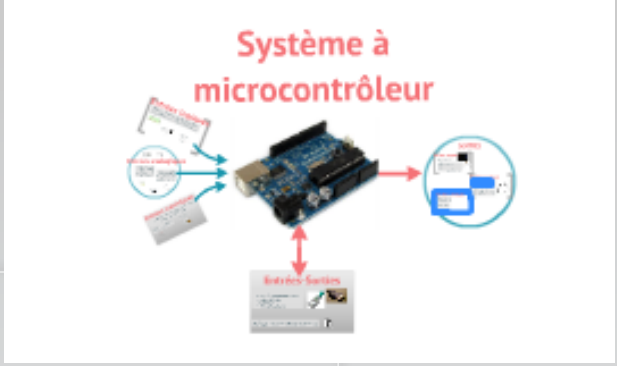
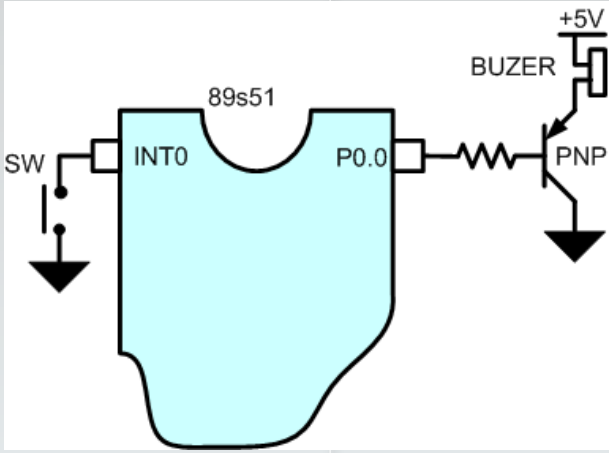
Interruption:



Une interruption, c'est un événement externe ou interne au système qui interrompt le déroulement du programme en cours et oblige le système à exécuter une routine particulière.

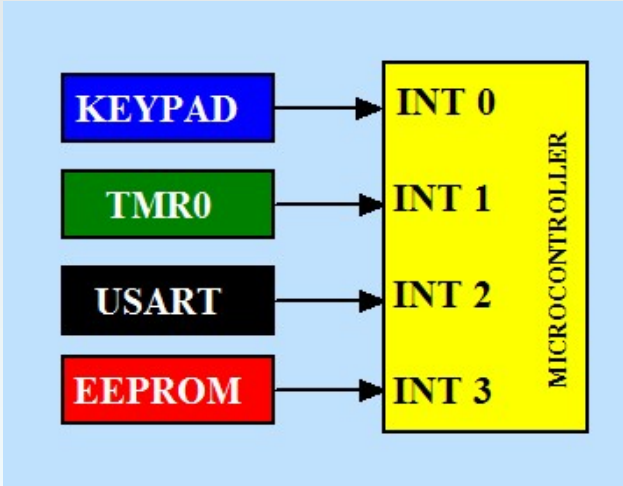
Événement externe:

- Changement sur une entrée spécifique appelée entrée d'interruption. (tout µC en a au moins une)
- Changement sur un bit de port. (sur le PIC16F877, seuls les bits PB_4 à PB_7 peuvent déclencher une interruption.)

