



Front montant d'un programme RAPE

Dans ce cours nous présentons le procédé utilisé pour les fronts montants utilisé dans le programme RAPE.

Table des matières

Front montant d'un programme RAPE	1
Utilité d'un front montant	1
Principe.....	1
Exemple utilisé dans le programme RAPE	1
Utilisation particulière du front montant sur une étape grafcet	2
Remarque	2

Utilité d'un front montant

Il signale un changement d'état d'un bit sur un tour de scrutation, de l'état bas à l'état haut.



Principe

On compare un état précédent (utilité du bit inter qui est l'état d'avant) avec l'état actuel.

Exemple : Lors d'un comptage d'impulsions nous devons prendre uniquement les fronts montants dans le cas ou nous prendrions l'état haut à chaque tour de scrutation, le comptage serait erroné.

Exemple utilisé dans le programme RAPE

```
void FM_BP(void)
```

```
{  
  //création du bit de front  
  if (bouton_poussoir & !inter_BP ) {fm_bit_BP = 1;} else {fm_bit_BP = 0;}  
  /* bit intermédiaire */  
  inter_BP= bit_BP;  
  return;
```



}

Description des différents tour de scrutation ;

Premier tour de scrutation bouton poussoir = 0 :

```
if (bouton_poussoir & !inter_BP ) {fm_bit_BP = 1;} else {fm_bit_BP = 0;}  
le bouton poussoir est à zéro donc fm_bit_BP = 0;  
le bit « inter_BP » est aussi à 0.
```

Deuxième tour de scrutation avec bouton poussoir = 1

```
le bit « inter_BP » est toujours à 0.  
if (bouton_poussoir & !inter_BP ) {fm_bit_BP = 1; else {fm_bit_BP = 0;}  
le bouton poussoir est à un et le bit inter_BP=0 donc fm_bit_BP = 1;
```

Troisième tour de scrutation

```
le bit « inter_BP » est à 1 et le bouton poussoir est à 1 ou 0(relâché ou non).  
if (bouton_poussoir & !inter_BP ) {fm_bit_BP = 1; else {fm_bit_BP = 0;}  
le bit inter_BP=1 donc fm_bit_BP = 0;
```

Il y aura un bit de front montant qui passera à 1 sur un seul tour de scrutation.

Dans quel cas utiliserons nous les fronts montants dans le programme :

Pour les comptage des tempos par exemple.

Utilisation particulière du front montant sur une étape grafcet

Dans ce cas nous avons déjà le bit de l'état d'avant qui est

copie_etapes_graf_cycle = etapes_graf_cycle ; placé en fin de grafcet remplace le bit inter

exemple utilisé dans le programme test_cycle_aléatoire

```
if (copie_etapes_graf_cycle == 7 && etapes_graf_cycle == 8 ) { chgt_sens_rotation = ! chgt_sens_rotation; } //inversion à chaque passage sur ce fm
```

explications : a chaque fois que l'étape 8 est active mais que la *copie_etape_graf_cycle = 7* (donc étape précédente) on a sur un seul tour de scrutation cette situation et on fait un changement de l'état du bit : *chgt_sens_rotation*.

Remarque sur *chgt_sens_rotation = ! chgt_sens_rotation* :

L'inverse de *Chgt_sens_rotation* est écrit dans *chgt_sens_rotation*.



Ne pas confondre avec le symbole d'inégalité qui est : « \neq » donc dans l'autre sens.