 1 Mask, \$100C :

Bits :	7	6	5	4	3	2	1	0
Nom :	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3	-	-	-
Reset :	0	0	0	0	0	0	0	0

OC1M[7:3] : Masque des comparaisons (=0 : OC1 est inhibé ; =1 : OC1 est validé pour contrôler les broches correspondantes du port A.

➤ OC1D, Output Compare 1 Data, \$100D :

Bits :	7	6	5	4	3	2	1	0
Nom :	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3	-	-	-
Reset :	0	0	0	0	0	0	0	0

Lorsque OC1Mi est à 1, la donnée dans OC1Di est affectée à la broche correspondante du port A lorsque la comparaison avec OC1 est vraie.

➤ **TCTL1**, Timer Control 1, \$1020 :

Bits :	7	6	5	4	3	2	1	0
Nom :	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5
Reset :	0	0	0	0	0	0	0	0

OM : Mode logique en sortie

OL : Niveau logique en sortie

OMi	OLi	Configuration
0	0	Pas d'action
0	1	Bascule de la ligne de sortie OC _i (changement d'état à chaque fois)
1	0	Met la ligne de sortie OC _i à 0
1	1	Met la ligne de sortie OC _i à 1

➤ **TMSK1**, Timer Interrupt Mask 1, \$1022 :

Bits :	7	6	5	4	3	2	1	0
Nom :	OC1i	OC2i	OC3i	OC4i	I4/O5i	IC1i	IC2i	IC3i
Reset :	0	0	0	0	0	0	0	0

Ce registre est utilisé pour valider ou inhiber les interruptions générées par les fonctions de capture ou de comparaison du compteur.

OC1i-OC3i : Validation des Output Compare : Si le bit est à 1, une interruption est générée

➤ **TFLG1**, Timer Interrupt Flag 1, \$1023 :

Bits :	7	6	5	4	3	2	1	0
Nom :	OC1F	OC2F	OC3F	OC4F	I4/O5F	IC1F	IC2F	IC3F
Reset :	0	0	0	0	0	0	0	0

Les bits de ce registre indiquent l'existence d'événements dans le système du compteur.

OC1F-OC4F (Output Compare i Flag) :

→ est mis à 1 pour chaque comparaison valide (la valeur du compteur TCNT est identique à la valeur définie pour la sortie i)

10.4 Real Time Interrupt (RTI) : L'interruption temps réel

Pour générer des interruptions à période fixe, on utilise l'interruption temps réel. Cette interruption est configurée par les 2 bits (RTR1 et RTR0) du registre de contrôle de l'accumulateur d'impulsion PACTL. Le bit RTII du registre TMSK2 valide cette fonction. Les quatre périodes possibles sont données par l'horloge E et la valeur des bits RTR[1:0] :

RTR[1:0]	DIV	E = x MHz	E = 2 MHz
0 0	1	$2^{13}/E$	4,096 ms
0 1	2	$2^{14}/E$	8,192 ms
1 0	4	$2^{15}/E$	16,384 ms
1 1	8	$2^{16}/E$	32,768 ms

Chaque débordement du compteur provoque la mise à 1 du bit RTIF (bit6) du registre TFLG2. Si le bit RTII (bit6) du registre TMSK2 est à un, une interruption est générée

➤ **TMSK2**, Timer Interrupt Mask 2, \$1024 :

Bits :	7	6	5	4	3	2	1	0
Nom :	TOI	RTII	PAOVI	PAII	-	-	PR1	PR0
Reset :	0	0	0	0	0	0	0	0

TOI	Timer Overflow Interrupt Enable 0 = Interruption TOF inhibée 1 = Demande d'interruption validée lorsque TOF est mis à 1
RTII	Real Time Interrupt Enable 0 = Interruption TOF inhibée 1 = Demande d'interruption validée lorsque RTIF est à 1
PR1 PR0	Prédiviseur du timer

➤ **TFLG2**, Timer Interrupt Flag 2, \$1025 :

