



Premiers déplacements du robot RAPE

Dans ce cours nous allons faire nos premiers mouvements sur le robot. Nous allons introduire la notion de GRAFCET.

Les mouvements : avance d'un certain nombre de pas puis arrêt.

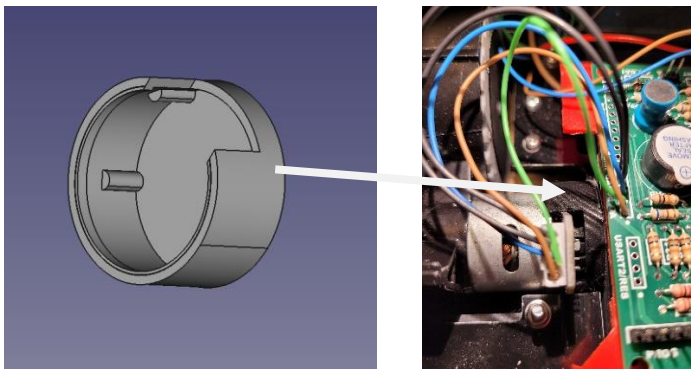
Nous vérifierons les positions, la perte ou les écarts de points entre droite et gauche.

Table des matières

Premiers déplacements du robot RAPE	1
Principe :	2
Traduction sous forme de ligne de C.....	2
test d'arrêt	4
Pour faire des changements sur la distance parcourue :	4
Le réglage des vitesse :	4
Premiers essais sans régulation :	5
Mise en service de la régulation :	6
Explications sur le programme :	8

Nom du programme choisi : **test_cycle_roue_continu_P3.c**

Attention : Il est impératif de mettre un capuchon sur le codeur. En effet les diodes sont sensibles à la lumière extérieure et cela peut perturber le comptage.





11/2023

Dans ce petit programme nous allons aussi introduire la notion de cycle et de grafcet à 2 étapes.

Pour en savoir plus sur le grafcet voici un lien :

http://philippe.berger2.free.fr/automatique/cours/G7/le_grafcet.htm

Principe du grafcet :

Dans l'étape 0

- les compteurs de roue sont mis à 0.
- La consigne est copiée.
- Les consignes sur les sorties sont mises à 0.

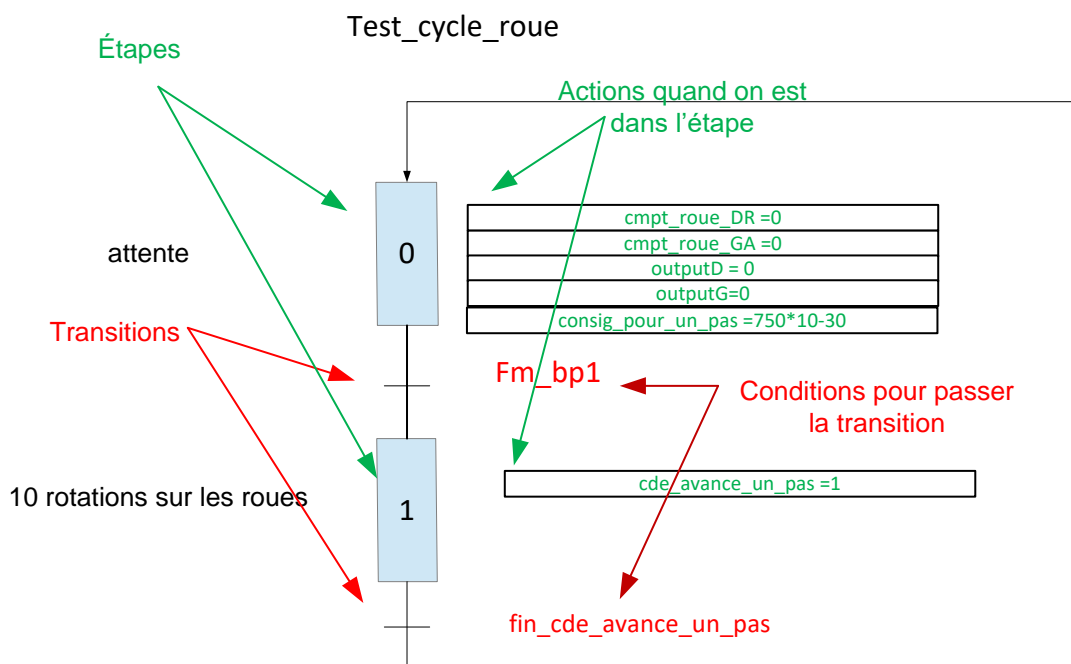
Transition de 0 à 1 Lors de l'appui sur le BP1 (gros bouton rouge).

