

# ***Le protocole de communication de la carte interface charlyrobot***

---

- *Composition d'une trame type*
- *Mode direct*
- *Mode indirect*

## COMPOSITION D'UNE TRAME TYPE

~ ~ ~

@0<Fonction><Paramètres>

@	En tête de trame.
0	Numéro d'adresse de la carte
<Fonction>	Les fonctions sont définies par un caractère. Pour les fonctions de déplacement ; si ce caractère est en majuscules, la carte retourne l'accusé de réception après exécution du déplacement ; si ce caractère est en minuscules, l'accusé de réception est retourné dès la réception de la trame.
<Paramètres>	L'unité de valeur des déplacements est exprimée en nombre de pas. L'unité de valeur pour les vitesses est exprimée en Hertz.

## MODE DIRECT

~ ~ ~

### Initialisation (mode direct)

@0G<n° de carte> Détermine le numéro de la carte, valeur par défaut 0.

@0<nbr d'axe> Détermine le nombre d'axes utilisés  
1er axe = 1, 2ème axe = 2, 3ème axe = 4

Exemple :    @01 machine à 1 axe  
                 @03 machine à 2 axes  
                 @07 machine à 3 axes

@0d<Fx,Fy,Fz>

Détermine la vitesse des courses de référence. La valeur par défaut est de 2000Hz.

Exemple :     @0d4000,4000,2000  
Les axes X et Y se déplaceront  
à une vitesse de 4000Hz et  
l'axe Z à 2000Hz.

@0R<nbr d'axe>

Demande d'exécution d'une course de référence.

@0r<nbr d'axe>

1er axe = 1, 2ème axe = 2, 3ème axe = 4

Exemple :     @0r7 course de référence  
des 3 axes.

@0? Autotest

La carte retourne sa version et ses caractéristiques d'interpolation.

### Déplacement (mode direct)

@0A<Px,Fx,Py,Fy,Pz1,Fz1,Pz2,Fz2>

@0a<Px,Fx,Py,Fy,Pz1,Fz1,Pz2,Fz2>

Déplacement en mode relatif des 3 axes aux positions Px, Py, puis Pz1 et Pz2 aux vitesses Fx, Fy, Fz1, Fz2..

@0M<Px,Fx,Py,Fy,Pz1,Fz1,0,32>

@0m<Px,Fx,Py,Fy,Pz1,Fz1,0,32>

Déplacement en mode absolu des 3 axes aux positions Px, Py, Pz1 aux vitesses Fx,Fy,Fz1.

Les valeurs 0 et 32 sont des paramètres constructeur à ne pas modifier.

@0e<plan>

Définition du plan d'interpolation.

Plan xy = 0, xz = 1, yz = 2

Le plan d'interpolation par défaut est le plan xy.

Exemple : @0e1

Les interpolations devront être exécutées dans le plan horizontal xz.

@0z1	Activation de l'interpolation en 3D.
@0z0	Désactivation de l'interpolation en 3D.
@0f0	Définition du sens de rotation pour une interpolation circulaire. 0 sens de rotation anti-trigonométrique -1 sens de rotation trigonométrique.
@0n<nbr d'axe>	Prendre la position actuelle comme point d'origine. 1er axe = 1, 2ème axe = 2, 3ème axe = 4  <i>Exemple : @0n7</i> <i>La position nouvelle des 3 axes est prise comme nouvelle origine.</i>
@0P	Demande de position des 3 axes, la carte retourne une trame de 18 caractères hexadécimaux.
@0Q	Interrompre définitivement tout processus en cours



### Interpolation circulaire (mode direct)

@0y<R,v,d,xs,ys,rdx,rdy> Déplacement suivant un arc de cercle en interpolation circulaire.

R : longueur de l'arc  
v : vitesse (de 32 à 10 000 Hz)  
d : paramètre d'interpolation (voir ci-dessous)  
xs : point de départ en X  
ys : point de départ en Y  
rdx : sens de déplacement en X (voir tableau)  
rdy : sens de déplacement en Y (voir tableau)

avec r : rayon de l'arc  
β : angle d'arrivée  
α : angle de départ

### Calcul de R :

- Formule par approximation :

$$R' = \frac{4r (\beta - \alpha)}{180} \times \frac{\text{nbre de pas moteur par tour}}{\text{pas de vis}}$$