Bits :	7	6	5	4	3	2	1	0
Nom :	TOF	RTIF	PAOVF	PAIF	-	-	-	-
Reset :	0	0	0	0	0	0	0	0

Les indicateurs de ce registre sont mis à zéro en écrivant un 1 dans le bit correspondant

TOF	Timer Overflow Interrupt Flag : Mise à 1 lorsque le registre TCNT passe de \$FFFF à \$0000
RTIF	Real Time Interrupt Flag : est automatiquement mise à 1 à la fin de chaque période RTI

10.5 L'accumulateur d'impulsions

Le MCU dispose d'un compteur 8 bits qui peut être configuré pour :

- compter les impulsions présentes qui sont appliquées sur la broche externe PAI (bit7 du port A). La fréquence de comptage maximum est E/2.
- L'accumulation d'impulsions pendant un temps de porte (gate time accumulation) sur la broche externe PAI, d'un signal de fréquence E/64.

Deux sources d'interruptions temps réel sont disponibles :

- quand le compteur déborde de \$FF à \$00.
- quand une transition valide est détectée PAI (bit7 port A)

□ Timing de l'accumulateur d'impulsions :

Quartz	Horloge E = (Quartz/4)	Période 1/E	2 ⁶ /E (64/E)	PACNT (overflow) 16384/E
8,0 MHz	2,0 MHz	500 ns	32 μs	8192 ms

➤ PACTL, Pulse Accumulator Control, \$1026 :

Bits :	7	6	5	4	3	2	1	0
Nom :	-	PAEN	PAMOD	PEDGE	-	I4/O5	RTR1	RTR0
Reset :	0	0	0	0	0	0	0	0

PAEN	Validation du système de comptage d'impulsions 0 = comptage d'impulsions inhibé 1 = comptage d'impulsions validé
PAMOD	Mode de fonctionnement du compteur d'impulsions 0 = comptage d'événements 1 = accumulation d'événements pendant un temps de porte
PEDGE	Contrôle du front de détection d'impulsions
I4/O5	Input Capture 4 / Output Compare 5 0 = Validation de la comparaison OC5 1 = Validation de la détection IC4
RTR[1:0]	Sélection de la période d'interruption RTI

PAMOD	PEDGE	Action déclenchée avec l'horloge
0	0	Le front descendant de PAI incrémente le compteur
0	1	Le front montant de PAI incrémente le compteur
1	0	Un niveau 0 sur PAI bloque le compteur
1	1	Un niveau 1 sur PAI bloque le compteur

➤ PACNT, Pulse Accumulator Count, \$1027 :

Bits :	7	6	5	4	3	2	1	0
Nom :	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

ce registre contient le comptage des événements externes arrivant sur la broche PAI.

11. LE CONVERTISSEUR ANALOGIQUE/DIGITAL

Le convertisseur analogique/digital est précédé d'un multiplexeur à 8 entrées.

Il est programmable à l'aide de deux registres (OPTION et ADCTL) et les résultats sont disponibles dans 4 registres (ADR1, ADR2, ADR3 et ADR4).

➤ **ADR1-ADR4**, registres des résultats de la conversion, \$1031-\$1034 :

\$1031	ADR1	7	6	5	4	3	2	1	0
\$1032	ADR2	7	6	5	4	3	2	1	0
\$1033	ADR3	7	6	5	4	3	2	1	0
\$1034	ADR4	7	6	5	4	3	2	1	0

➤ **OPTION**, \$1039 :

Bits :	7	6	5	4	3	2	1	0
Nom :	ADPU	CSEL						
Reset :	0	0						

ADPU	Mise en service du convertisseur 0 = A/D inhibé 1 = A/D activé
CSEL	Clock Select : Sélection de l'horloge pour le convertisseur 0 = Utilisation normale de l'horloge E

- Tension généralement utilisée : entre 0 et 5 V
- Si la tension d'entrée est égale à V_{rl} , le résultat de la conversion sera \$00. Si la tension d'entrée est égale à V_{rh} , le résultat de la conversion sera \$FF
- Pas d'indication de dépassement : si la tension est supérieure à 5 V, le résultat reste \$FF
- Le multiplexeur permet de sélectionner une entrée analogique parmi 16 (seules les 8 premiers canaux correspondent au port E)
- Trois fonctionnements possibles par programmation du registre ADCTL (Cf. littérature) :
 - en canal unique,
 - en mode multiplexé,
 - en mode Stop et Wait

