



Premiers déplacements du robot RAPE

Avance et retour RAPE

Les mouvements : avance jusqu'à bloquer sur un objet.

Table des matières

Premiers déplacements du robot RAPE	1
Avance et retour RAPE	1
Introduction :	1
Analyse du programme :	2
explications sur la ligne du printf :	4

Introduction :

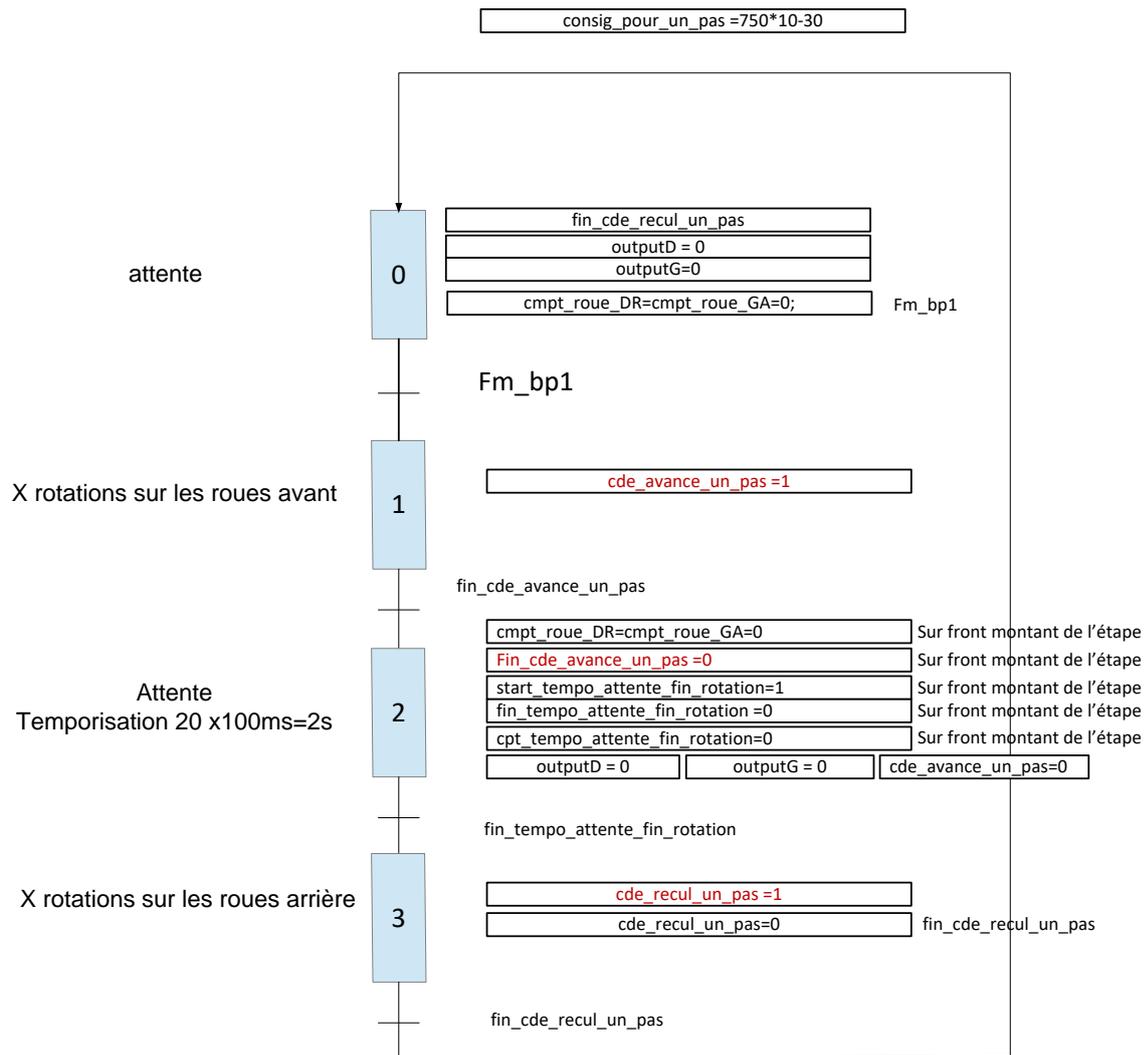
Dans ce cours nous allons faire des mouvements rectilignes à notre robot puis un retour

Programme : test_cycle_roue_avec_ralenti + retour_P5

C'est le même programme que le programme précédent mais on des étapes grafcet supplémentaires.



grafcet



Nous remarquons deux nouveautés :

Une temporisation a été ajoutée, des fronts montants sur étape grafcet.