Dans l'étape 1

- Commande de l'avance du robot

Transition de l'étape 1 à 0 lorsque le comptage est arrivé à la valeur de consigne fait 10 tours de roue et on supprime l'inertie d'arrêt (30pts)

Retour à l'étape 0.



Traduction sous forme de ligne de C.

```
void cycle ()
```

```
{
```

```
    // grafcet Test_cycle_roue
```



//Transition E0 a E1

```
if (copie_etapes_graf_cycle ==0 && fm_bp1)
    {consig_pour_un_pas= 750*2-10; (chargement de la valeur de consigne)
    etapes_graf_cycle =1; (mise à 1 de l'étape grafcet)
    cmpt_roue_DR=cmpt_roue_GA=0;}// (mise à 0 des compteurs)
```

- $750*2-10 = > 750$ est le nombre de points sur un tour de roue
- 2 est le nombre de tour de roue
- 10 est l'inertie du moteur

//ETAPE1 avance

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

// Transition de E1 à E0

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

//ETAPE 0 attente

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

petit mécanisme pour ne faire qu'une étape par tour de scrutation.

```
    copie_etapes_graf_cycle = etapes_graf_cycle;
}
```

Commande des moteurs et régulation de la roue gauche et de la roue droite.

chgt des consignes sur front montant de la commande du mouvement

```
    if (fm_cde_mvt_avance) {setpointG= setpointD = consig_pour_un_pas;}//
```

on découple la commande d'avance en deux commandes : droite et gauche

si la valeur des compteurs est inférieure à la consigne alors on commande le moteur sinon on arrête le moteur

```
    if (cde_avance_un_pas && (cmpt_roue_DR < consig_pour_un_pas) )
    {cde_avance_un_pas_DR=1;} else {cde_avance_un_pas_DR=0;}//
```

```
    if (cde_avance_un_pas && (cmpt_roue_GA < consig_pour_un_pas) )
    {cde_avance_un_pas_GA=1;} else {cde_avance_un_pas_GA=0;}
```

quand les deux moteurs ont atteint la consigne on coupe la commande d'avance. C'est plus rapide que d'attendre que l'étape passe en 0.

```
    if (cde_avance_un_pas && (((cmpt_roue_GA >= consig_pour_un_pas) && (cmpt_roue_DR
    >= consig_pour_un_pas))) ) {cde_avance_un_pas=0;fin_cde_avance_un_pas=1;}//
```



test d'arrêt

Donc les roues font 10 tours (faire un tracé sur un scotch dans une position quelconque car le programme met les valeurs à 0 avant de commencer sa rotation) puis vous pouvez compter le nombre de tours de roue effectué et vérifier que la roue s'arrête bien sur le trait.

Pour faire des changements sur la distance parcourue :

Il faut modifier la consigne : *consig_pour_un_pas= 750*2-10;*

Rappel : //un tour de roue = 500pts distance = 526mm de déplacement

//750= nb point sur un tour de roue pour avancer un pas

// chaque tour de roue est le chiffre multiplicateur de pas

// -10 correspond à l'inertie du robot

Remarque nous aurions pu mettre aussi la valeur de 1490 ($750*2-10$)

Le réglage des vitesses :

vitesse_min=180;

vitesse_max = 312;

vitesse_moyenne = 260; étant la vitesse qui nous importe dans cet essai



Premiers essais sans régulation :

