

LES INTERRUPTIONS EN ARDUINO

Par : Blaise Letakamba

INTRODUCTION

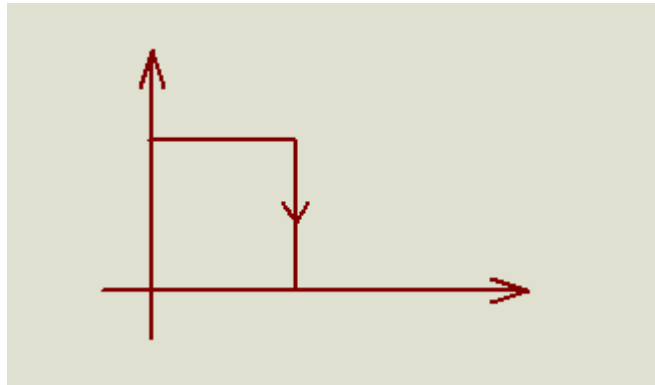
Imaginer que vous avez un téléphone entrain de suivre une vidéo puis on vous appelle la vidéo va d'abord s'arrêter à cause de l'appelle car pour un téléphone l'appelle est prioritaire. En effectuant l'appelle une fois l'appelle terminer la vidéo revient là où elle était pour continuer. En quelque sorte c'est ça l'interruption il s'agit d'une action prioritaire coupant le déroulement normal du programme pour s'exécuter puis le programme retourne à son déroulement normal.

On distingue deux sortes d'interruptions internes et externes dans ce présent travail nous allons parler sur les interruptions externes.

FORME LOGIQUE DE L'INTERRUPTION

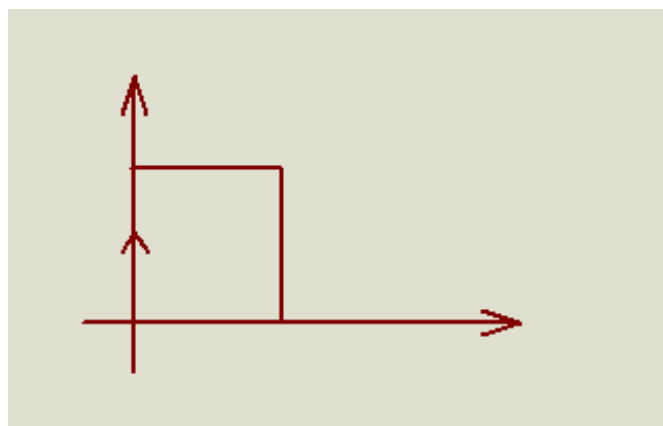
En ARDUINO il y a principalement quatre formes logiques d'interruptions

1. **FALLING** : il s'agit d'une interruption sur le front descendant sur les broches 3 ou 2.



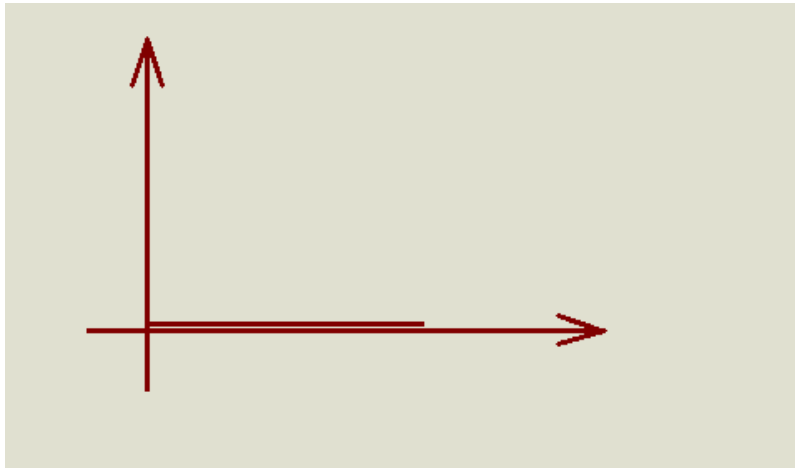
Dans ce cas il y a interruption si la broche passe de 1 à 0. Il s'agit d'une interruption externe sur front descendant.

2. **RISING** : il s'agit d'une interruption sur front montant sur les broches 3 ou 2



Dans ce cas il y a interruption si la broche passe de 0 à 1. Il s'agit d'une interruption externe sur front montant.

