

5TC option Embed

MSP430 schématique et périphériques

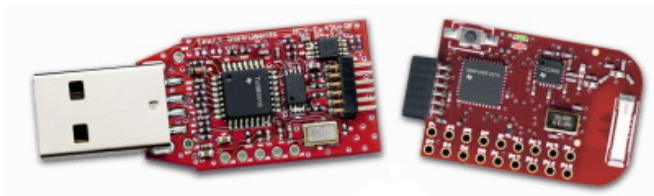
Antoine Fraboulet, Tanguy Risset

prénom.nom@insa-lyon.fr
Labo CITI, INSA de Lyon, Dpt Télécom



19 août 2016

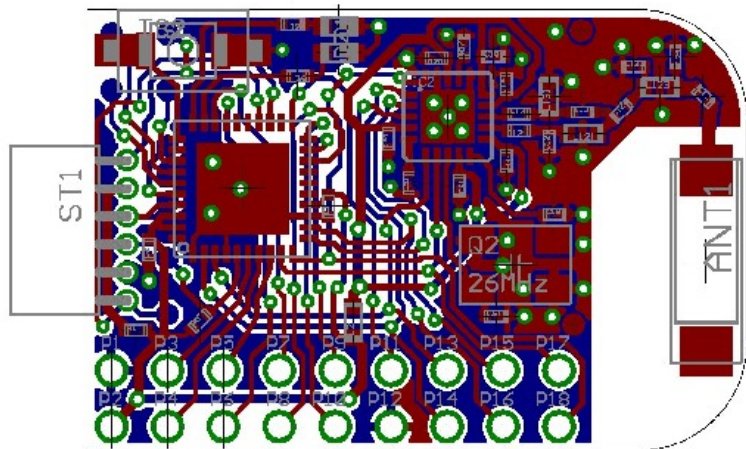
Présentation des périphériques



Le module eZ430-RF2500 est équipé d'un msp430f2274. Ces modèles de MSP ont les propriétés suivantes :

- Fréquence interne jusqu'à 16MHz
- TimerA avec 3 registres 16 bits
- TimerB avec 3 registres 16 bits
- USART avec UART + SPI + I2C
- Convertisseurs 10 bits

Schéma



Brochage du MSP430f2274

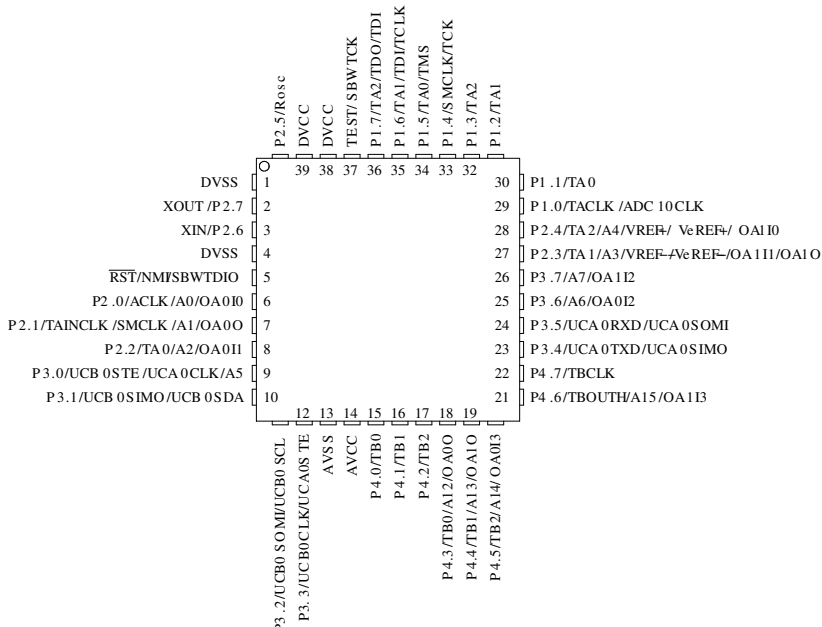
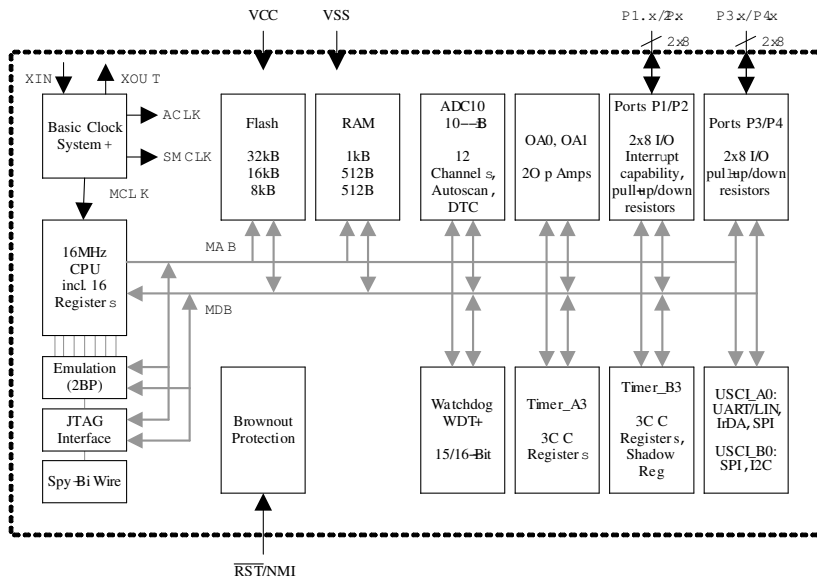