Voici le programme : mettre en commentaire les lignes ci-dessous

```
void gestion_moteur()
{
    //*****GESTION moteur*****
    //cde avance
    if (fm_cde_mvt_avance) {setpointG= setpointD = consig_pour_un_pas;} //chgt des consignes
    if (cde_avance_un_pas && (cmt_roue_DR < consig_pour_un_pas) ) {cde_avance_un_pas_DR=1;} else {cde_avance_un_pas_DR=0;} //
    if (cde_avance_un_pas && (cmt_roue_GA < consig_pour_un_pas) ) {cde_avance_un_pas_GA=1;} else {cde_avance_un_pas_GA=0;}
    if (cde_avance_un_pas && (((cmt_roue_GA >= consig_pour_un_pas) && (cmt_roue_DR >= consig_pour_un_pas) )) {cde_avance_un_pas=0;fin_cde_avance_un

    //*****REGULATION DE LA VITESSE DES ROUE *****
    //permet au robot d'avancer droit
    // gestion moteur

    vitesse_min_GA=200;
    vitesse_min_DR=180; // min ne pas modifier
    vitesse_max_GA = 310; // max ne pas modifier
    vitesse_max_DR = 290;
    vitesse_moyenne_GA =260; // @2
    vitesse_moyenne_DR = 260; //260 @2

    vitesse_demandee_GA = vitesse_moyenne_GA ;
    vitesse_demandee_DR = vitesse_moyenne_DR;

/*
    ecart_roue = cmt_roue_DR - cmt_roue_GA ; // negatif ga G>D gauche ralentit rapide // positif D<G droite ralentit
    //setpoint est le nombre de points visé

    //arrêt en vitesse lente
    if (setpointG - 200 <= cmt_roue_GA) {vitesse_demandee_GA = vitesse_min_GA-10;} //
    if (setpointD - 200 <= cmt_roue_DR) {vitesse_demandee_DR = vitesse_min_DR;}

    // entre les petites vitesses de démarrage et le ralentissement d'arrivée
    if (setpointG - 200 > cmt_roue_GA & cmt_roue_GA > 200) {vitesse_demandee_GA = vitesse_moyenne_GA;} //pour arrêter les moteurs quand on arrive en posit
    if (setpointD - 200 > cmt_roue_DR & cmt_roue_DR > 200) {vitesse_demandee_DR = vitesse_moyenne_DR;}

    //démarrage en vitesse lente
    if ( cmt_roue_GA < 200) {vitesse_demandee_GA = vitesse_min_GA-10;} //pour avoir un démarrage lent
    if ( cmt_roue_DR < 200) {vitesse_demandee_DR = vitesse_min_DR;}

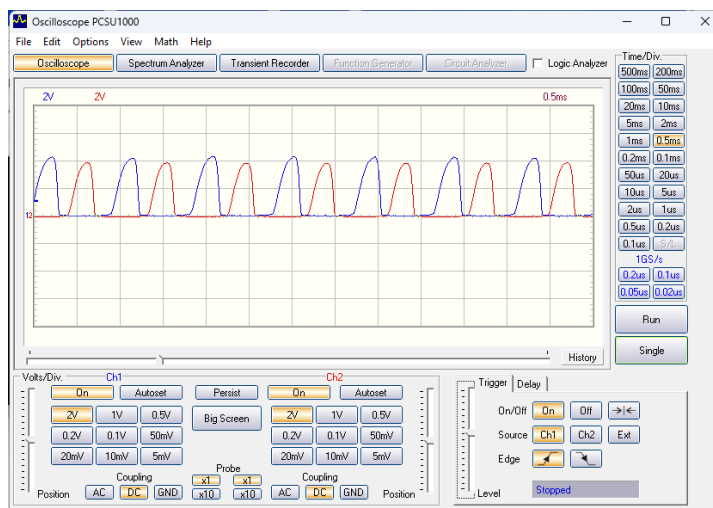
    if (vitesse_demandee_GA == vitesse_min_GA) {sortie_led_bleue=1;} else {sortie_led_bleue=0;}

    //régulation lors du déplacement pour avoir les roues en phase
    if (ecart_roue < 0) {vitesse_demandee_GA = 0; } //setpointG est la valeur à atteindre par le codeur
    if (ecart_roue > 0) {vitesse_demandee_DR = 0;}

    //arrêt total quand on arrive en position
    if (setpointG <= cmt_roue_GA) {outputG = 0;vitesse_demandee_GA=0; } //pour arrêter les moteurs quand on arrive en position
    if (setpointD <= cmt_roue_DR) {outputD = 0;vitesse_demandee_DR=0;}

*/
    outputG = vitesse_demandee_GA;
    outputD = vitesse_demandee_DR;
}
```

on se rend vite compte que si l'on pilote les roues avec la même consigne que le robot avance en arc de cercle.(cela dépend bien sûr des moteurs).