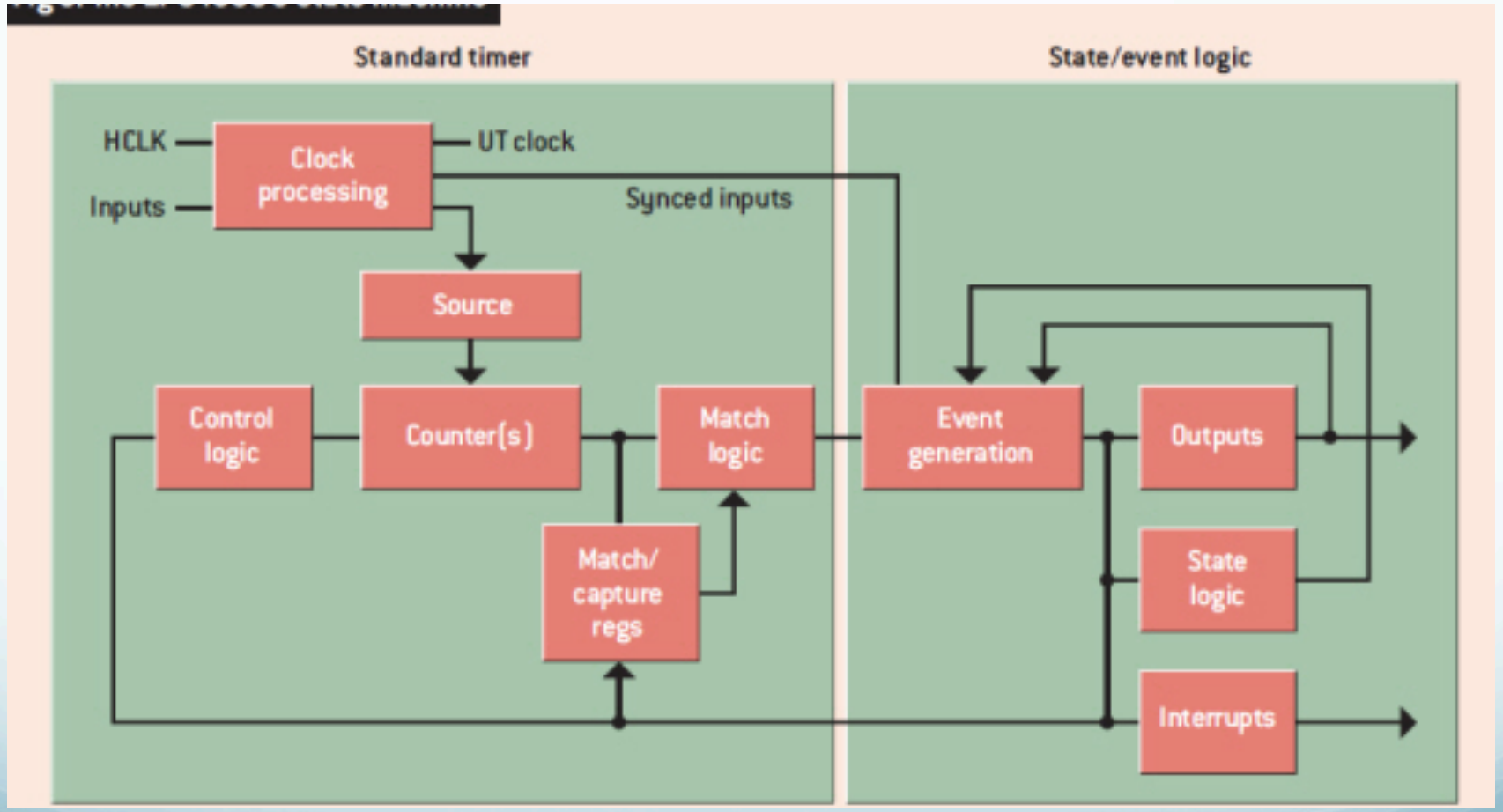


Événement interne:

- Fin de conversion A/N
- Événement sur le bus série (fin d'émission, réception …)
- Événement I²C ou SPI
- Débordement Timer



