

Club Robot

Documentation sur les commandes Uart

Carte de propulsion

19 octobre 2014

1 Historique des révisions

Révision	Date	Modifications
A	01/2014	Création
B	10/2014	Correction

Table des matières

2 Introduction

Ce document a pour objectif de décrire l'ensemble des commandes UART comprises par la carte de commande moteur, ainsi que l'ensemble des informations fournies par la carte sur son statut.

Baud rate : 9600

Bit de stop : 1

Data bit : 8

Parité : none

3 Commandes

Les ordres peuvent être passés par un ou plusieurs mots codés en ASCII. On notera que n'importe quel séparateur peut être utilisé pour séparer les mots (espaces, tabulation, retour chariot)

3.1 Commandes simples en un mot

Commandes d'ordre :

Start : Lance le parcours configuré dans le robot.
Restart : Relance le parcours à partir du 1^{er} point du parcours.
Stop : Arrête le robot à sa position actuelle.
Continu : Fait repartir le robot au point où il s'est arrêté dans le parcours.
Telecommande : Déclenche le mode télécommande : le parcours est suspendu et seul les ordres de vitesse moteur peuvent modifier l'état du robot.
DfltAss : Re-initialise les paramètres d'asservissement par défaut.

Commandes de paramétrage des informations envoyées par la carte de propulsion

à compléter

Requête d'envoi d'une information par la carte de propulsion :

VaC?	: Valeur de consigne de la vitesse angulaire
V1C?	: Valeur de consigne de la vitesse du moteur 1
V2C?	: Valeur de consigne de la vitesse du moteur 2
DsC?	: Valeur de consigne de la distance parcourue
AgC?	: Valeur de consigne de l'angle du robot
AMC?	: Valeur de consigne de l'angle du modulo π
XrC?	: Valeur de consigne de la position du robot en x
YrC?	: Valeur de consigne de la position du robot en y
VmC?	: Valeur de consigne de la vitesse maximum
MdC?	: Mode de déplacement demandé au robot
EdC?	: Valeur de consigne de l'erreur de distance
EaC?	: Valeur de consigne de l'erreur angulaire tolérée pour acquitter la position demandée
ExC?	: Valeur de consigne de l'erreur en x tolérée pour acquitter la position demandée
EyC?	: Valeur de consigne de l'erreur en y tolérée pour acquitter la position demandée
dOC?	: Distance minimum avant l'identification d'un obstacle par l'ultrason
VdR?	: Valeur réelle de la vitesse de déplacement
VaR?	: Valeur réelle de la vitesse angulaire
V1R?	: Valeur réelle de la vitesse du moteur 1
V2R?	: Valeur réelle de la vitesse du moteur 2
DsR?	: Valeur réelle de la distance parcourue
AgR?	: Valeur réelle de l'angle du robot
AMR?	: Valeur réelle de l'angle du robot modulo π
XrR?	: Valeur réelle de la position du robot en x
YrR?	: Valeur réelle de la position du robot en y
EdR?	: Erreur entre la consigne de distance et la valeur réelle
EaR?	: Erreur entre la consigne d'angle et la valeur réelle
ExR?	: Erreur entre la consigne cordonnée en x et la valeur réelle
EyR?	: Erreur entre la consigne cordonnée en y et la valeur réelle
dOR?	: Distance à l'obstacle mesurée

Vitesse des moteurs

VtsM <Val1> <Val2> : Impose la consigne de vitesse au moteur 1 <Val1> et au moteur 2 <Val2>
 Exemple :
 VtsM 100 100
 Valeur comprise entre -500 et 500.

Asservissement

Asserv <Te> <KvPure> <KvPID> <Tiv> <Tdv> <Av> <KpD> <KpA> <Tip> <Tdp> <Ap> <cor_type_v>
 <corr_type_p> <Seuil> <Acceleration> <Rayon_Roue> <Distance_Roue> <Rapport_Reduction> <Nb_

imp>

<Te> : Période d'échantillonnage

<KvPure> : Gain du correction de vitesse pour une action proportionnelle pure.

<KvPID> : Gain du correction de vitesse pour une action PI ou PID.

<Tiv> : Constante de temps de l'action intégrale du correcteur de vitesse.

<Tdv> : Constante de temps de l'action dérivé du correcteur de vitesse.

<Av> : A_v

<KpD> : Gain du correcteur de position sur la vitesse de déplacement

<KpA> : Gain du correcteur de position sur la vitesse angulaire

<Tip> : Constante de temps de l'action intégrale du correcteur de position.

<Tdp> : Constante de temps de l'action dérivé du correcteur de position.

<Ap> :

<cor_type_v> : Type de correcteur pour la vitesse.

– 0=>P (fonctionnel)

– 1=>PI (fonctionnel)

– 2=>PID (non fonctionnel)

<corr_type_p> : Type de correcteur pour la position.

– 0=>P (fonctionnel)

– 1=>PI (non fonctionnel)

– 2=>PID (non fonctionnel)

<Seuil> Valeur max de l'erreur de vitesse après laquelle le correcteur de vitesse ne prend plus en compte l'action intégrale et/ou dérivé.

