Introduction aux périphériques du PIC18

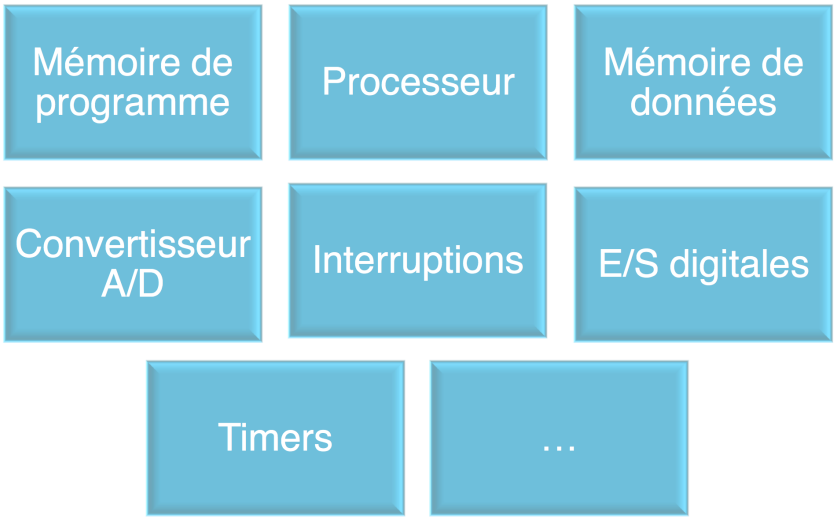
LLSMF2018 : Projet Technologique

UCL

20 octobre 2014

Plan

Les périphériques du microcontrôleur



- La communication bidirectionnelle entre le processeur et ses périphériques se fait au travers de "registres à fonctions spéciales" = emplacements dans la mémoire qui ont une fonction particulière.
- Ces registres à fonctions spéciales ont des noms pré-définis que l'on peut utiliser dans notre programme.