Machine d'Etat

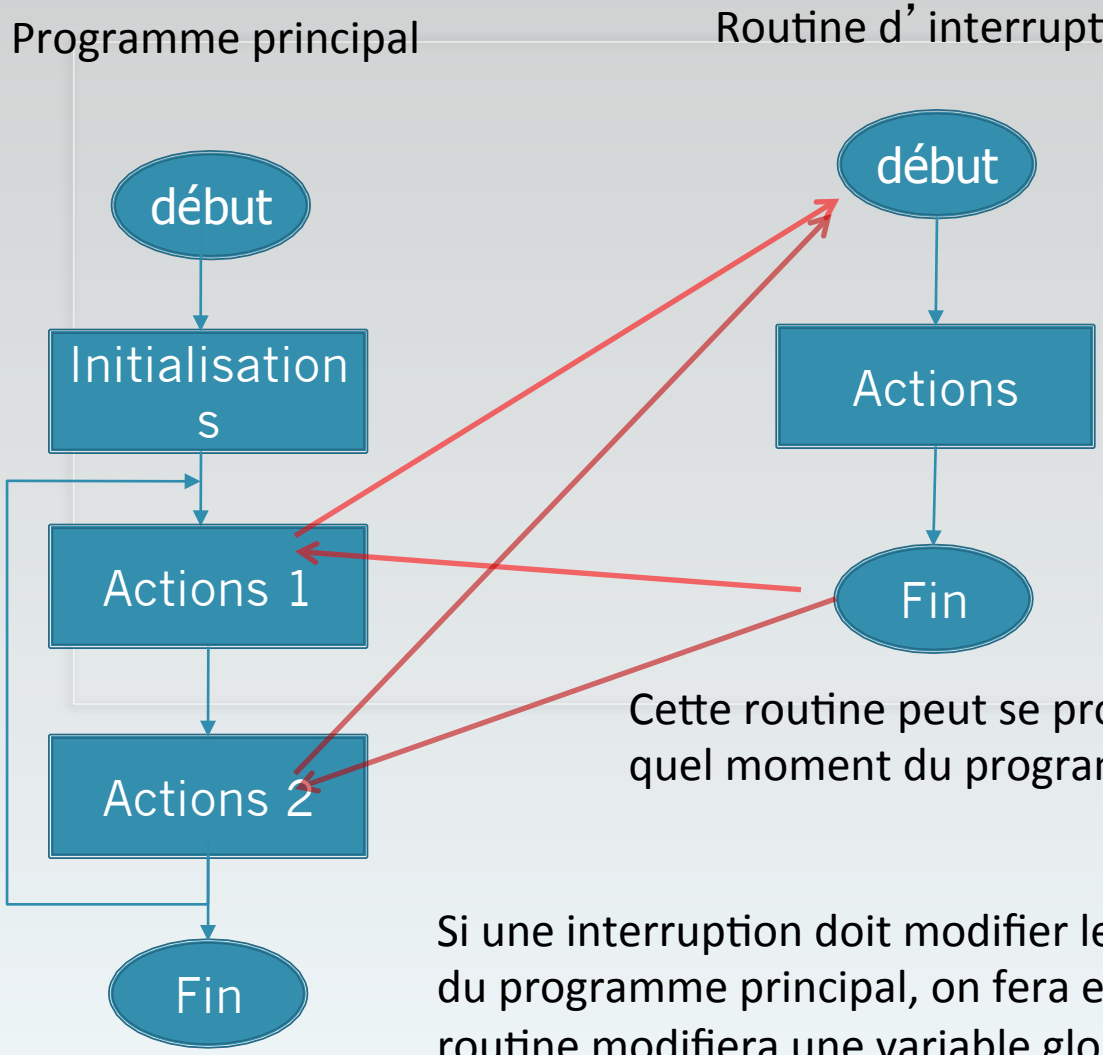


Autorisation des Interruptions

Sur les PIC, il y a 2 niveaux d'autorisations:

1. **Autorisation individuelle**: on configure la ou les interruptions autorisées.
2. **Autorisation globale**: on autorise ou interdit toutes les interruptions (qui doivent être autorisées individuellement).