Schéma bloc du MSP430f2274

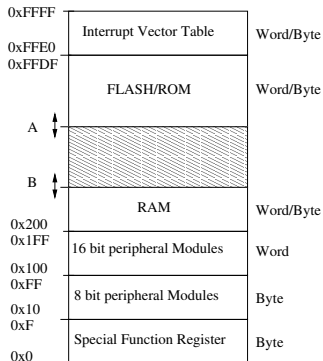


Pilotage des Entrées / Sorties

- Les broches sont multiplexées
 - Un port peut être utilisé comme I/O générique
 - Il peut correspondre à une fonction (bloc de périphérique)
- Programmation logicielle
 - PxSEL : sélection IO / périphérique
 - PxIN : registre des données en lecture
 - PxOUT : registre des données en écriture
 - PxDIR : direction du port
 - Les blocs sont pilotés à des adresses fixes.

Mapping mémoire des MSP430

- Accès aux registres des périphériques
- Chaque registre est mappé en mémoire :



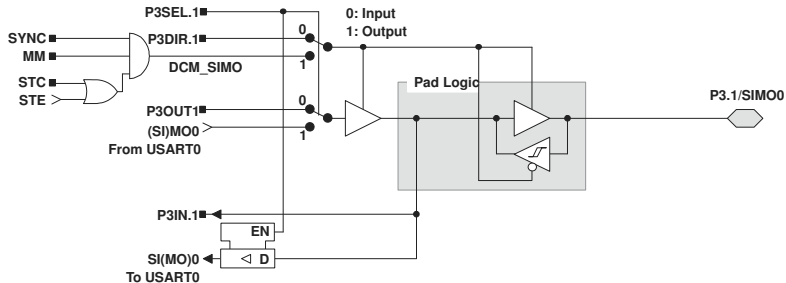
```
volatile unsigned char P5OUT asm("0x0031");  
P5OUT = 0x13;
```

```
char *p5 = (char*)0x0031;  
*p5 = 0x13;
```

Exemple de port (MSP430f149)

input/output schematic (continued)

port P3, P3.1, input/output with Schmitt-trigger

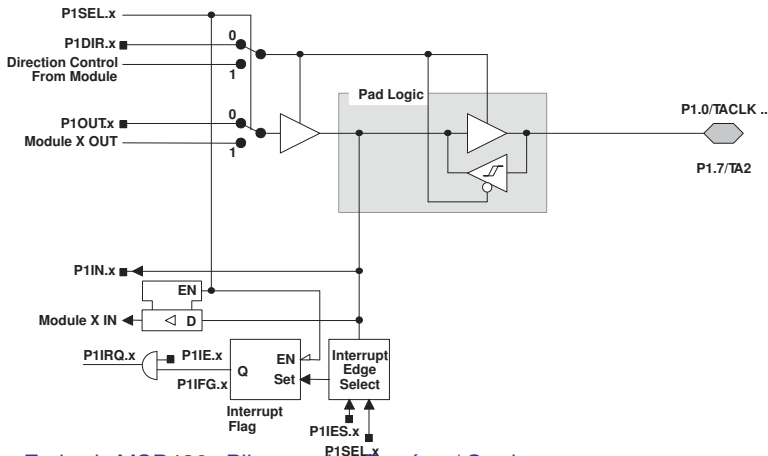


Exemple de port (MSP430f149)