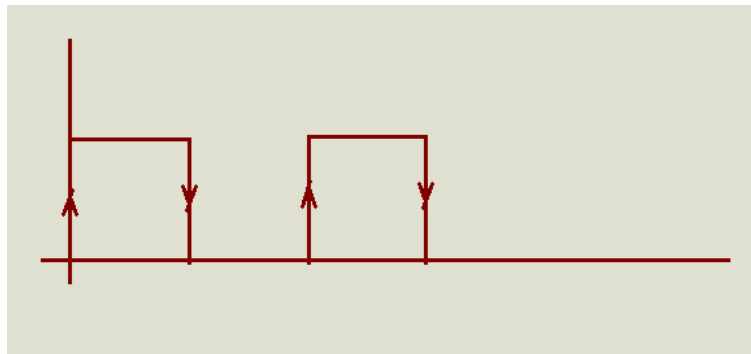
3. **LOW** : il s'agit d'une interruption qui s'effectue lorsque la broche est à l'état 0



Ici si la broche est à 0 cela est source d'interruption

4. CHANGE

Dans ce cas il y a interruption lorsqu'il y a un changement d'état de la broche de 0 à 1 ou de 1 à 0.



PROGRAMMATION DE L'INTERRUPTION EN ARDUINO

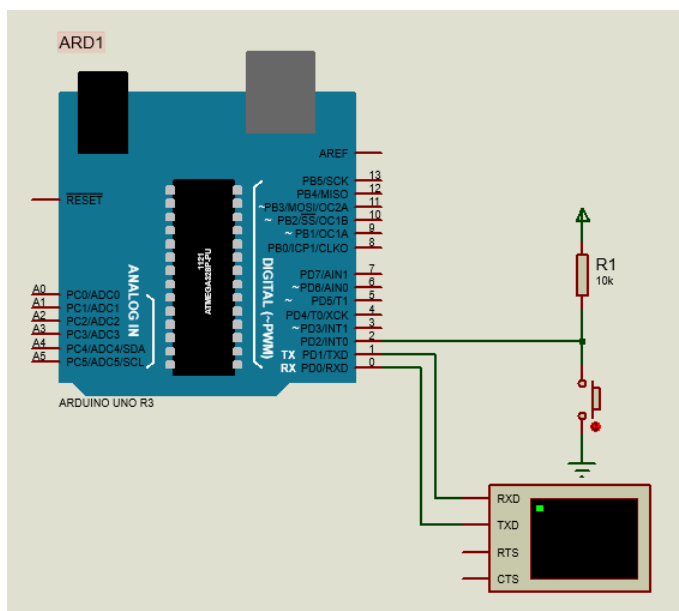
1. UTILISATION DE L'INTERRUPTION FALLING

Il est utile à signaler que cette interruption ne peut s'effectuer que sur la borne 3 et la borne 2 de la carte ARDUINO. On peut voir le code pour l'implémentation de cette interruption :

```

1  int compteur ;
2  void setup() {
3    Serial.begin(9600);
4    attachInterrupt(0,impulsioncompte,FALLING);
5  }
6
7  void loop() {
8    Serial.print("la valeur de compte est :");
9    Serial.println (compteur);
10 }
11 void impulsioncompte() {
12   compteur++;
13   if (compteur == 100){
14     compteur = 0;
15   }
16 }
17

```



```

4  attachInterrupt(0,impulsioncompte,FALLING);