Analyse du programme :

Modification des paramètres :

```
consig_pour_un_pas= 750*4-10;//donc 4 tours - 10points d'inertie
```

```
//Transition E0 a E1
```

```
if (copie_etapes_graf_cycle == 0 & fm_bp1)
```

```
{etapes_graf_cycle = 1; cmpt_roue_DR = cmpt_roue_GA = 0; //un tour de scrutation
```

```
//ETAPE1 avance
```

```
if (etapes_graf_cycle == 1) {cde_avance_un_pas = 1;} //en continu pendant tout le déplacement
```

**// Transition de E1 à E2**

```
if ((copie_etapes_graf_cycle == 1) && fin_cde_avance_un_pas) {etapes_graf_cycle = 2;cde_avance_un_pas=0;outputD=0;outputG=0;}
```

//front montant de l'étape

```
if (etapes_graf_cycle ==2 & copie_etapes_graf_cycle != 2) // sur un seul tour de scrutation
```

```
{  
  cmpt_roue_DR=cmpt_roue_GA=0;  
  start_tempo_attente_fin_rotation = 1;  
  fin_cde_avance_un_pas =0;  
  fin_tempo_attente_fin_rotation=0;  
  cpt_tempo_attente_fin_rotation=0;  
}
```

Dans cette ligne on prend l'étape actuelle et si elle est différente de sa copie en fin de programme donc sur un tour de scrutation, on réalise toutes les actions qui sont entre les accolades.

//etape 2 temporisation

```
//tempo
```

```
if (start_tempo_attente_fin_rotation & fm_bit_100ms) {cpt_tempo_attente_fin_rotation++;} //à chaque front montant le compteur est incrémenté  
if (cpt_tempo_attente_fin_rotation==20)  
{cpt_tempo_attente_fin_rotation=0;fin_tempo_attente_fin_rotation=1;start_tempo_attente_fin_rotation=0;}
```

La temporisation est sur une base de temps de 100ms, le compteur tempo_attente_fin_rotation est incrémenté sur chaque front montant de la base de temps. Arrivée à la valeur de 20, la tempo est terminée, le bit fin de tempo est mis à 1 et la tempo initialisée.

// Transition de E2 à E3

```
if (copie_etapes_graf_cycle ==2 & fin_tempo_attente_fin_rotation ) {etapes_graf_cycle = 3;}
```

//ETAPE3 recul

```
if (etapes_graf_cycle == 3 ) {cde_recul_un_pas=1;} //en continu pendant tout le déplacement
```

// Transition de E2 à E0

```
if ((copie_etapes_graf_cycle == 3) && fin_cde_recul_un_pas) {etapes_graf_cycle = 0;cde_recul_un_pas=0;}
```

//etape 0

```
if (etapes_graf_cycle == 0) {fin_cde_recul_un_pas =0;outputD=0;outputG=0;}
```

```
//-----passage de l'étape dans une copie pour que le grafctet ne puisse pas faire une rotation sans reboucler à chaque étape-----
```

```
copie_etapes_graf_cycle = etapes_graf_cycle;
```

```
mot_etape_gra.byte = etapes_graf_cycle;
```



explications sur la ligne du printf :

|| : signifie un OU.

&& : signifie une ET.

Si on a le fm_tempo et pas de mouvement il y a l'affichage du printf.

Si on a besoin d'avoir un printf malgré que l'on soit en mouvement alors on appui sur le BP2.

```
if (bp2 || ( fm_bit_500ms && !en_mouvement ) )
{
    printf("\r\n");
    printf("  cmpt_roue_GA: %d ", cmpt_roue_GA);
    printf("  cmpt_roue_DR: %d ", cmpt_roue_DR);
    printf("  ecart roue : %d ", ecart_roue);
    printf("  etapes_graf_cycle: %d ", etapes_graf_cycle);
    printf("  cpt_tempo_attente_fin_rotation: %d ", cpt_tempo_attente_fin_rotation);
    printf("\r\n");
}
```

Traduction de la ligne : L'appui sur le BP2 ou le fm_bit_500ms, si on est pas en mouvement (ce bit est à un si un moteur est commandé) alors on fait un affichage.

Le but de cette commande : pendant que le robot est en mouvement il est inutile de faire de l'écriture d'autant que la ressource utilisée pour le printf est très importante et modifie les arrêts de moteurs !!! donc on attend d'être à l'arrêt pour faire des impressions. Mais il se peut que l'on soit bloqué dans une étape grafcet avec une commande des moteurs dans ce cas l'analyse est facilitée par la lecture des impression donc une pression sur le bp2 nous permettra d'avoir accès aux informations d'impression.