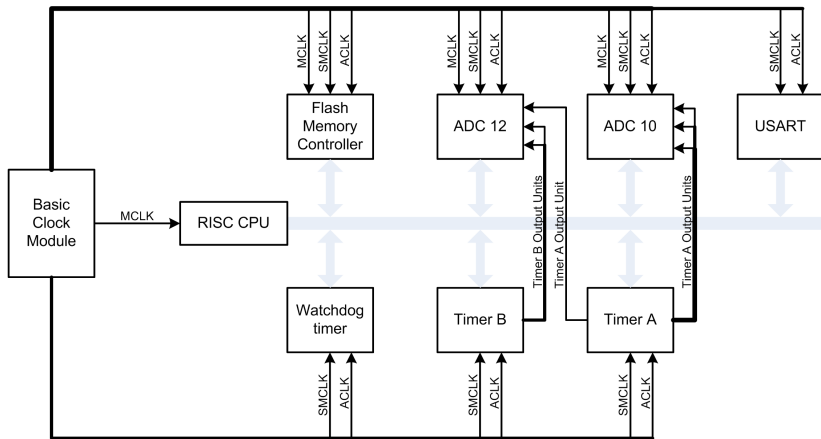
- Ports 1 et 2 avec interruptions
- Registres PxIFG (flag) et PxIE (interrupt enable)

input/output schematic

port P1, P1.0 to P1.7, input/output with Schmitt-trigger

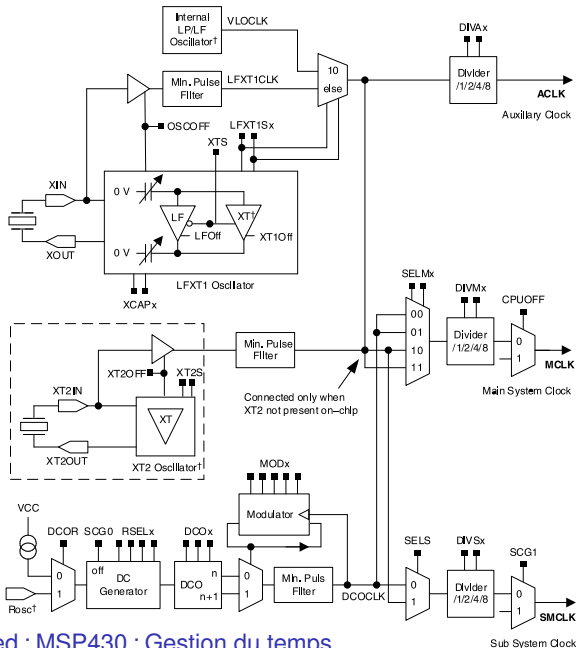


Gestion des horloges



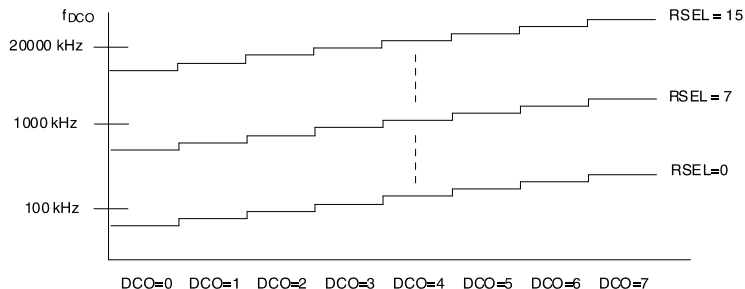
- Un seul bloc d'horloge est utilisé comme référence
- Les blocs doivent être paramétrés en fonction de ces références

Basic Clock +



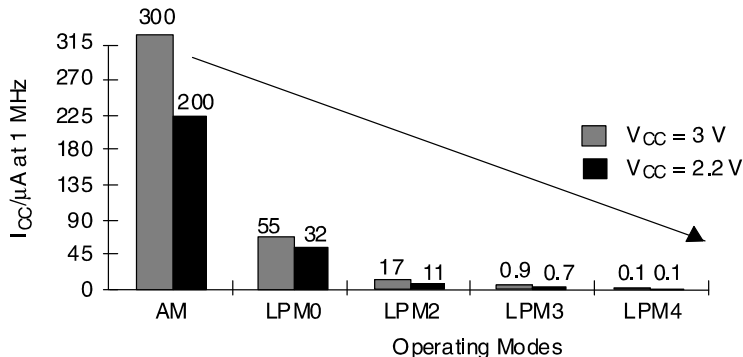
Réglage du DCO

Le bloc DCO est programmable à l'aide de son jeu de registre. Les fréquences atteignables couvrent toute la plage de 1.1MHz à 16MHz. La dérive de la fréquence est sensible.