```

La fonction `attachInterrupt` comprend trois paramètres

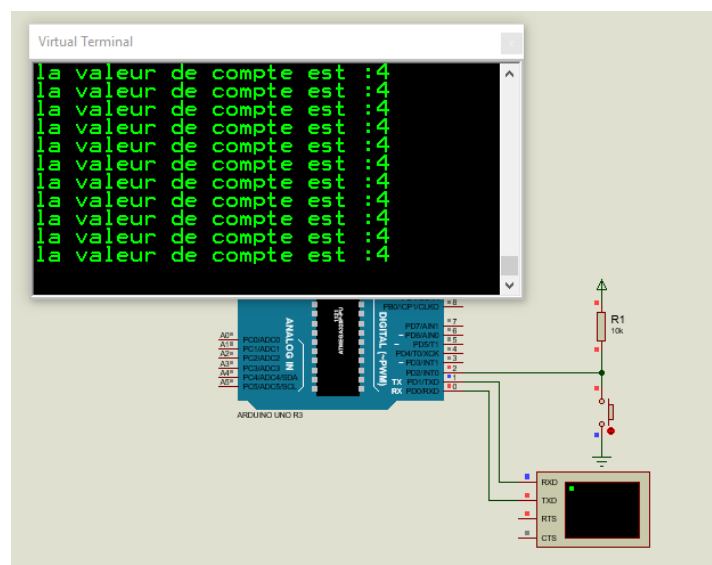
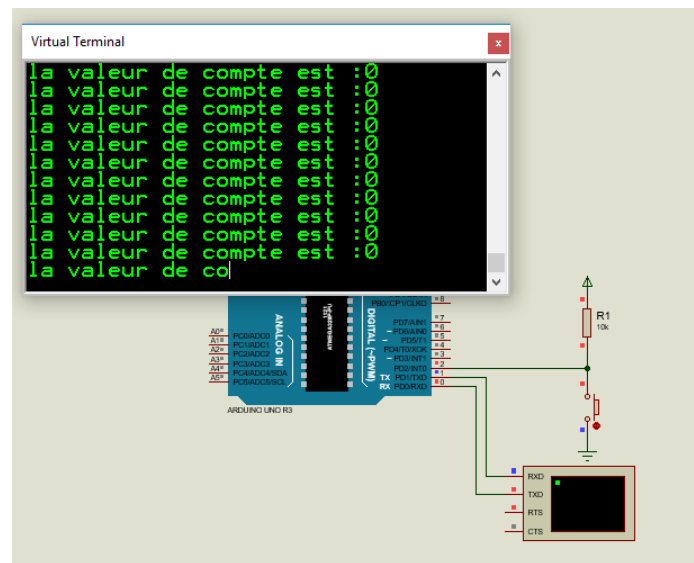
- ✚ Le premier paramètre détermine la broche de la carte arduino qui est sera objet de l'interruption
 - ✓ Si ce paramètre est égal à 0 donc l'interruption se fera sur la broche 2 de la carte ARDUINO.

- ✓ Si ce paramètre est égal à 1 donc l'interruption se fera sur la broche 3 de la carte ARDUINO.

✚ Le deuxième paramètre détermine la fonction qui pourra s'exécuter lorsqu'il y a interruption ici dans notre cas à chaque interruption la fonction qui s'exécute c'est la fonction `impulsioncompte()` cette fonction permet d'incrémenter la variable `compte` donc à chaque entrée dans l'interruption la variable `compte` est incrémentée.

✚ Le troisième paramètre détermine la nature de l'interruption par exemple ici dans notre cas il y aura interruption quand l'état de la broche 2 de la carte ARDUINO passe de 1 à 0.