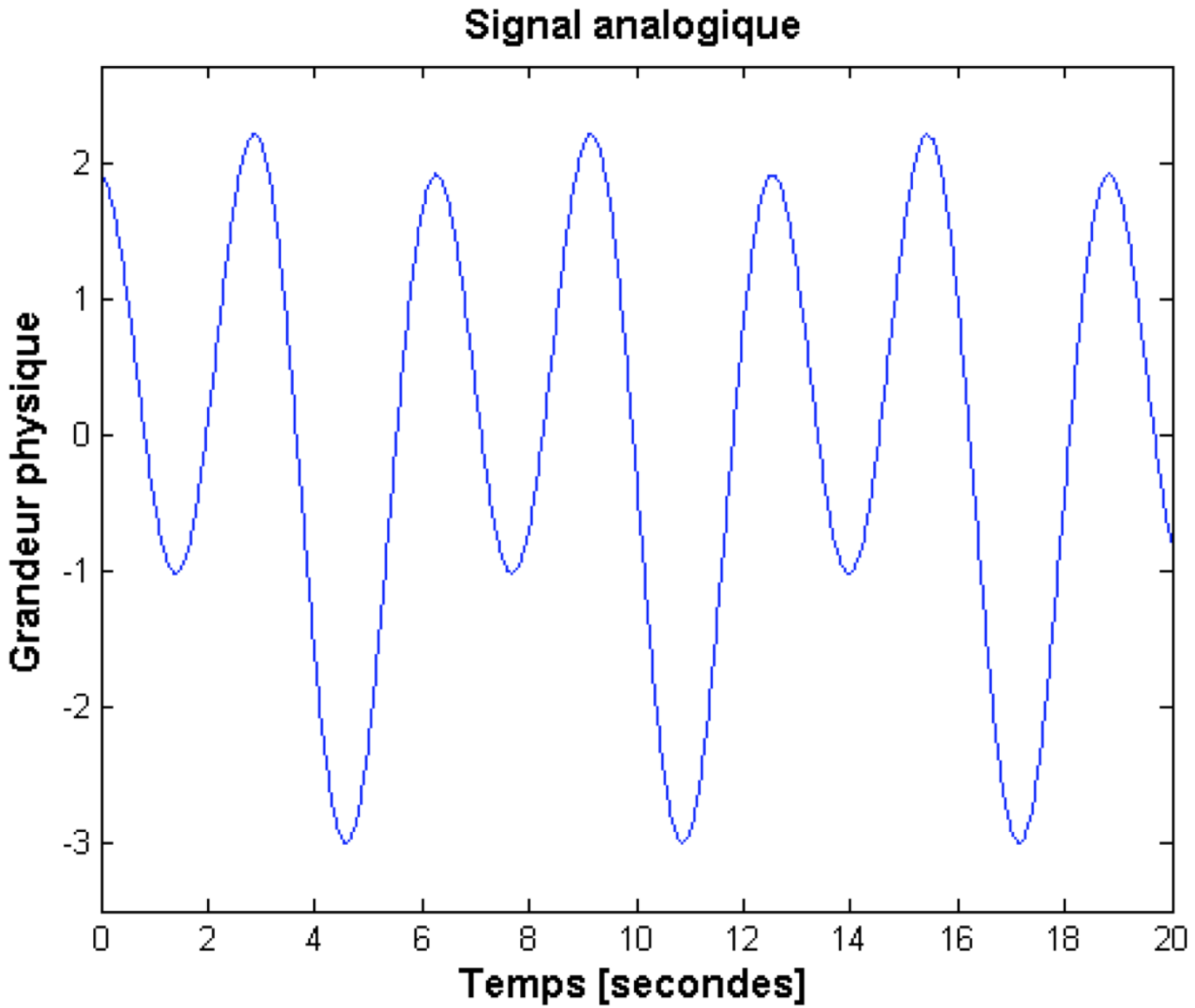
Entrées/sorties digitales

- Un port d'E/S est un module qui contrôle 8 broches du microcontrôleur
- La communication entre le module et le programme a lieu au travers de 3 registres à fonctions spéciales
  - TRIS[x] : direction (x : lettre du port)
  - PORT[x] : lecture des pins
  - LAT[x] : écriture d'un état logique
- Remarque : on peut accéder à un bit particulier de la manière suivante :
  - LAT[x]bits.LAT[x][n] (n : le numéro de la broche [0..7])
  - PORT[x]bits.R[x][n]
  - TRIS[x]bits.TRIS[x][n]