Modes de veille

- La consommation du MSP430 est gérée grâce à des modes de veille
- Le premier mode permet de couper les instructions
- Les autres modes coupent les autres horloges



Modes de veille

- Macro LPM0,
 - Modification du bit CPUOFF du registre de contrôle
- La sortie d'un mode de sommeil est faite à la réception d'une interruption (timer)
- LPM0_EXIT modifie la valeur du bit CPUOFF dans la sauvegarde de SR sur la pile
- Désactivation partielles des périphériques du MSP
 - LPM0 : horloge MCLK principale désactivée
 - LPM1 : MCLK et DCO désactivés
 - LPM2 : MCLK, DCO, SMCLK
 - ...
- Utiliser les interruptions pour contrôler le programme
- Reste la mise en veille des périphériques

Interruptions

- Interruptions à 2 niveaux :
 - GIE dans le registre SR
 - Gestion des sources dans les blocs
- Interruptions à source uniques : l'acquittement est implicite
- Interruptions à sources multiples : acquittement nécessaire
 - Bits PxIE + PxIES + PxIFG des ports 1 et 2.
 - Comparateurs des registres des timers
 - ...
- Les interruptions sont, par défaut, non ré-entrantes. Le bit GIE est mis à 0 par le matériel à l'entrée d'un gestionnaire d'interruption.

Ecriture d'un gestionnaire d'interruption

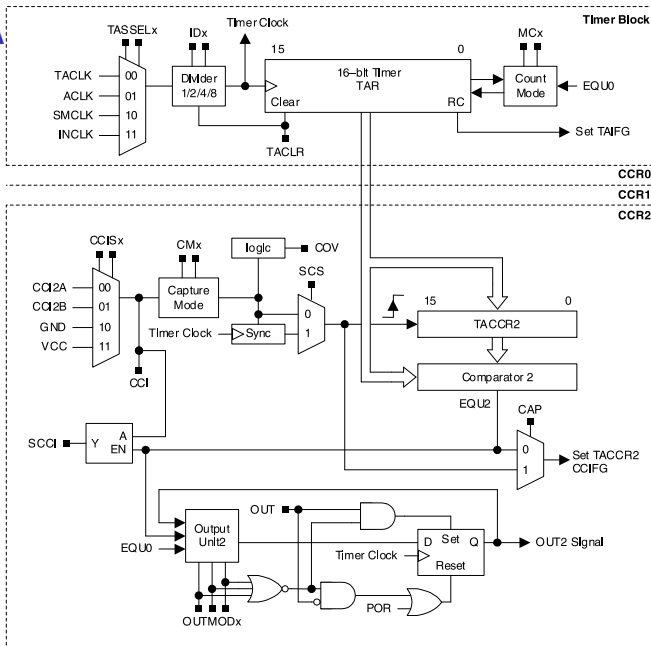
```
interrupt (PORT1_VECTOR) port1_irq_handler(void)
port1_irq_handler(void)
{
    if (P1IFG & (P1IE & (1 << 2)))
    {
        SWITCH_RED_LED();
    }
    P1IFG = 0;
}

/*
   #pragma vector=PORT1_VECTOR
   __interrupt void port1_irq_handler(void)
*/
```

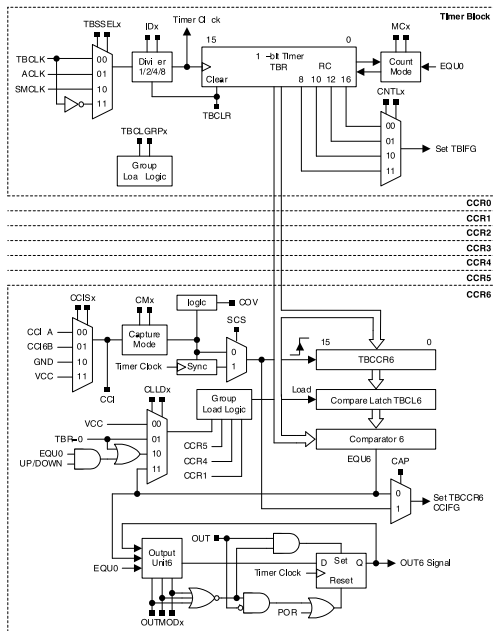
Timers

- Utilisation d'une horloge de référence par bloc
- Comparateurs multiples et gestion des interruptions
- Interruptions
 - acquittement automatique (TxCCR0)
 - acquittement logiciel (TxIV)
 - Les interruptions sont priorisées
- Les timers contiennent des registres sur 16 bits
- Les différents modes permettent d'avoir une gestion efficace du temps

