



# Démon : interruptions et servomoteur





# Interruptions : Qu'est-ce que c'est ?

Sur le PRisme :

- Internes ( Timers ,Communication,...)
- Externes (capteur, interrupteurs,...)



# Quel avantage ?



## Sans interruption (Polling) :

```
int main (void)
{
    if(detecteur != 0)
    {
        setSpeed(100,100);
        _delay_ms(1000);
        setSpeed(-50,50);
        _delay_ms(500);
    }
}
```



## Avec interruption :



```
int main (void)
{
    while(1)
    {
        setSpeed(100,100);
        _delay_ms(1000);
        setSpeed(-50,50);
        _delay_ms(400);
    }
}
```

```
void interrupt(void)
{
    digitalWrite(C,2,1);
    _delay_ms(1000);
    digitalWrite(C,2,0);
}
```



# Programmation



**#include** int\_ext.h

Cette ligne de code doit figurer  
au début de votre code pour  
pouvoir utiliser les fonctions  
d'interruptions

**Inclure int\_ext.h & int\_ext.c  
dans le dossier du projet**

Ces fichiers contiennent toutes  
les fonctions pour manier  
les interruptions

# Interruptions externe

Cette commande va activer l'interruption choisie et dire quelle fonction appeler en cas d'interruption

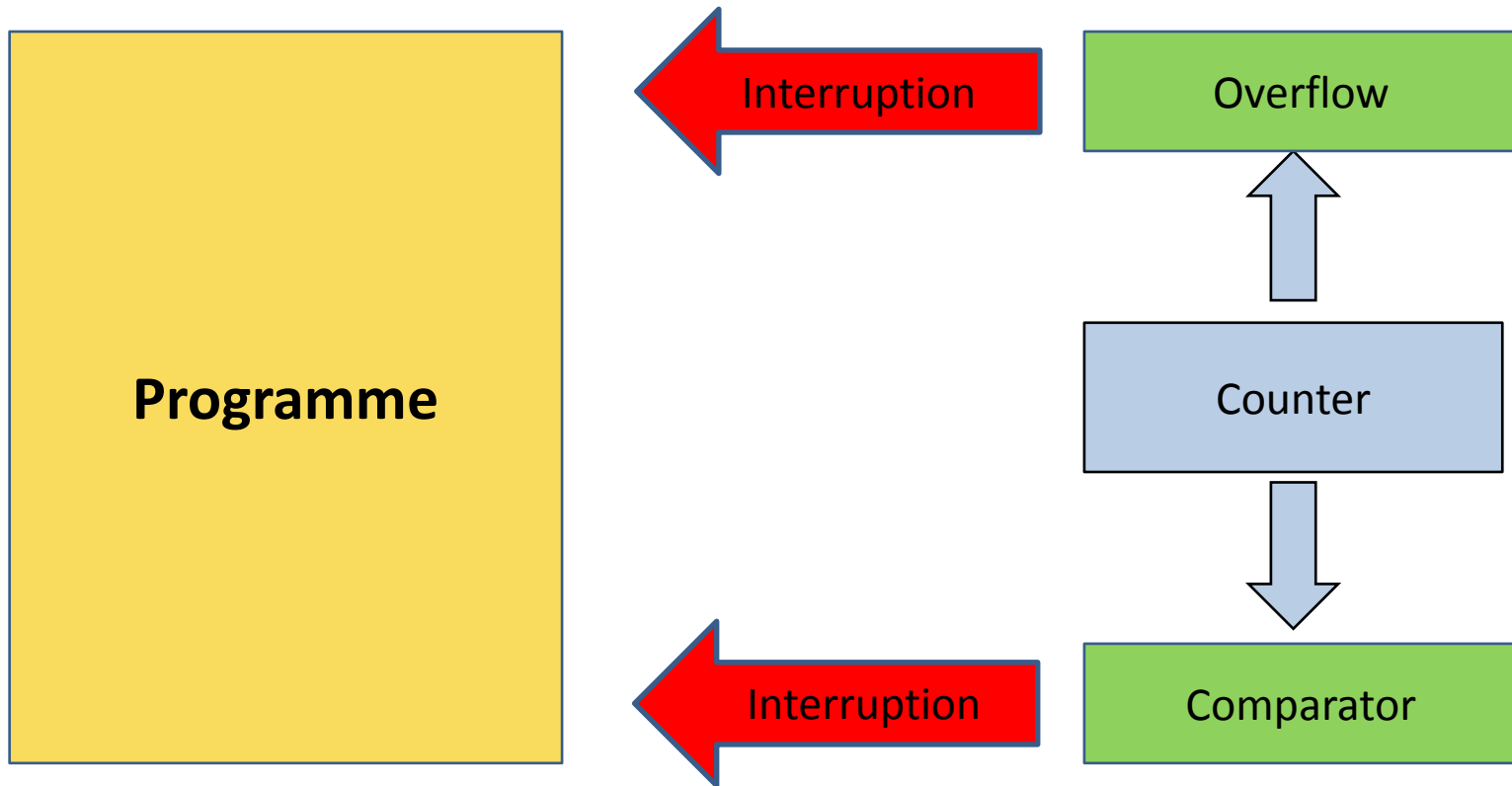
```
void configure_INT0(etat, function);
```