R est la valeur entière de  $R' + 0,5$

**- Formule exacte :**

A = angle de départ ; E = angle d'arrivée

```

if {Sens = Sens anti-horaire} then begin
  while (A<0) do begin A:=A+2.0*pi; E:=E+2.0*pi; end;
  while (A>=pi/2.0) do begin A:=A-pi/2; E:=E-pi/2; end;
  R:=0.0;
  while (E-A>=pi/2.0) do begin E:=E-pi/2.0; R:=R+2.0*Rayon; end;
  if (E>pi/2.0) then begin R:=R+2.0*Rayon; E:=E-pi/2.0; end;
  R:=[R+Rayon*(cos(A)-cos(E)+sin(E)-sin(A))] x  $\frac{\text{nbre de pas mot.}}{\text{pas de vis}}$  ;end

else begin {Sens = Sens horaire }
  while (A>0) do begin A:=A-2.0*pi; E:=E-2.0*pi; end;
  while (A<=pi/2.0) do begin A:=A+pi/2; E:=E+pi/2; end;
  R:=0.0;
  while (A-E>=pi/2.0) do begin E:=E+pi/2.0; R:=R+2.0*Rayon; end;
  if (E<-pi/2.0) then begin E:=E+pi/2.0; R:=R+2.0*Rayon; end;
  R:=[R+Rayon*(cos(A)-cos(E)+sin(A)-sin(E))] x  $\frac{\text{nbre de pas moteur}}{\text{pas de vis}}$  ;

end;
if (R<0) then R:=-R
  
```

**Calcul de xs et ys :**

1) xs

$$x's = (r \times \cos \alpha) \times \frac{\text{nbre de pas moteur}}{\text{pas de vis}}$$

$x's > 0$

xs est la valeur entière de  $x's + 0,5$

$x's < 0$

xs est la valeur entière de  $x's - 0,5$



2)  $y_s$

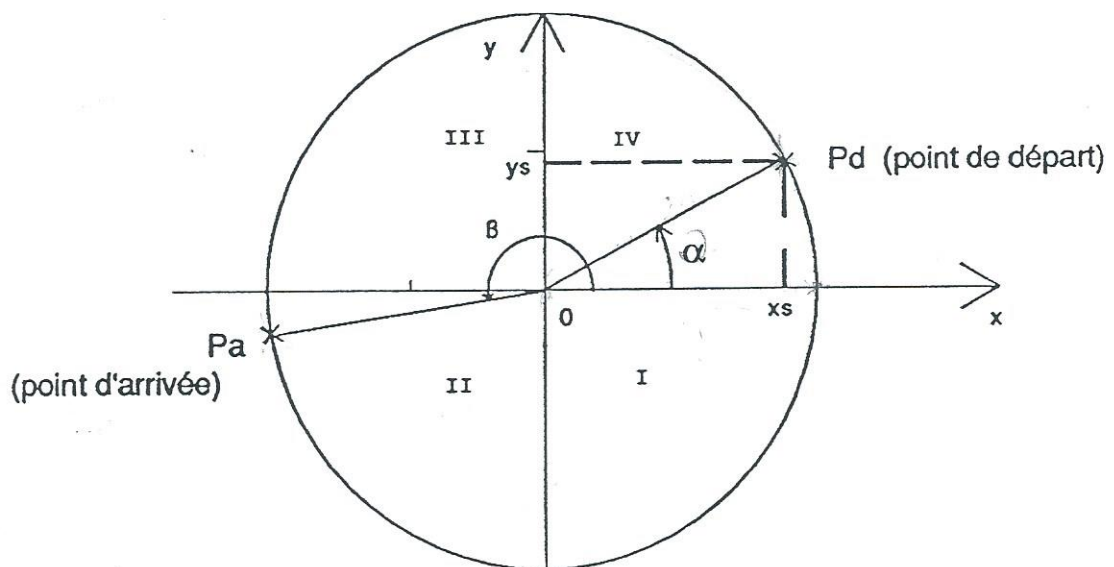
$$y's = (r \times \sin\alpha) \times \frac{\text{nbre de pas moteur}}{\text{pas de vis}}$$

$y_s$  est la valeur entière de  $y's + 0,5$

Sens horaire :

(!)

La carte interface doit être initialisée sur 3 axes



(!!)

Le centre du cercle est toujours placé en fonction du point de départ.

Tableau pour les coefficients rdx et rdy :

Partie	I	II	III	IV
rdx	-1	-1	+1	+1
rdy	+1	-1	-1	+1

Calcul du paramètre d :

$$x = r \cdot \cos \alpha$$

$$y = r \cdot \sin \alpha$$

$$d' = [ (rdx \cdot rdy \cdot r) + (rdx \cdot rdy \cdot \text{somme} (r-1)) \\ - rdx \cdot \text{somme} (x + (rdx - rdy) / 2) \\ + rdy \cdot \text{somme} (y + (rdx + rdy) / 2) ] / 2$$

$d' > 0$  : d est la valeur entière de ( $d' \times \text{nbre de pas moteur} / \text{pas}$ ) + 0,5

$d' < 0$  : d est la valeur entière de ( $d' \times \text{nbre de pas moteur} / \text{pas}$ ) - 0,5

**Procédure somme :**

somme (x)