➤ **ADCTL**, registre de contrôle du convertisseur A/D, \$1030 :

Bits :	7	6	5	4	3	2	1	0
Nom :	CCF	-	SCAN	MULT	CD	CC	CB	CA
Reset :	X	0	X	X	X	X	X	X

CCF	Conversions Complete Flag : Indicateur de fin de conversion. Cet indicateur est mis à 1 lorsque les 4 registres ADR1-ADR4 contiennent des résultats valides. Chaque opération d'écriture dans le registre ADCTL entraîne la mise à zéro automatique de ce bit et une nouvelle séquence de conversion est lancée
SCAN	Continuous Scan Control : 0 = Les 4 demandes de conversion sont effectuées une seule fois et les résultats sont transférés dans les registres ADR1-ADR4 1 = La séquence de conversion est effectuée en permanence et le registres ADR1-ADR4 sont mis à jour à chaque conversion
MULT	Multiple Channel/Single Channel control : Sélection du mode de fonctionnement en canal unique ou multiplexé 0 = Quatre conversions successives sur le canal choisi par CD-CA 1 = Une conversion sur chacun des 4 canaux. Chaque canal est alors affecté à un registre particulier
CD-CA	Channel Select D-A : Sélection des canaux analogiques D-A

CC	CB	CA	Entrée analogique	Résultat contenu dans ADR _i si MULT=1
0	0	0	AN0 sur PE ₀	ADR1
0	0	1	AN1 sur PE ₁	ADR2
0	1	0	AN2 sur PE ₂	ADR3
0	1	1	AN3 sur PE ₃	ADR4
1	0	0	AN4 sur PE ₄	ADR1
1	0	1	AN5 sur PE ₅	ADR2
1	1	0	AN6 sur PE ₆	ADR3
1	1	1	AN7 sur PE ₇	ADR4

12. ANNEXES

12.1 Choix du mode de fonctionnement.

Selon l'état logique des entrées MODA et MODB, le HC11F1 démarre dans un mode particulier :

MODB	MODA	Mode de fonctionnement	
1	0	Mode normal	Microcontrôleur ou monopuce
	1		Microprocesseur ou étendu
0	0	Mode spécial	Bootstrap
	1		Test

➤ *SPECIAL BOOTSTRAP*

Le mode Bootstrap permet le chargement d'un programme via la RS232 dans la RAM interne du microprocesseur. Après le Reset, le micro démarre le « Bootloader » de 192 octets qui se trouve en ROM de l'adresse \$BF40 à \$BFFF. Ce Bootloader charge les octets du programme arrivant sur la RS232. Ce programme, dont la taille est limitée à 512 octets dans le F1, doit démarrer en \$0000 car le Bootloader lancera l'exécution à partir de cette adresse lorsque le chargement du programme sera terminé. Il est préférable de quitter le mode Bootstrap dès que le programme principal commence son exécution, en écrivant la valeur correcte dans le registre HPRIO. Dès lors, le programmeur peut choisir l'un des trois autres modes de fonctionnement selon l'état des bits SMOD et MDA du registre HPRIO

➤ *MODE MONOUCHE (SINGLE CHIP)*

Le mode monopuce établit un fonctionnement dans lequel tout le programme d'application est contenu dans la mémoire interne du HC11. Dans ce mode, le microprocesseur dispose de ses ports en entrées-sorties classiques. Ce mode peut être établi après le chargement d'un programme (en spécial Bootstrap) par mise à zéro du bit SMOD dans le registre HPRIO. Dans ce mode, lors d'un reset, le MCU utilise uniquement les vecteurs établis dans l'EEPROM interne.

➤ *SPECIAL TEST*

Ce mode permet d'accéder à toutes les possibilités offertes par le HC11F1. Il convient de bien comprendre le fonctionnement du MPU avant de laisser une application « tourner » en Spécial test. En effet, toutes les protections internes sont désactivées et la destruction accidentelle (dans le sens logique!) de tout ou partie de la mémoire est possible.