Voir la première vidéo « sans régulation »

Mise en service de la régulation :

Donc nous allons vérifier le fonctionnement de la régulation (simple) en mettant en commentaire la régulation des moteurs.

```
vitesse_demandee_GA = vitesse_moyenne_GA ;
```

```
vitesse_demandee_DR = vitesse_moyenne_DR;
```

en supprimant les commentaires

```
/*
```

```
.....
```

```
..... */
```

sur la régulation voici ce que nous obtenons :

```

//REGULATION DE LA VITESSE DES ROUE
//permet au robot d'avancer droit
// gestion moteur
vitesse_min_GA=200;
vitesse_min_DR=180;// min ne pas modifier
vitesse_max_GA = 310; // max ne pas modifier
vitesse_max_DR = 290;
vitesse_moyenne_GA =260; // @2
vitesse_moyenne_DR = 260;//260 @2

// vitesse demandee GA = vitesse moyenne_GA ;
// vitesse demandee DR = vitesse moyenne_DR;

ecart_roue = cmpt_roue_DR - cmpt_roue_GA ;;// negatif ga C>D gauche ralentit rapide // positif D<G droite ralentit
//setpoint est le nombre de points visé

//arrêt en vitesse lente
if (setpointG - 200 <= cmpt_roue_GA) {vitesse_demandee_GA = vitesse_min_GA-10;}//
if (setpointD - 200 <= cmpt_roue_DR) {vitesse_demandee_DR = vitesse_min_DR;}

// entre les petites vitesses de démarrage et le ralentissement d'arrivée
if (setpointG - 200 > cmpt_roue_GA & cmpt_roue_GA > 200 ) {vitesse_demandee_GA = vitesse_moyenne_GA;}//pour arrêter les moteurs quand on arrive en pos
if (setpointD - 200 > cmpt_roue_DR & cmpt_roue_DR > 200) {vitesse_demandee_DR = vitesse_moyenne_DR;}

//démarrage en vitesse lente
if ( cmpt_roue_GA < 200) {vitesse_demandee_GA = vitesse_min_GA-10;}//pour avoir un démarrage lent
if ( cmpt_roue_DR < 200) {vitesse_demandee_DR = vitesse_min_DR;}

if (vitesse_demandee_GA == vitesse_min_GA ) {sortie_led_bleue=1;} else {sortie_led_bleue=0;}

//régulation lors du déplacement pour avoir les roues en phase
if (ecart_roue < 0 ) {vitesse_demandee_GA = 0; } //setpointG est la valeur à atteindre par le codeur
if (ecart_roue > 0 ) {vitesse_demandee_DR = 0;};

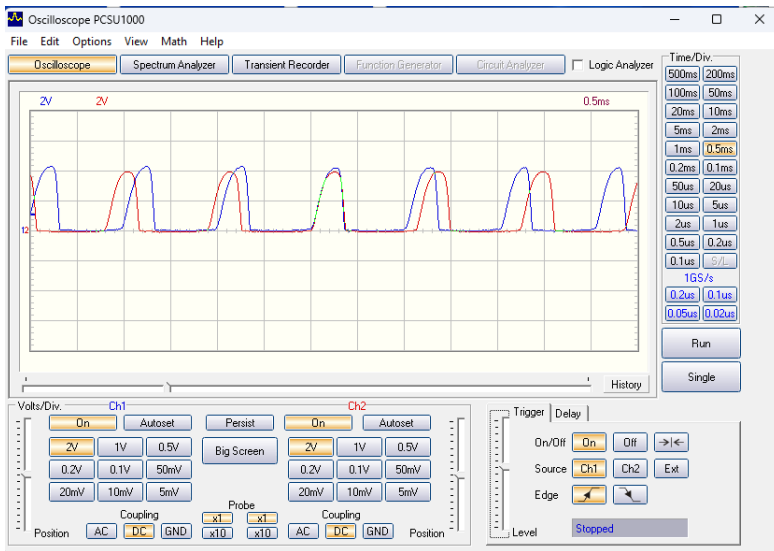
//arrêt total quand on arrive en position
if (setpointG <= cmpt_roue_GA) {outputG = 0;vitesse_demandee_GA=0; }//pour arrêter les moteurs quand on arrive en position
if (setpointD <= cmpt_roue_DR) {outputD = 0;vitesse_demandee_DR=0;};

outputG = vitesse_demandee_GA;
outputD = vitesse_demandee_DR;

```

Nouveau test :

Le robot démarre lentement puis avance en ligne droite et ralentit avant de s'arrêter.