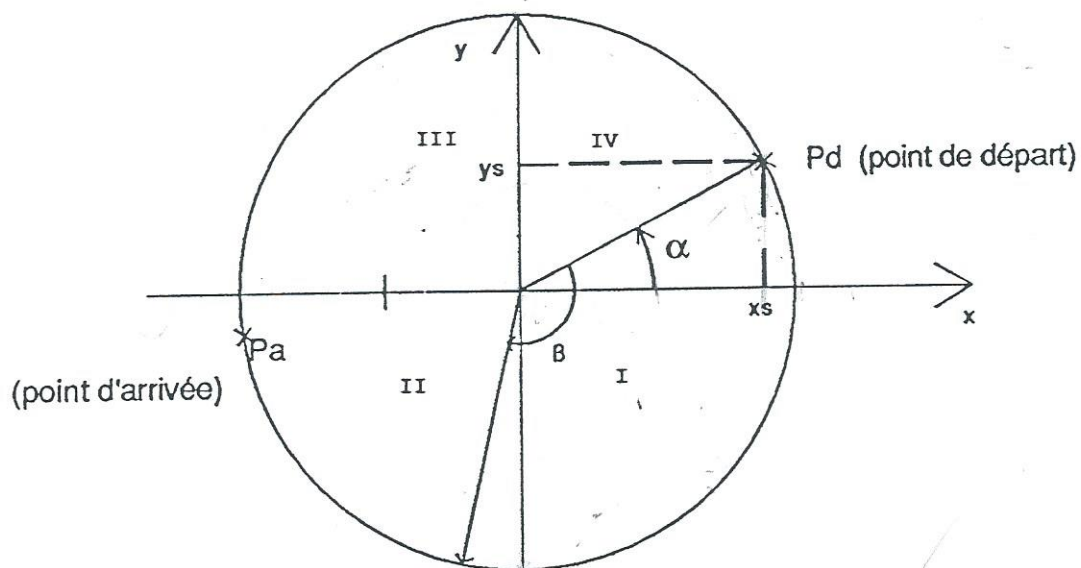
si  $x > 0$ , alors  $x \cdot (x+1)$

si  $x < 0$ , alors  $x \cdot (x-1)$

## Sens trigonométrique :

(!)

La carte interface doit être initialisée sur 3 axes.



(!!)

Le centre du cercle est toujours placé en fonction du point de départ.

Tableau pour les coefficients rdx et rdy :

Partie	I	II	III	IV
rdx	+1	+1	-1	-1
rdy	-1	+1	+1	-1

Calcul du paramètre d :

$$d' = [ (-rdx \cdot rdy \cdot rayon) - (rdx \cdot rdy \cdot somme (rayon - 1)$$

$$- rdx \cdot somme (x + (rdx + rdy) / 2 )$$

$$+ rdy \cdot somme (y + (rdy - rdx) / 2)] / 2$$

Pour calculer d, référez vous à la procédure expliquée dans la partie trigonométrie.

### Entrées Sorties (mode direct)

Les 2 ports de sortie ont les adresses 65529 et 65530, le port d'entrée a l'adresse 65531.

@0B<adresse,data>      Activation des sorties

Exemple :      @0B65529,255  
Activation des 8 sorties du port 1.

@0B65529,0  
Désactivation des 8 sorties du port 1.

@0b<adresse>      lecture des ports entrées et sorties

Exemple :      @0b65531  
Lecture du port d'entrées. la valeur retour est codée en hexadécimal sur 8 bits.



Utilitaires (mode direct)

@0C<on/off>

Envoi des caractères CR/LF en fin de trame.

@0C1 = ON    @0C0 = OFF

Exemple :    @0C1

*Dès la réception de cette instruction, la carte enverra les caractères CR / LF en fin de trame.*

@0S

@0s

Exécution du programme mémorisé en RAM sur la carte série.

@0u

Copie le contenu de la RAM sur la carte mémoire.

**@0t**

Exécution du programme en pas à pas.  
A l'exécution de chaque pas, la carte  
retourne une trame contenant une trace  
de l'exécution.

**@0i**

Mémoriser les instructions suivantes.  
Cette instruction permet le passage en  
mode indirect.

**@0k**

Effacement du programme mémorisé par  
le système **charlyrobot**.

\* \* \*

## MODE INDIRECT

~ ~ ~

Dans la structure d'un programme, les instructions suivantes sont toujours comprises entre l'instruction @0i qui indique le début de la mémorisation des instructions et 9 qui détermine la fin de la zone mémorisée.

### Déplacement (mode indirect)

0<Px,Fx,Py,Fy,Pz1,Fz1,Pz2,Fz2>

Déplacement en mode relatif des 3 axes aux positions Px, Py, puis Pz1 et Pz2 aux vitesses Fx, Fy, Fz1, Fz2.

m<Px,Fx,Py,Fy,Pz1,Fz1,0,32>

Déplacement en mode absolu des 3 axes aux positions Px, Py, puis Pz1 aux vitesses Fx, Fy, Fz1.