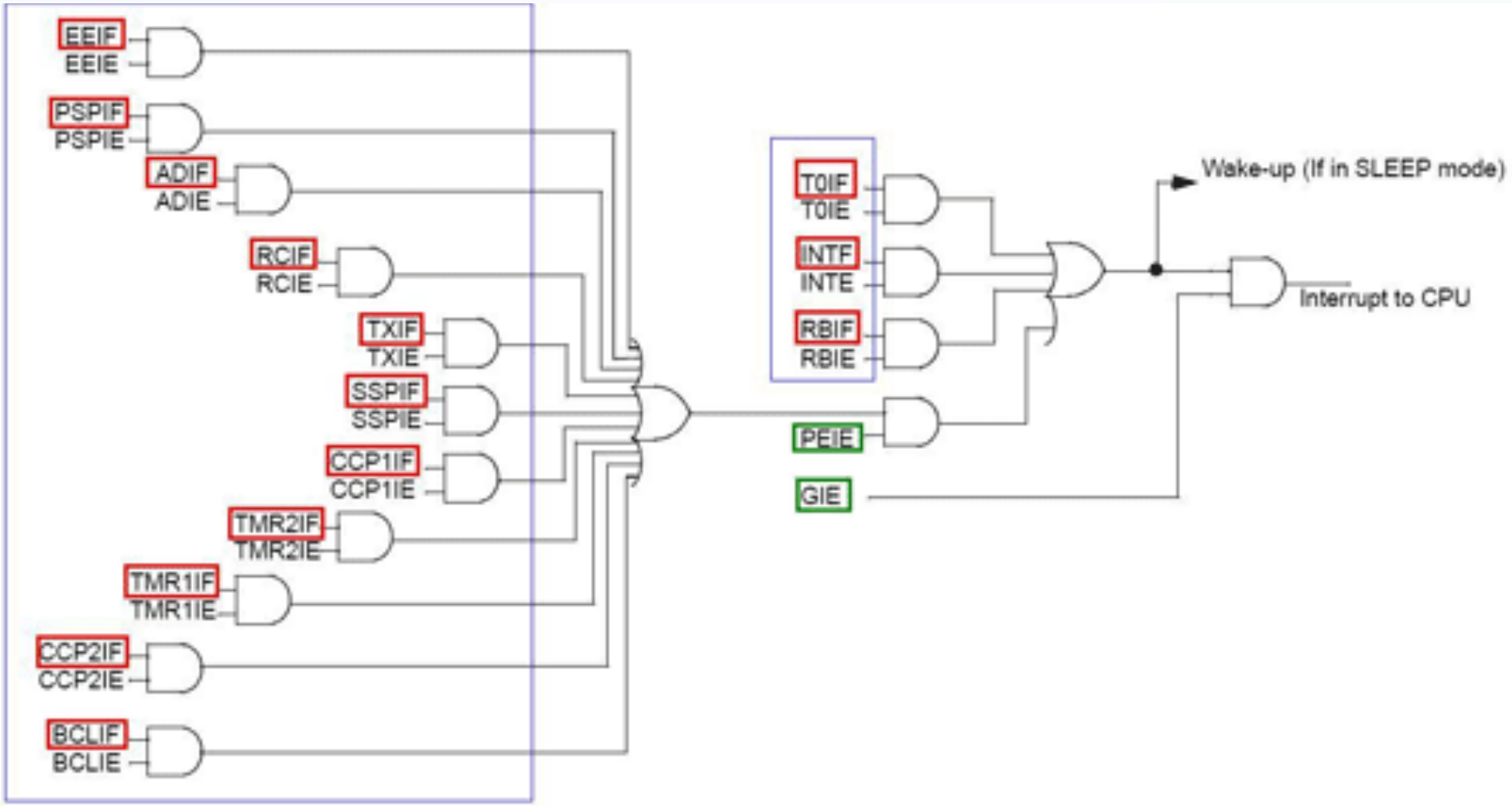
Pour qu'une interruption soit prise en compte, les 2 autorisations sont requises.



Cette routine peut se produire à n'importe quel moment du programme principal.