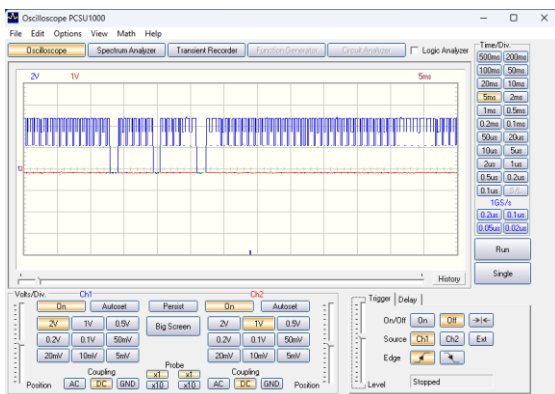
```
// gestion moteur

vitesse_min=180;
vitesse_max = 310;
vitesse_moyenne = 260;
ecart_roue = cmpt_roue_DR - cmpt_roue_GA ;// negatif ga G>D gauche ralentit rapide // positif D<G droite ralentit
//setpoint est le nombre de points vise
if (setpointG - 30 > cmpt_roue_GA) {vitesse_demandee = vitesse_moyenne;}//pour arrêter les moteurs quand on arrive en position
if (setpointD - 30 > cmpt_roue_DR) {vitesse_demandee = vitesse_moyenne;}

if (setpointG - 30 <= cmpt_roue_GA) {vitesse_demandee = vitesse_min;}//
if (setpointD - 30 <= cmpt_roue_DR) {vitesse_demandee = vitesse_min;}

if (ecart_roue < 0) {outputG = 0; outputD = vitesse_demandee;} //setpointG est la valeur à atteindre par le codeur
if (ecart_roue > 0) {outputG = vitesse_demandee;outputD = 0;}
if (ecart_roue == 0) {outputG = outputD = vitesse_demandee;}

if (setpointG <= cmpt_roue_GA) {outputG = 0;}//pour arrêter les moteurs quand on arrive en position
if (setpointD <= cmpt_roue_DR) {outputD = 0;}
```



On peut voir des trous dans l'émission des impulsions qui correspondent à des rattrapages de vitesse.

Voir la deuxième vidéo.



Explications sur le programme :

// negatif ga G>D gauche ralentit rapide // positif D<G droite ralentit

`ecart_roue = cmpt_roue_DR - cmpt_roue_GA ;`

L'écart `ecart_roue` est regardé en permanence et coupe la sortie qui est en avance (voir la suite ligne )

Quand le robot arrive presque en position finale `setpointG - 10` alors on fixe une vitesse lente

//arret en vitesse lente //setpoint est le nombre de points visé

`if (setpointG - 10 <= cmpt_roue_GA) {vitesse_demandee_GA = vitesse_min_GA;}`

`if (setpointD - 10 <= cmpt_roue_DR) {vitesse_demandee_DR = vitesse_min_DR;}`

Quand le robot à cessé l'accélération et ne décélère pas encore alors on fixe une vitesse moyenne.

// entre les petites vitesses de démarrage et le ralentissement d'arrivée

`if ((setpointG - 10 > cmpt_roue_GA) & (cmpt_roue_GA > 10)) {vitesse_demandee_GA = vitesse_moyenne_GA;}`

//pour arrêter les moteurs quand on arrive en position


`if ((setpointD - 10 > cmpt_roue_DR) & (cmpt_roue_DR > 10)) {vitesse_demandee_DR = vitesse_moyenne_DR;}`

//démarrage en vitesse lente


`if (cmpt_roue_GA < 10) {vitesse_demandee_GA = vitesse_min_GA;}`

`if (cmpt_roue_DR < 10) {vitesse_demandee_DR = vitesse_min_DR;}`

//régulation lors du déplacement pour avoir les roues en phase

`if (ecart_roue > 0) {vitesse_demandee_DR = 0; } `

//setpointG est la valeur à atteindre par le codeur

`if (ecart_roue < 0) {vitesse_demandee_GA = 0; } `

//arret total quand on arrive en position

`if (setpointG <= cmpt_roue_GA) {outputG = 0;vitesse_demandee_GA=0; }`

//pour arrêter les moteurs quand on arrive en position

`if (setpointD <= cmpt_roue_DR) {outputD = 0;vitesse_demandee_DR=0;}`

envoi de la vitesse demandée sur le PWM



attention l'ordre des valeurs est important on prend toujours la dernière.

```
outputG = vitesse_demandee_GA;
```

```
outputD = vitesse_demandee_DR;
```