IMAGE DE SIMULATION

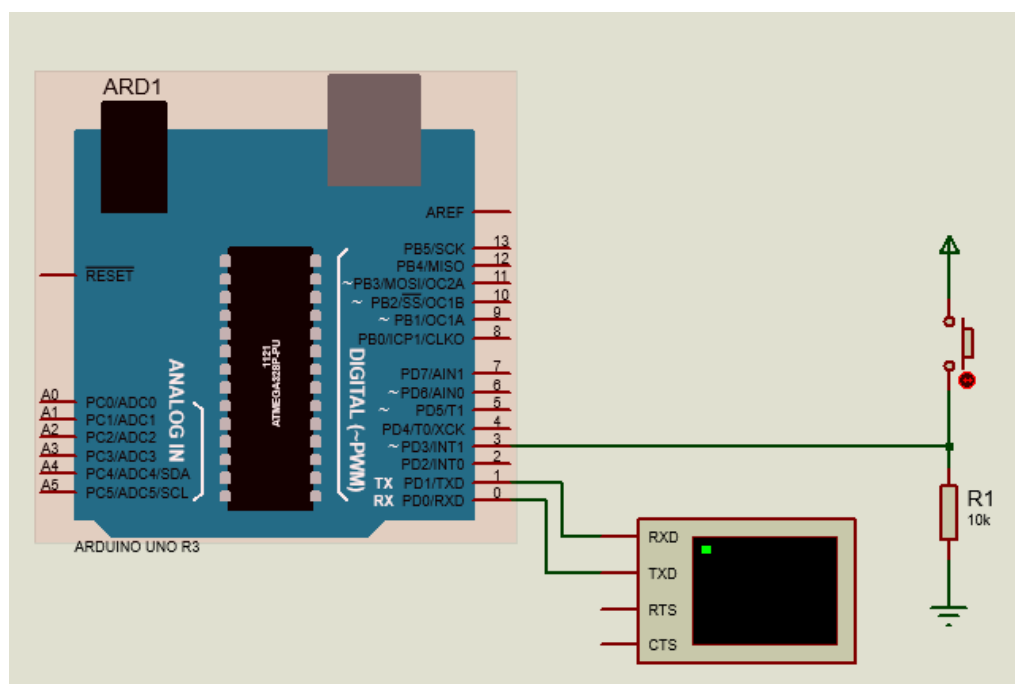


A chaque appuie sur le bouton la variable `compte` se voit incrémenter quand la valeur `compte` arrive à 100 elle retourne à 0.

Il est utile à signaler que l'interruption s'exécute juste pendant un temps. Donc quand il y a interruption le programme s'arrête pour aller exécuter la fonction d'interruption après cette exécution il retourne dans son déroulement normal.

UTILISATION DE L'INTERRUPTION RISING

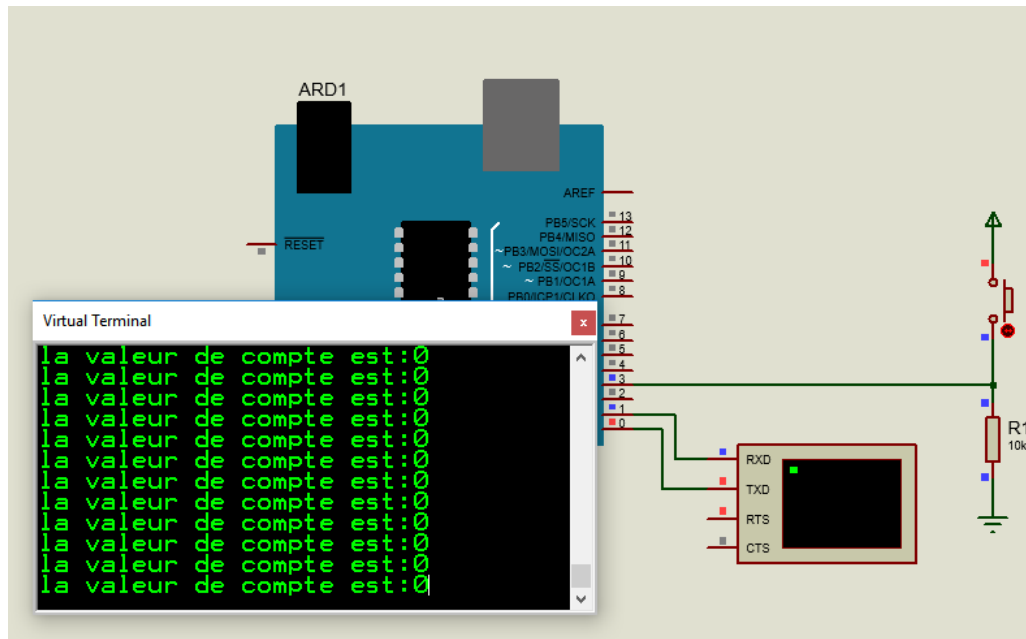
```
interruption $
1  int compteur ;
2  void setup() {
3    Serial.begin(9600);
4    attachInterrupt(1,impulsioncompte,RISING);
5  }
6
7  void loop() {
8    Serial.print("la valeur de compte est:");
9    Serial.println(compteur);
10 }
11 void impulsioncompte() {
12   compteur++;
13   if (compteur == 100){
14     compteur = 0;
15   }
16 }
17
18 |
```