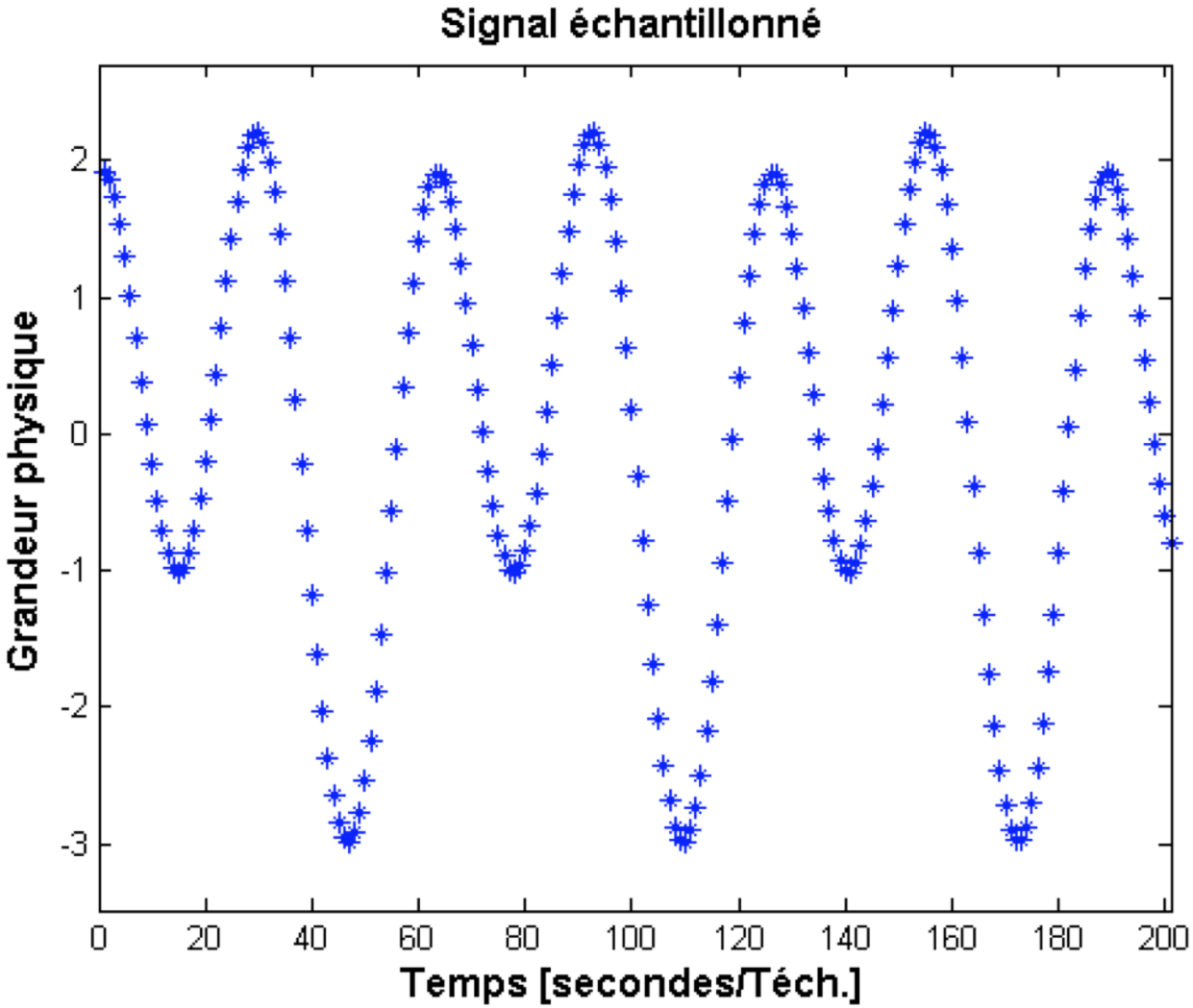
Entrées/sorties digitales, Exemples

- getButton()
  Le bouton est connecté sur la broche 0 du port B
- setLeds() Les LEDs sont connectées sur toutes les broches du port D
- LATDbits.LATB1 = 1 ; <=> LATD | 0x02 ;
- LATDbits.LATB2 = 0 ; <=> LATD & 0xFB ;

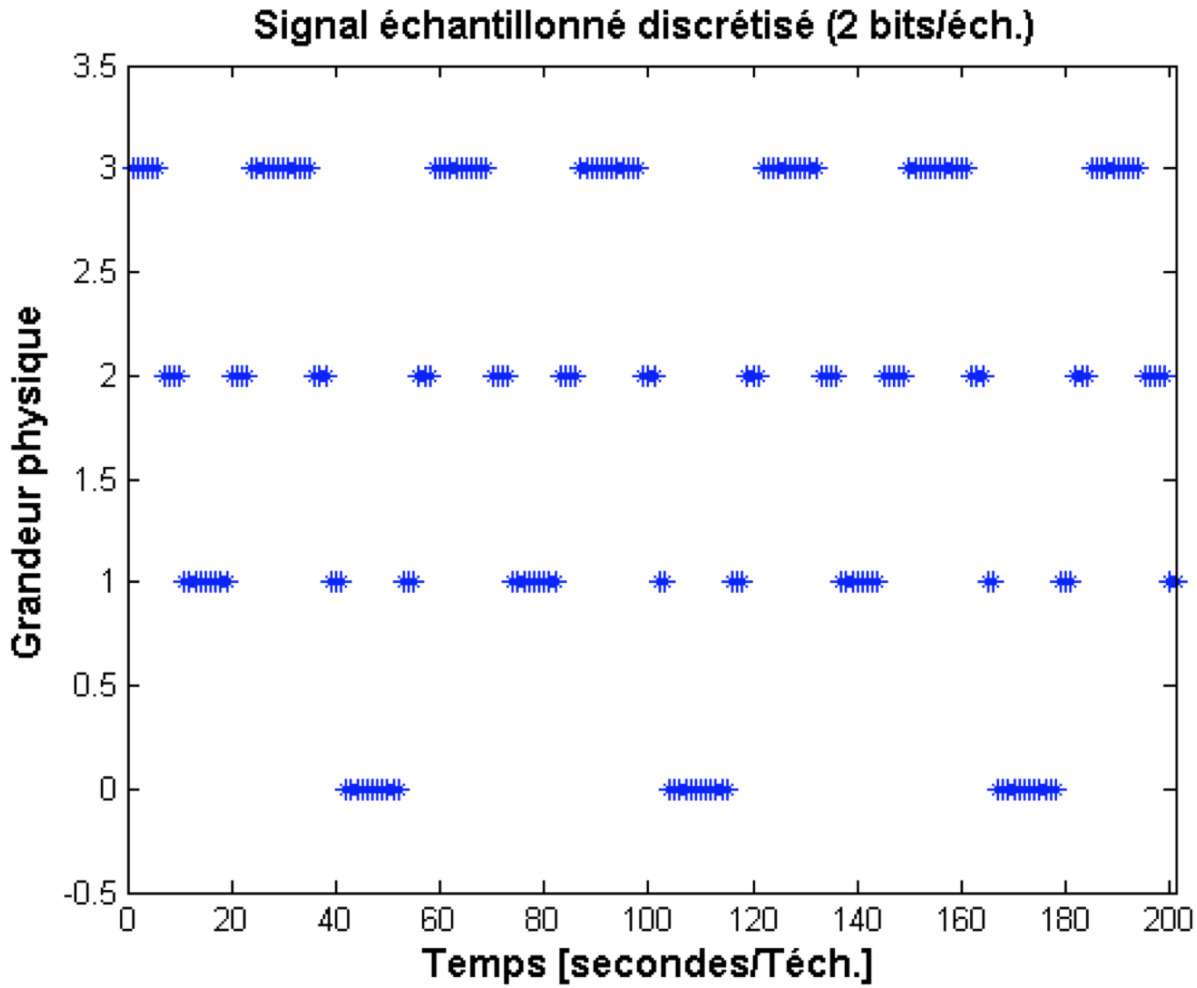
Conversion analogique-digital (1)