➤ *MODE ETENDU (EXPANDED)*

- Dans ce mode, le microprocesseur accède à tout le plan d'adresse de \$0000 à \$FFFF, incluant les périphériques. Ce mode implique la perte des ports B, C, F et partiellement G. Pour palier son manque de ports utilisés pour adresser les mémoires, on utilise un 68HC24. C'est le mode de fonctionnement le plus pratique pour le F1 car ce microprocesseur ne dispose pas d'une mémoire morte interne suffisante pour stocker des programmes élaborés.

12.2 Les interruptions du 68HC11

Vecteurs d'interruption	Utilisation	Fonction
\$FFC0 - \$FFD5	Réservés	
\$FFD6 - \$FFD7	RS232	Port SCI
\$FFD8 - \$FFD9	SPIE	Port SPI
\$FFDA - \$FFDB	PAIi	Accu impulsions entrée.
\$FFDC - \$FFDD	PAOVi	débordement Accu impulsions
\$FFDE - \$FFDF	Toi	Débordement TIMER.
\$FFE0 - \$FFE1	OC5i	Input compare
\$FFE2 - \$FFE3	OC4i	Input compare 4
\$FFE4 - \$FFE5	OC3i	Input compare 3
\$FFE6 - \$FFE7	OC2i	Input compare 2
\$FFE8 - \$FFE9	OC1i	Input compare 1
\$FFEA - \$FFEB	IC3i	Input capture 3
\$FFEC - \$FFED	IC2i	Input capture 2
\$FFEE - \$FFEF	IC1i	Input capture 1
\$FFF0 - \$FFF1	RTIi	Interruption temps réel
\$FFF2 - \$FFF3	IRQ	Interruption externe
\$FFF4 - \$FFF5	XIRQ	Interruption externe
\$FFF6 - \$FFF7	SWI	Interruption logicielle
\$FFF8 - \$FFF9	Code opération illégal	
\$FFFA - \$FFFB	NOCOP	Défaut computer
\$FFFC - \$FFFD	CME	Surveillance horloge (défaut)
\$FFFE - \$FFFF	RESET	Vecteur de RESET.

12.3 Programmation en assembleur

□ Ecriture de programmes

Le programme en assembleur est le code source. La transformation du code source en code machine peut être réalisée par tout logiciel d'assemblage pour 68HC11 (AS11.EXE de Motorola). Ensuite, un « downloader » pour télécharger votre programme dans le 68HC11 (PCBUG11.exe) est nécessaire.

Le code source s'écrit dans un format très particulier. Chaque ligne de commande doit comporter quatre champs, séparés par des espaces ou des tabulations.

Champ	Contenu
étiquette	Etiquette (non obligatoire)
Code opératoire	Mnémonique
Opérande	Opérande
commentaire	Commentaire du programmeur (non obligatoire)

Remarque : une ligne sans étiquette doit commencer par un espace.

L'écriture du programme peut être effectuée avec un éditeur de texte. Le code source doit être sous forme ASCII pour être transformé correctement en code machine.

Exemple de sous-programme :

```
ENVINS    PSHY
          BLCR    PORTA,X %01000000    ; reset de RS
          STAA   PORTB/F,X
          BSET   PORTA,X %00000100    ; E = 0
          LDAA   #$4F                 ; affichage
          PULY
          RTS
```

12.4 Modes d'adressage du MC68HC11xx

Les MCU de la famille des MC68HC11xx possèdent 6 modes d'adressages :

- Immédiat,
- Direct,
- Etendu,
- Indexé,
- Inhérent,
- Relatif.

- **IMMEDIAT (indicateur #)**

L'argument est contenu dans l'octet qui suit le code opératoire. Le nombre d'octet qui suit l'instruction correspond à la taille du registre utilisé pour effectuer la fonction demandée.