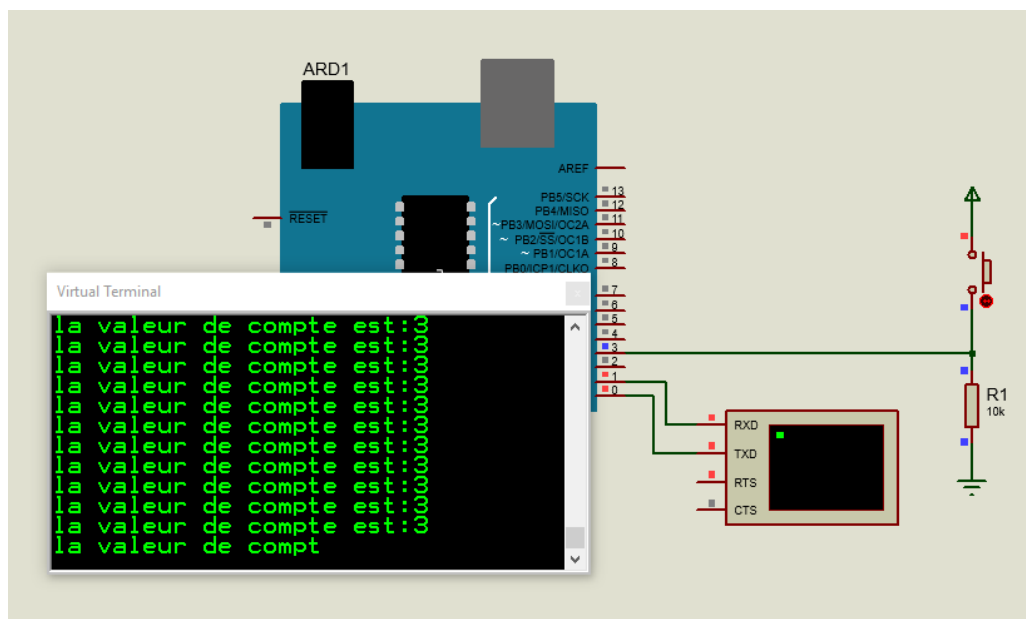
Ici l'interruption se passe sur la broche 3 de la carte ARDUINO. Il s'agit de l'interruption **RISING** dans ce cas l'interruption est déclenché lorsque l'entre passe de 0 à 1 ici il s'agit de l'interruption sur front montant.

RESULTAT DE SIMULATION

Résultant avant appuie sur le bouton poussoir :



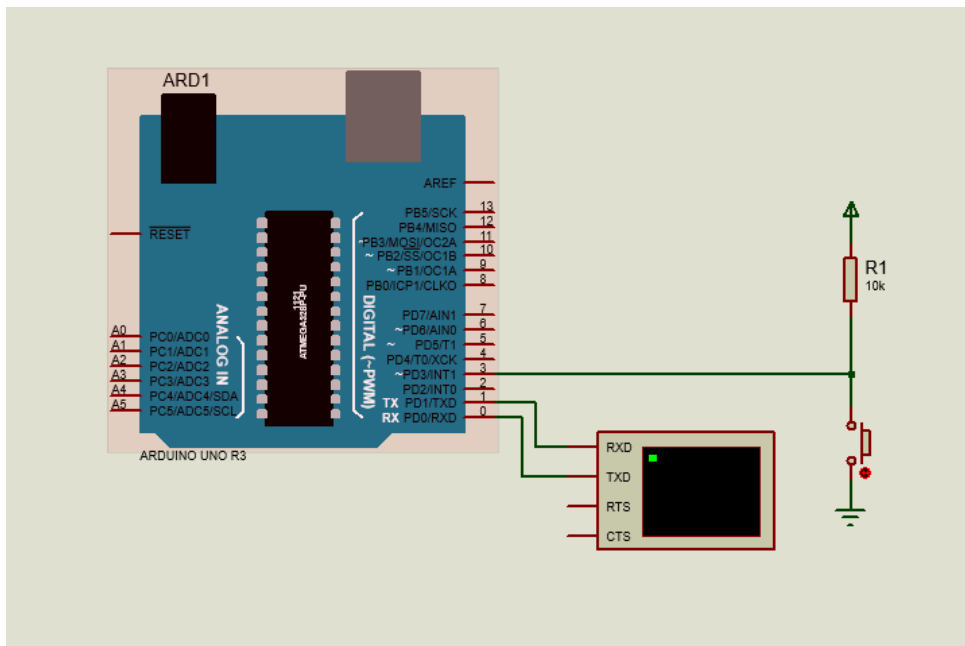
Résultat après trois appuies sur le bouton poussoir :