Timer A



Timer B



CCR0

CCR1

CCR2

CCR3

CCR4

CCR5

CCR6

CCR6

CCR6

CCR6

CCR6

CCR6

CCR6

CCR6

CCR6

CCR6

CCR6

CCR6

CCR6

CCR6

CCR6

CCR6

CCR6

CCR6

CCR6

CCR6

CCR6

CCR6

CCR6

CCR6

CCR6

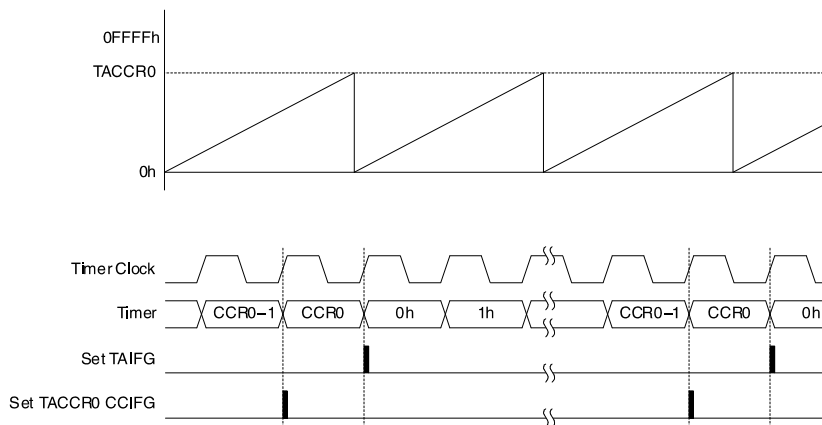
CCR6

CCR6

CCR6

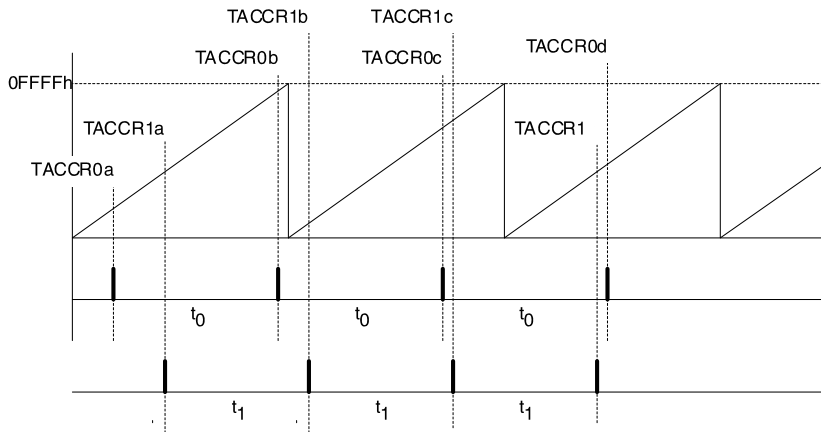
Timers et modes

- Le mode UP est sans doute le plus utilisé.



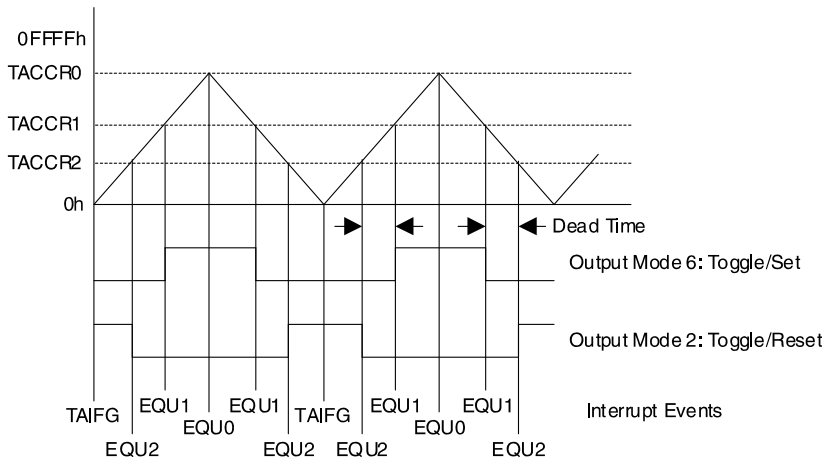
Timers et modes (2)