Exemple: **LDAA #\$4F** ; Le registre A est chargé par la valeur hexa 4F
LDAA #%01001111 ; Le registre A est chargé par la valeur binaire 01001111

- **DIRECT (indicateur <)**

L'octet bas de l'adresse de l'opérande est contenu dans un seul octet qui suit le code opératoire car l'octet haut de cette adresse est \$00. Cela permet d'accéder directement à l'espace mémoire RAM compris entre \$00 et \$FF (l'opérande est à une adresse sur 8 bits). Cette méthode (rapide) doit être utilisée pour accéder aux données les plus fréquentes qu'on aura mis dans l'espace \$0000-\$00FF.

Exemple: **STAA <\$0F** ; Le contenu du registre A est rangé à l'adresse \$0F.

- **ETENDU (indicateur >)**

L'adresse effective de l'argument est contenue dans les deux octets qui suivent le code opératoire. Toute la mémoire dans l'espace adressable est accessible.

Exemple: **STAA >\$1000** ; Le contenu du registre A est rangé à l'adresse \$1000

- **INDEXE (indicateur ,)**

Un décalage (offset), codé sur un octet non signé, est ajouté à la valeur contenue dans un des registres d'index (IX ou IY) pour former l'adresse effective de l'argument. On accède à tout l'espace adressable par le MCU

Exemple: **LDX #\$1000**
LDAA 5,X ; Le registre A est chargé par l'octet de l'adresse \$1005.

- **INHERENT**

L'opérande est toujours un registre et son nom est codé directement dans le code opératoire de l'instruction.

Exemple: **TAB** ; Transfert du contenu du registre A vers le registre B.

- **RELATIF**

Ce mode est utilisé pour les instructions de branchement et permet de définir un saut du compteur programme (PC). L'utilisation des étiquettes permet de s'affranchir du calcul du saut nécessaire.

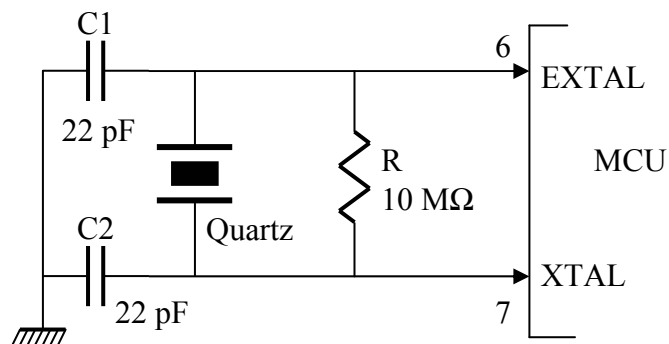
Exemple: **LDY \$FFFF** ; la valeur \$FFFF est chargé dans le registre Y
TEMPO DEY ; Y est décrémenté de 1
BNE TEMPO ; si Y différent de 0 alors boucle tempo
END ; si Y = 0 alors fin du programme.

12.5 XTAL et EXTAL : contrôle du quartz ou entrée d'horloge

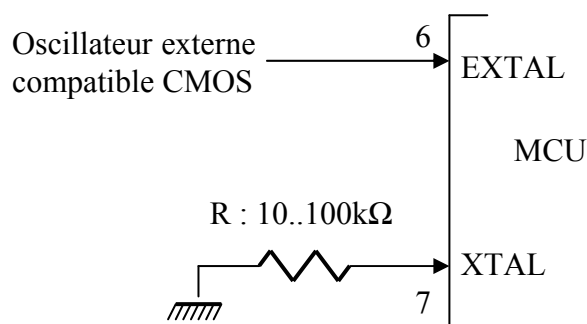
Pour cadencer le fonctionnement du MC68HC11F1, une horloge est nécessaire. XTAL et EXTAL permettent la connexion d'un quartz ou d'un oscillateur externe.

- La broche XTAL est à un niveau indéterminé lorsque l'on utilise une horloge externe de fréquence désirée sur la broche EXTAL.
- Les broches XTAL & EXTAL sont utilisées pour piloter un quartz oscillant à la fréquence désirée.

□ Connexion de l'oscillateur à quartz



□ Connexion d'un oscillateur extérieur



□ Connexion de deux MCU avec un unique quartz