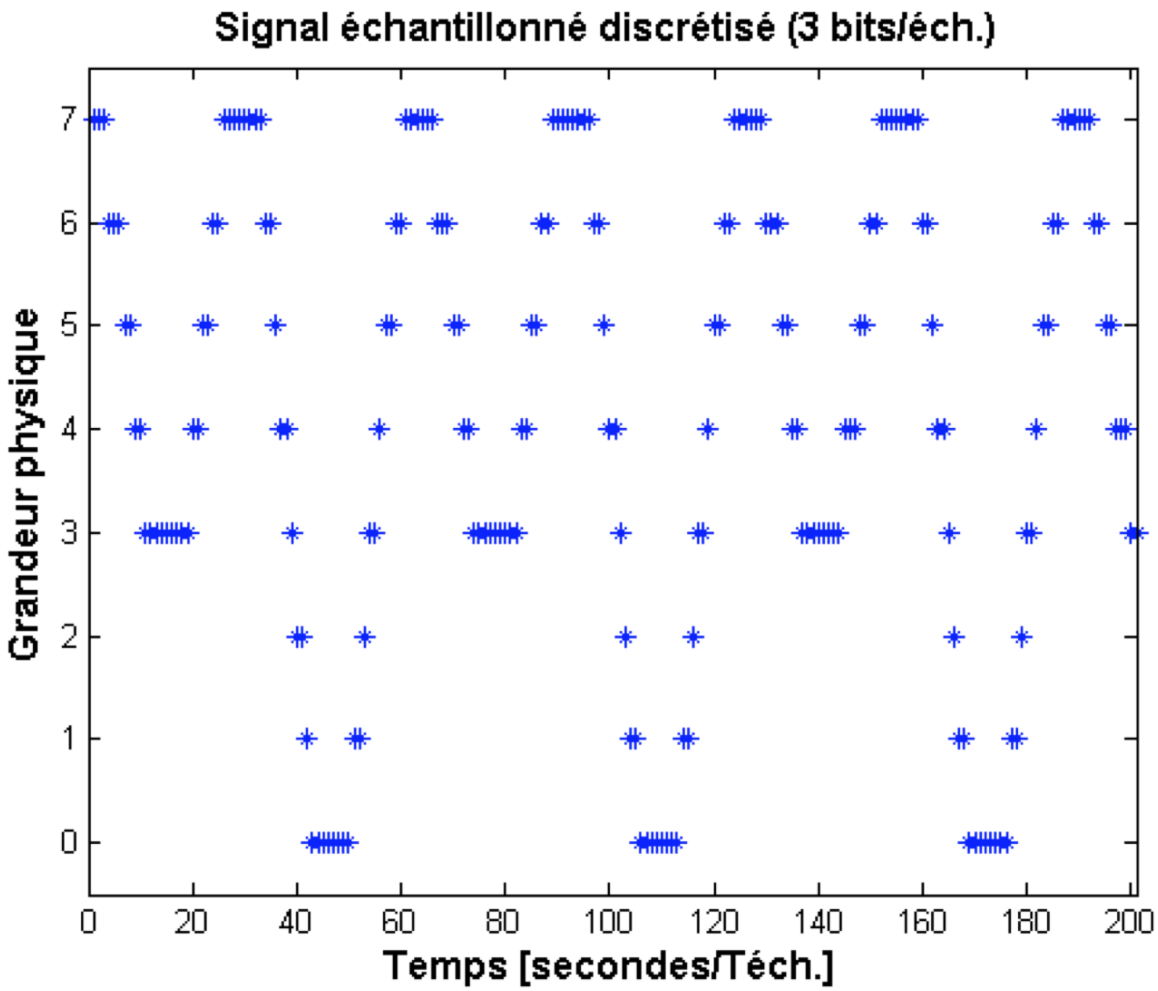
Conversion analogique-digital (2)