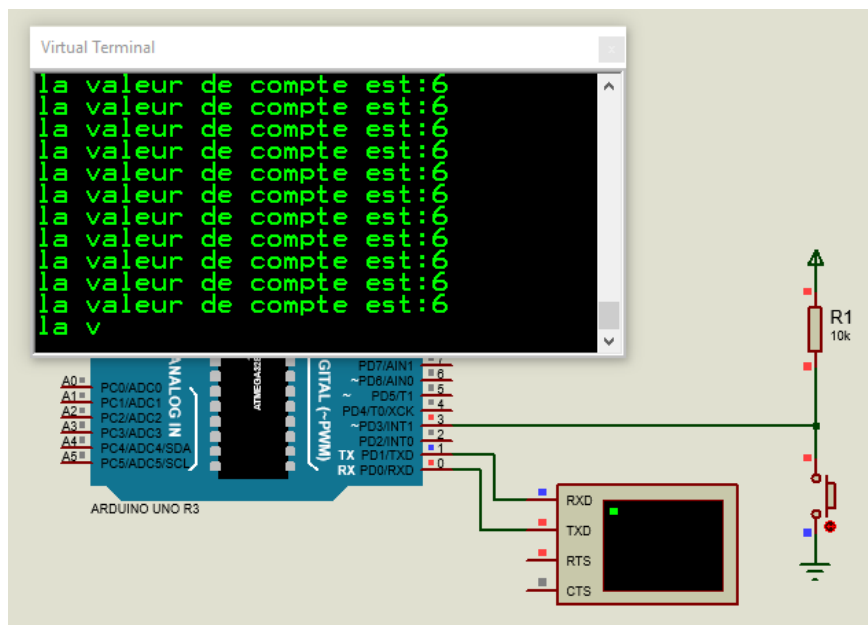
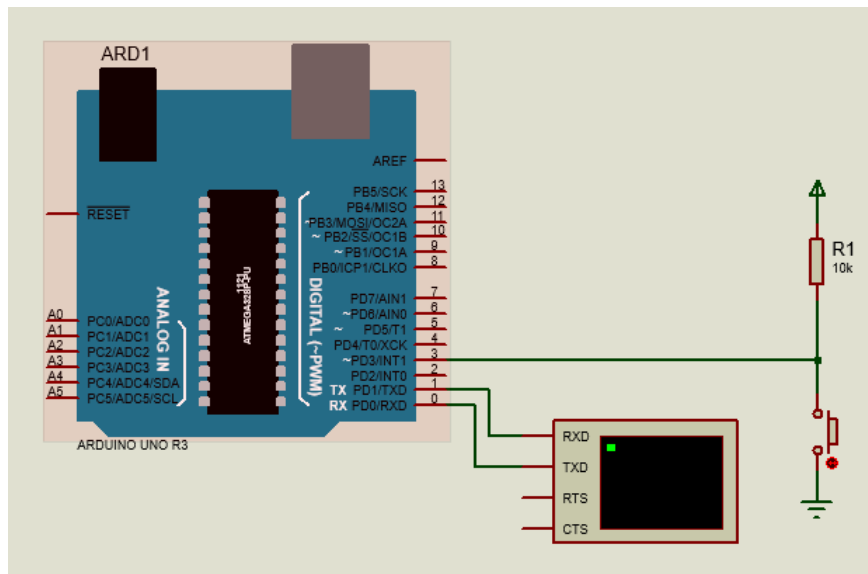
UTILISATION DE L'INTERRUPTION LOW

interruption §

```
1  int compteur ;
2  void setup() {
3    Serial.begin(9600);
4    attachInterrupt(1,impulsioncompte,LOW);
5  }
6
7  void loop() {
8    Serial.print("la valeur de compte est:");
9    Serial.println(compteur);
10 }
11 void impulsioncompte() {
12   compteur++;
13   if (compteur == 100){
14     compteur = 0;
15   }
16 }
17
18
```



Ici pour ce montage il y a interruption lorsque la borne 3 est à 0.



Ici il y a interruption lorsqu'il y a changement d'état sur la broche 3 la variable compte incrémente lorsqu'on appuie et lorsqu'on lâche le bouton poussoir donc à chaque changement d'état il y a interruption.

L'interruption est une notion trop importante en lisant cet article je crois vous avez capable d'appréhender cette notion.

Il est utile de noter que les interruptions sont des commandes prioritaires qui s'exécutent pendant une courte durée puis le programme revient dans son déroulement normal.