- *Continuous Mode*



Timers et modes (3)

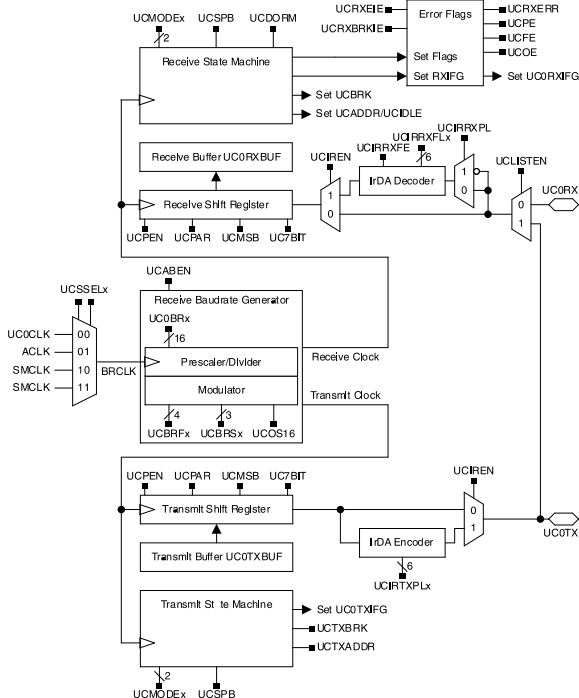
- *Up/Down Mode*



Ports série

- Plusieurs modèles disponibles
 - Certains sont en mode exclusif
 - ▶ Modules USI (SPI, I2C)
 - ▶ Modules USART (UART, SPI, I2C)
 - Ceux du 2274 sont doubles (Modules USCI)
 - ▶ USCI_Ax (UART + IrDA + SPI)
 - ▶ USCI_Bx (SPI + I2C)
- Les ports séries peuvent se piloter par interruption
- Le débit est calculé à partir d'une horloge de référence
- Le débit est dépendant des horloges externes

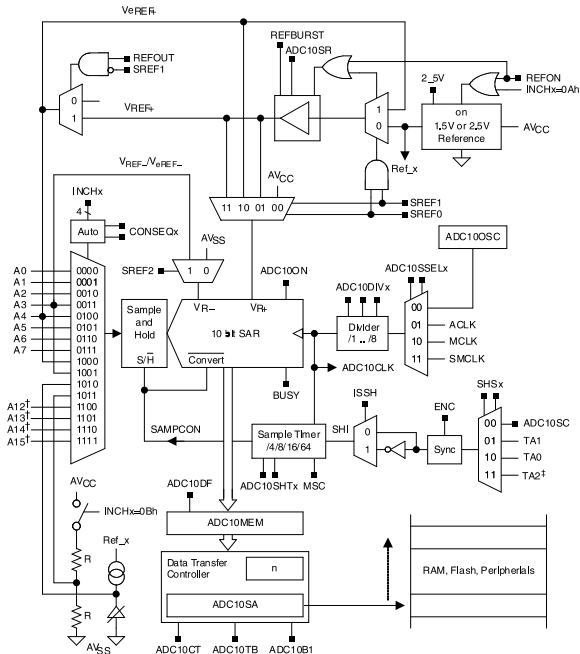
USCI A



Ports série

- Les paramètres à utiliser pour la modulation sont à choisir dans les tables fournies par le manuel.
- La fréquence de référence ne doit pas changer.
- Le choix de la fréquence de référence est donc important.
- Si beaucoup d'erreurs apparaissent il est conseillé de passer à un code correcteur d'erreur logiciel.

ADC 10



ADC 10

- La programmation de l'ADC reste dans le modèle que pour les autres composants :
 - Choisir une fréquence d'échantillonnage par rapport à une fréquence de référence
 - Choisir un canal à numériser
 - Attendre la stabilisation
 - Demander la lecture
 - Attendre la validation
 - Possibilité de recevoir une interruption au besoin