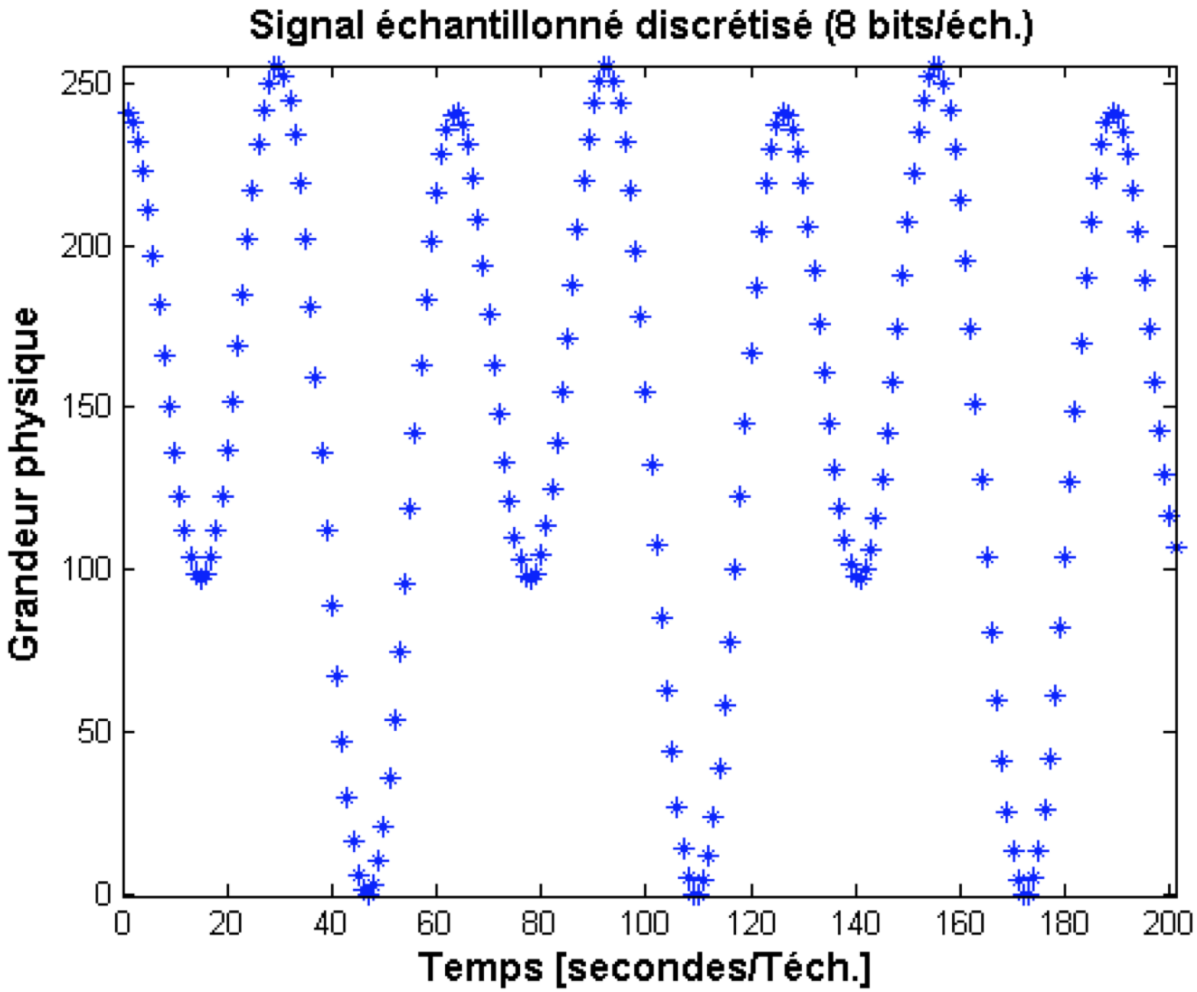
Conversion analogique-digital (3)