<Acceleration> : Valeur de l'accélération

<Rayon_Roue> : Rayon de la roue

<Distance_Roue> : Distance entre les roues du robot

<Rapport_Reduction> : Rapport de réduction total entre le moteur et la roue.

<Nb_imp> : Nombre d'impulsions du capteur incrémental par tour moteur.

Exemple (valeur par défaut) :

Asserv 0.004 10.0 5.0 0.1 1.0 1.0 0.8 0.02 1.0 1.0 1.0 1 0 50.0 3.0 40.0 203.0 24.0 500.0

Table de cordonnées

Il est possible d'envoyer à la carte de propulsion une liste de points où le robot doit se rendre. La commande à utiliser est : PstCons <nb> <Dis> <Ang> <x> <y> <Vmax_M> <Acc_M> <mode> <wait> <d_Obst> <ErrAng> <Err_x> <Err_y>

<nb> : Nombre de points du parcours

<Dis> : Consigne de distance (utile en mode 2) [mm].

<Ang> : Consigne de l'angle du robot (utile en mode 4) [rad].

<x> : Consigne de la position du robot en x (utile en mode 3, 5, 6) [mm].

<y> : Consigne de la position du robot en y (utile en mode 3, 5, 6) [mm].

<Vmax_M> : Consigne de la vitesse maximum.

<Acc_M> : Consigne d'accélération.

- <mode> : Mode de déplacement demandé au robot
- 0 => Mode télécommande
 - 1 => Vitesse moteur = V_{max_M} ;
 - 2 => Déplacement en distance et angle
 - 3 => Déplacement (soit avance, soit recule) en x et y
 - 4 => Ajustement de l'angle
 - 5 => Avance en x et y
 - 6 => Recule en x et y
 - 10 => Arrêt des moteurs.
- <wait> : Temps d'attente à chaque point [s].
- <d_ Obst> : Distance minimum avant l'identification d'un obstacle par l'ultrason [cm].
- <ErrAng> : Erreur angulaire tolérée pour acquitter la position demandée (utile en mode 4) [rad].
- <Err_ x> : Erreur en x tolérée pour acquitter la position demandée (utile en mode 3, 5, 6) [mm].
- <Err_ y> : Erreur en y tolérée pour acquitter la position demandée (utile en mode 3, 5, 6) [mm].

Les lignes **PstCons** doivent être enchainées pour constituer le parcours total. Pour terminer le parcours, il faut envoyer le mot **End**.

Exemple :

```
PstCons 10 0 0.0 0.0 0.0 150 3.0 0 0.05 20 0.05 10.0 10.0
PstCons 10 0 0.0 1000.0 0.0 150 3.0 3 0.0 20 0.05 10.0 10.0
PstCons 10 0 1.77 1000.0 0.0 150 3.0 4 0.0 2 0.05 10.0 10.0
PstCons 10 0 1.77 1000.0 1000.0 150 3.0 3 0.0 20 0.05 10.0 10.0
PstCons 10 0 3.14 1000.0 1000.0 150 3.0 4 0.0 2 0.05 10.0 10.0
PstCons 10 0 3.14 0.0 1000.0 150 3.0 3 0.0 20 0.05 10.0 10.0
PstCons 10 0 4.91 0.0 1000.0 150 3.0 4 0.0 2 0.05 10.0 10.0
PstCons 10 0 1.77 0.0 0.0 150 3.0 3 0.0 20 0.05 10.0 10.0
PstCons 10 0 1.77 0.0 0.0 150 3.0 4 0.0 2 0.05 10.0 10.0
PstCons 10 0 1.77 0.0 0.0 150 3.0 0 0.0 20 0.05 10.0 10.0
End
```

4 Feedback

La carte de propulsion transmettra soit en continu soit de manière individuelle (en fonction des informations sélectionnées pour la transmission) les données suivantes. Le type est toujours la même : le nom de la valeur, le caractère ":" et la valeur.

Dans le cas d'une transmission continue, une ligne d'informations est envoyée pour un instant donné (le temps sera toujours la 1^{er} donnée envoyée). Chaque bloc est séparé par des tabulations et la ligne se finit par le caractère "\n".

Dans le cas d'une transmission par requête, seul le bloc <nom de la valeur> :<valeur> est envoyé l'envoi se termine par le caractère "\n".

```
Tps:<val>
VdC:<val>
VaC:<val>
DsC:<val>
VdR:<val>
```

VaR:<val>
DsR:<val>
MdC:<val>
d0C:<val>
d0R:<val>
V1C:<val>
V2C:<val>
AgC:<val>
AMC:<val>
XrC:<val>
YrC:<val>
VmC:<val>
EdC:<val>
EaC:<val>
ExC:<val>
EyC:<val>
V1R:<val>
V2R:<val>
AgR:<val>
AMR:<val>
XrR:<val>
YrR:<val>
EdR:<val>
EaR:<val>
ExR:<val>
EyR:<val>