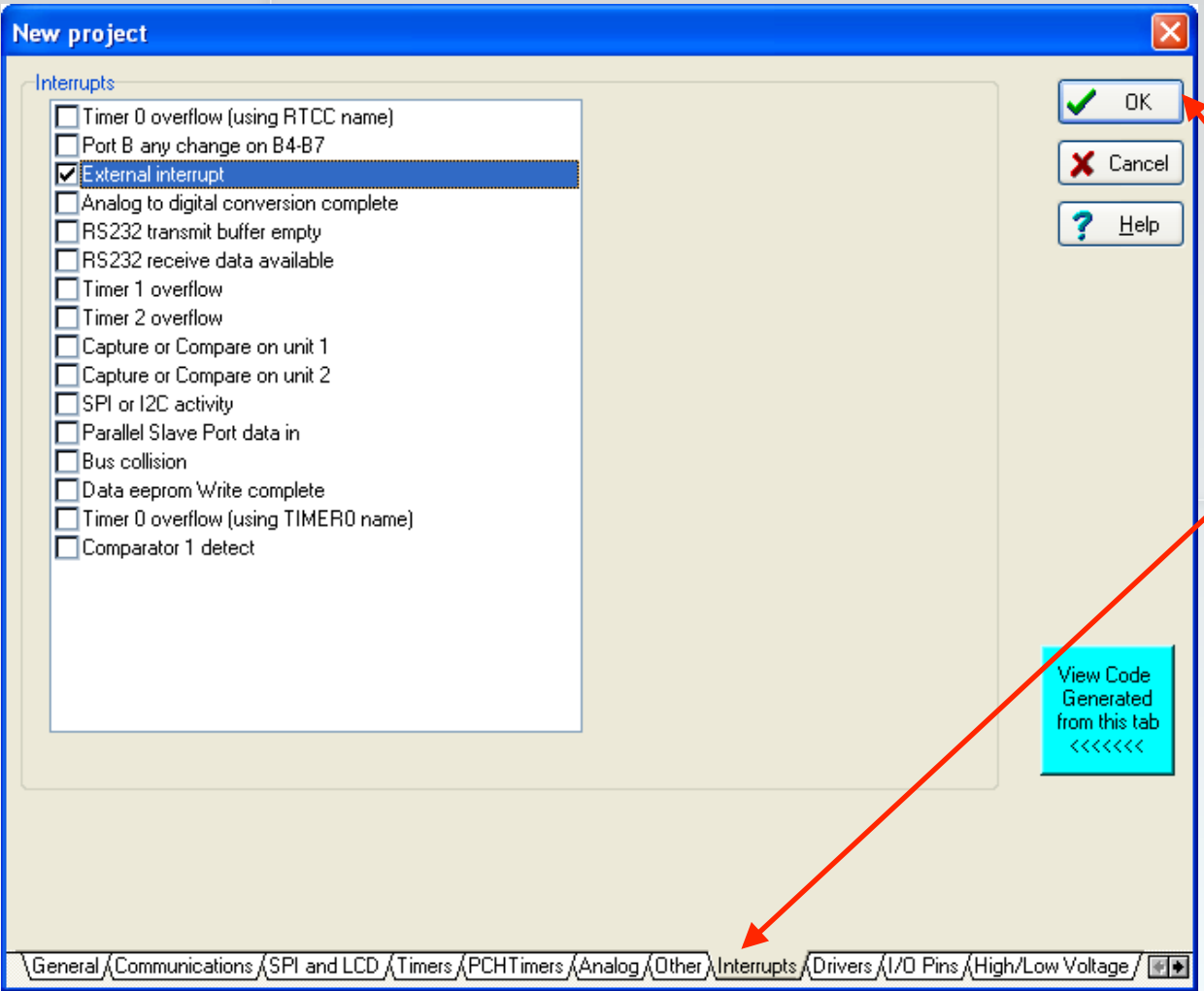
Si une interruption doit modifier le déroulement du programme principal, on fera en sorte que sa routine modifiera une variable globale qui sera testée dans le programme principal.

Les interruptions du PicC?



Les interruptions du PicC?

Exemple: On veut une interruption sur l'entrée INT/ RB_0 du µC:



Dans l'assistant, on va sur l'onglet « Interrupts »:

On coche « External interrupt ».

Si les autres onglets ont été renseignés, on peut cliquer sur « OK ».

Programme Interruption :

Inter1.h :

```
#include <16F877A.h>
#device ICD=TRUE
#device adc=10
#FUSES
NOWDT,HS,NOPUT,NOPROTECT,NODEBUG,NOBROWNOUT,NOLVP,NOCPDNOWRT
#use delay(clock=20000000)
#use rs232(baud=9600,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=8)
```

Inter1.c :

```
#include "inter1.h"

#int_EXT
EXT_isr()
{

}
void main()
{

. . .
   enable_interrupts(INT_EXT);
   enable_interrupts(GLOBAL);

}
```

Routine d' interruption externe précédée de la directive de compilation #int_ext

Programme principal

Validation de l' interruption externe

Validation globale des interruptions

Algorithme de la routine d'interruption et du programme principal …

```
#byte PORTB=0x06
#bit LED_ROUGE=PORTB.5
int16 compt;
#int_EXT
EXT_isr()
{
   compt=compt+1;
   LED_ROUGE = !LED_ROUGE;
}
void main()
{
. . .
   enable_interrupts(INT_EXT);
   enable_interrupts(GLOBAL);

   i=0;
   do
      {
      printf("valeur du compteur %lu  \r", compt);
      delay_ms(300);
      }
   while(TRUE);
}
```

Définition des ports et des bits de port (c'est différent du #define!)

Déclaration de la variable globale compt

Routine d'interruption: on incrémente compt et on inverse l'état de la LED rouge

Boucle infinie du programme principal qui se contente d'afficher la valeur de compt toutes les 1/3 secondes.