Conversion analogique-digital (4)



Conversion analogique-digital (5)



Conversion analogique-digital : à retenir

- Deux fonctions utiles (pour le moment)
  - void ADC_Init(void)
  - unsigned char getVoltage(void)

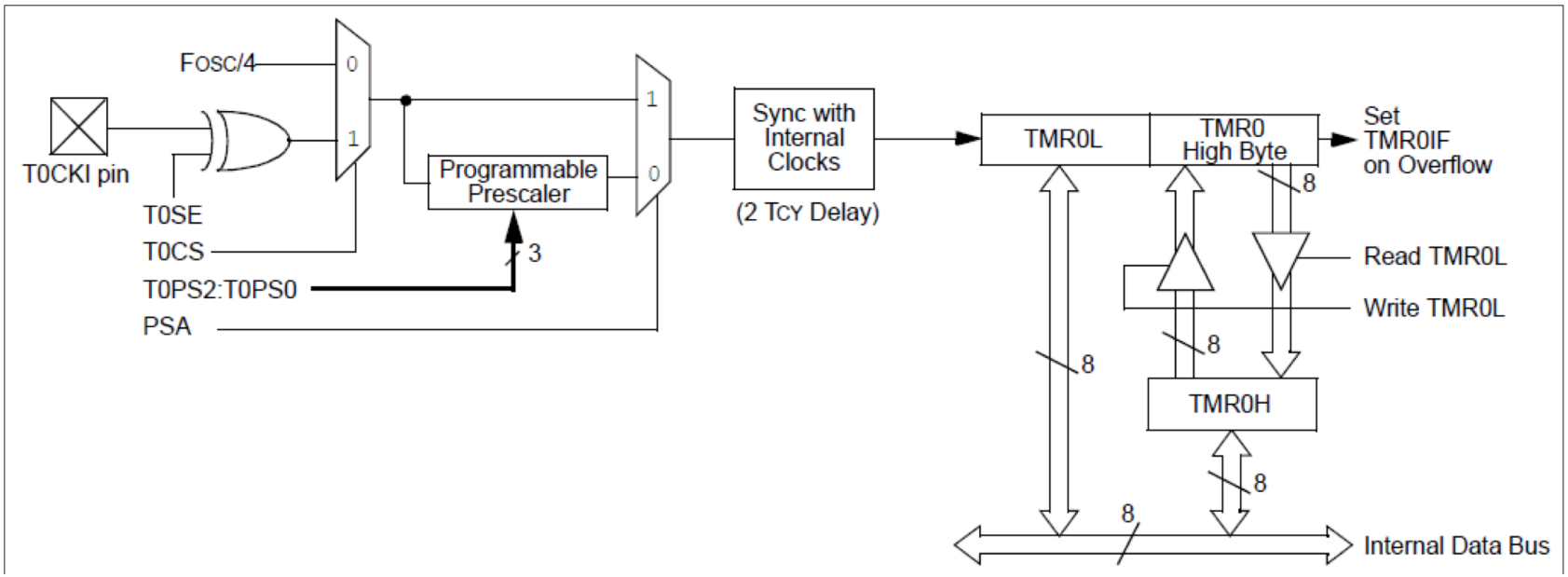
Interruptions

- Rappels
  - Une demande d'interruption est émise pas un des modules périphériques via la mémoire.
  - Si cette interruption est acceptée par notre configuration (interruption en général et interruption particulière), le code de l'interruption est exécuté.
  - À la fin de cette exécution, le programme continue exactement où il se trouvait avant l'interruption.
  - La demande d'interruption doit être effacée manuellement par routine d'interruption elle-même.

Interruptions : Fonctions

- void interruptEnable(void)
- void interruptDisable(void)
- void buttonInterruptEnable(void)
- void buttonInterruptDisable(void)
- void clearButtonInterruptRequest(void)

Timers : fonctionnement



Timers : Fonctions

- initTimer(unisgned char preDivision, unsigned char priority)
  preDivision :
  - 111 = 1 :256
  - 110 = 1 :128
  - 101 = 1 :64
  - 100 = 1 :32
  - 011 = 1 :16
  - 010 = 1 :8
  - 001 = 1 :4
  - 000 = 1 :2
- void timerInterruptEnable(void)
- void timerInterruptDisable(void)
- void clearTimerInterruptRequest(void)