External interrupt –AtMega328

Tous les microcontrôleurs peuvent créer une interruption si une ou plusieurs pins changent d'état. Pour l'AVR 328, ce sont les pins 2 et 3 associées aux interrupts Int0 et Int1. Un "Interrupt on pin change" concerne des groupes de pins, par exemple pour un clavier balayé, nous n'en parlons pas ici.
Attention, les interruptions réagissent en moins d'une microsecondes. Lire un poussoir par interruption ne peut pas de faire sans autre (voir en annexe).

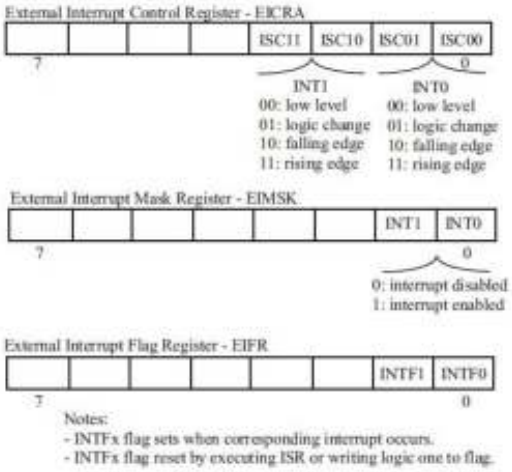
Appel Arduino
Arduino a mis en place une fonction pour cacher les registres qui gère cet interruption.
http://arduino.cc/en/Reference/attachInterrupt

La fonction `attachInterrupt(pin,function,mode)` fait tout ce qu'il faut.

`pin`	0 ou 1 selon la pin 2 ou 3 (3 ou 2 selon la carte)
`function`	le nom de la fonction appelée
`mode`	`0,1,2,3 ou` LOW, CHANGE, RISING, FALLING

Exemple: un capteur de Hall tout-ou-rien est utilisé comme fin de course sur la pin 2. On mettra dans le set-up `attachInterrupt(0,StopMoteur(),3);` La fonction `StopMoteur()` sera appelée toutes les fois quand le signal capteur passe de 1 à 0.

Appel direct en C
Il y a trois registres dans le processeur pour gérer ces deux pins d'interruption.
Le registre EICRA "External Interrupt Control Register" ne contient que 2 fois 2 bits qui pour chaque pin codent le mode vu plus haut
Le registre EIMSK "External Interrupt Mask Register" contient les 2 bits INT0 INT1 qui autorisent les interruptions 0 et 1.
Le registre EIFR "External Interrupt Flag Register" contient les 2 bits INTF0 INTF1qui sont activés par les interruptions 0 et 1.
Il faut encore savoir le nom de la fonction que le compilateur sait associer au vecteur d'interruption: ISR(INT0_vect) { nos instructions }.



Dans le set-up il faudra écrire

```
EICRA = 3 ;  // actif à 0
EIMSK = 1<<INT0;
sei();  // active toutes les interruptions
```

On a ensuite la fonction/routine/procédure d'interruption

```
ISR(INT0_vect) {
  // ce qu'il faut faire
}
```

Le flag INTF0 est automatiquement mis à zéro.

S'il y a des rebonds de contact, une solution est d'attendre quelques ms avant de réactiver l'interruption. Ceci doit se faire avec un timer par interruption si on ne veut pas que le programme principal soit bloqué. L'external interrupt n'est absolument pas adapté pour lire un poussoir, il faut des signaux propres. On peut ajouter de l'électronique pour supprimer les rebonds avant d'envoyer le signal sur la pin d'interruption, mais c'est une solution archaïque.

jdn 131110/20/140520