Les valeurs 0 et 32 sont des paramètres constructeur à ne pas modifier.

6<Px,Fx,Py,Fy,Pz1,Fz1,0,32>

Déplacement en mode absolu des trois axes aux positions Px, Py, Pz1 aux vitesses Fx, Fy, Fz1 jusqu'à réception d'une impulsion sur l'entrée.

Les valeurs 0 et 32 sont des paramètres constructeur à ne pas modifier.

z1

Activation de l'interpolation en 3D.

z0

Désactivation de l'interpolation en 3D.

f0

Définition du sens de rotation pour une

f-1

interpolation circulaire :

0 sens de rotation anti-trigonométrique

-1 sens de rotation trigonométrique.

y<r,v,d,xs,ys,rdx,rdy>

Déplacement suivant un arc de cercle, en interpolation circulaire.

e<plan>

Définition du plan d'interpolation.

plan xy = 0, xz = 1, yz = 2.

Le plan d'interpolation par défaut est le plan xy.

### Entrées / Sorties (mode indirect)

Les 2 ports de sortie ont les adresses 65529 et 65530, le port d'entrée a l'adresse 65531.

#### 4<Option>

Activation et désactivation de l'entrée et de la sortie de la carte 1 entrée / 1 sortie.

Les options sont au nombre de 6 :

- 1 Mise à 1 de la sortie
- 2 Mise à 0 de la sortie
- 3 Mise à 1 de la sortie durant 0,5 sec.
- 4 Attendre une impulsion sur l'entrée.
- 5 Envoyer une impulsion sur la sortie et attendre une impulsion sur l'entrée.
- 6 Attendre une impulsion sur l'entrée et envoyer une impulsion sur la sortie.



p<adresse,mode,data> Ecriture d'une valeur sur les ports de sorties en mode bit ou byte.

Mode : - Adressage en bit à bit valeur 0 ou 1.  
- Adressage en mode byte valeur de 0 à 255.

Data : - En mode bit valeur 0 ou 1.  
- En mode byte valeur de 0 à 255.

Le mode bit permet d'adresser les sorties une à une et le mode byte d'adresser un port de 8 sorties avec un seul mot

Exemple :    *p65529,2=0*  
                  *Mise à zéro de la sortie 2 en*  
                  *mode bit.*  
                  *p65529,10=255*  
                  *Mise à 1 des 8 sorties en mode*  
                  *byte.*

`o<adresse,mode,data,offset>`

Lecture d'une valeur sur le port d'entrée en mode bit ou byte et saut suivant celle-ci.

Mode : - Adressage en bit à bit valeur de 1 à 8.  
- Adressage en mode byte valeur 0.

Data : - En mode bit valeur 0 ou 1.  
- En mode byte valeur de 0 à 255.

Offset : - Si la valeur lue est égale à la valeur data alors un saut est effectué de la valeur de l'offset.

*Exemple :   o65531,2=0,10  
Si l'entrée 2 est à zéro, un saut de 10 instructions sera effectué par le programme.*

*o65531,10=127,2  
Si la valeur 127 est lue sur les 8 entrées en mode byte, alors le programme saute 2 instructions.*

Utilitaires (mode indirect)

- |                      |  |
|----------------------|--|
| 1<ascii>             | Envoi d'un code ascii compris entre 65 et 127.   |
| 2<ascii,offset>      | Branchement de la valeur de l'offset si le caractère reçu est égal à la valeur de ascii+1.   |
| 3<nbr boucle,offset> | Exécute une boucle sur les instructions comprises entre cette instruction et l'instruction pointée par l'offset, le nombre de boucles étant défini.<br>L'offset doit être une valeur négative. |
| 5<Temps>             | Temporisation en 1/10 de seconde.  |

7<nbr d'axe>	Demande d'exécution d'une course de référence. 1er axe = 1, 2ème axe = 2, 3ème axe = 4.  <i>Exemple : 77 course de référence des 3 axes .</i>
8<N° de carte,option>	Communication avec une autre carte de même type.
Rn	Demande d'exécution d'une course de référence sur n axes et attendre l'accusé de réception.
rn	Demande d'exécution d'une course de référence sur n axes sans attendre l'accusé de réception.
S	Exécution du programme mémorisé en RAM sur l'autre carte série et attendre l'accusé de réception.

s                      Exécution du programme mémorisé en  
RAM sur l'autre carte série sans attendre  
l'accusé de réception.

9                      Fin de la zone des données mémorisées.

k<n° de touche,offset>    Saut du programme de la valeur de l'offset  
suivant le numéro de la touche du clavier  
**charlyrobot** actionné.

~ ~ ~