Voici les différents mode de déclenchement de l'interruption:



<b>RISING_EDGE</b>	Flanc montant : la valeur du capteur passe de 0 a 1
<b>FALLING_EDGE</b>	Flanc descendant : la valeur du capteur passe de 1 a 0
<b>ANY_CHANGE</b>	Flanc montant ou descendant
<b>OFF</b>	Éteint l'interruption

# Interruption internes : les timers



# Programmation



```
setCallback(fonction,durée,nombre_execution);
```

multiple de 1 ms

1->255

```
clearCallback(num_callback);
```

0->7







# Servomoteur

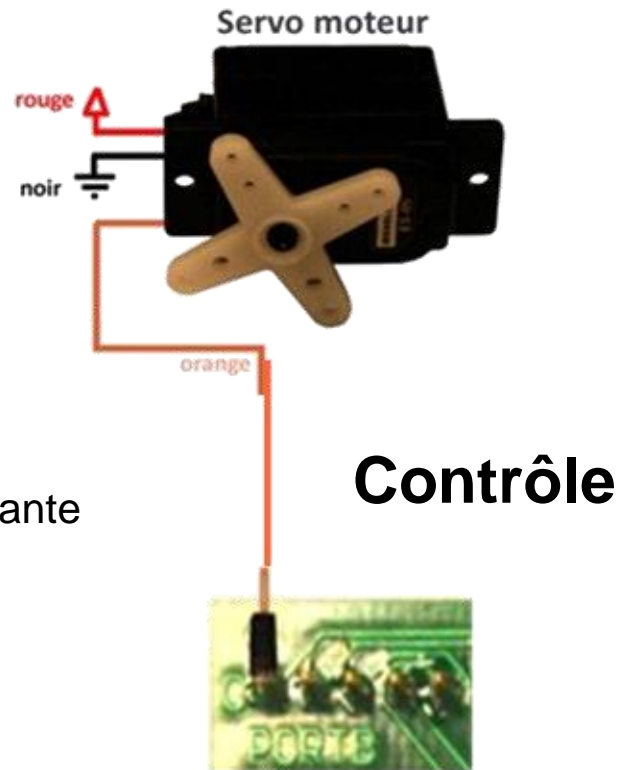


# Branchement



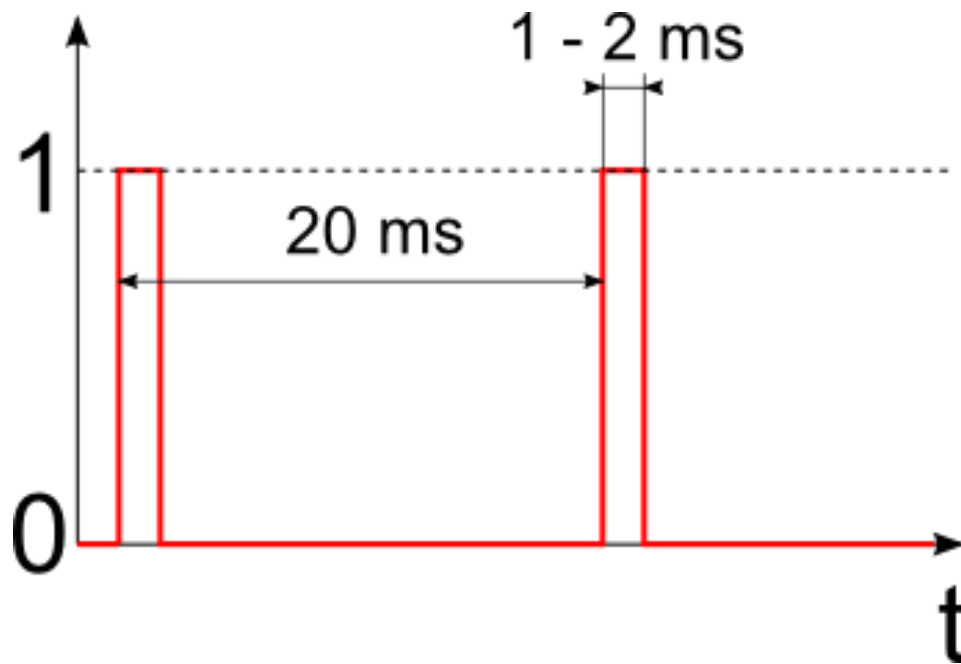
## Alimentation

Les servomoteurs peuvent être branchés sur le port B 0 à 7.  
Vous pouvez brancher jusqu'à 8 servomoteurs sur le Prisme  
Mais il faut une alimentation suffisante





# PWM



Signal :  $1\text{ms} = 0^\circ$   
 $2\text{ms} = 180^\circ$

Tant que le servomoteur reçoit ce signal,  
il va maintenir l'angle demandé

# Programmation



```
setServo( num_servo, angle_servo )
```

0 à 8

0 à 180°



# Exemple

