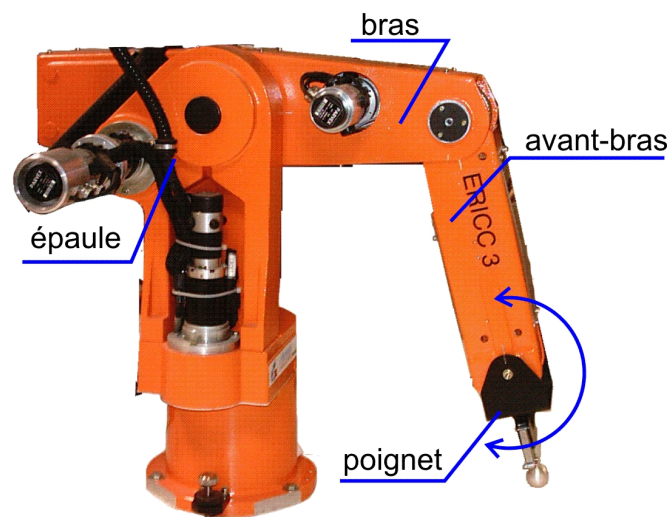


# Robot ERICC3

Chaîne fonctionnelle articulation d 'épaule  
Modélisation - simulation



# 1 Performances annoncées

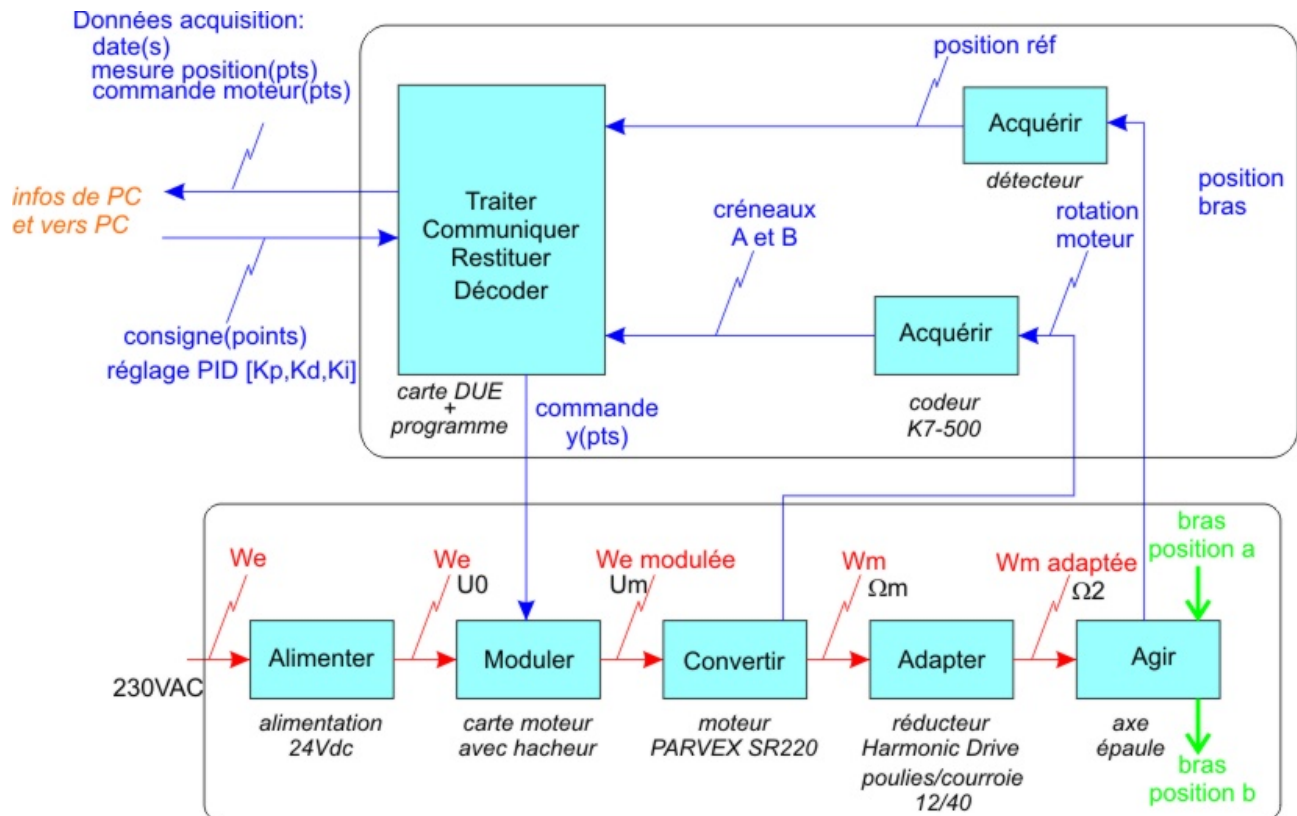
Le robot Ericc3 est un robot 5 axes pour lequel on étudie l'axe de poignet. Les performances annoncées à l'époque (1997) de son achat étaient pour l'articulation d'épaule :

- Course :  $-45^{\circ} \rightarrow 90^{\circ}$
- Vitesse maximum : 70 degrés/s
- accélération : 410 degrés/s<sup>2</sup>

Cet axe asservi en position a été remis en fonctionnement en modifiant essentiellement la chaîne d'information. Pour la chaîne d'énergie la fonction moduler est réalisée par une carte moteur DRI0041.

## 2 Description du système

Le schéma chaîne d'énergie -chaîne d'information est donné ci-dessous :



## 2.1 Chaîne d'énergie :

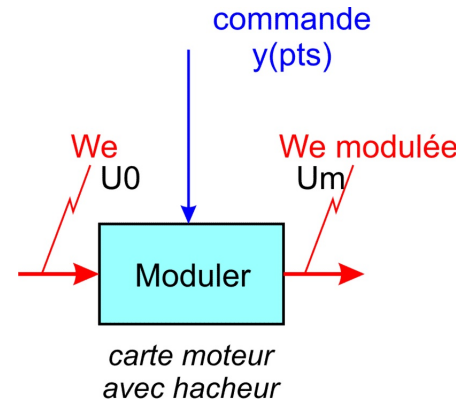
### 2.1.1 Alimentation :

L'alimentation de la carte moteur est  $U_0 = 24V$ , obtenue avec un adaptateur AC/DC

### 2.1.2 Carte moteur :

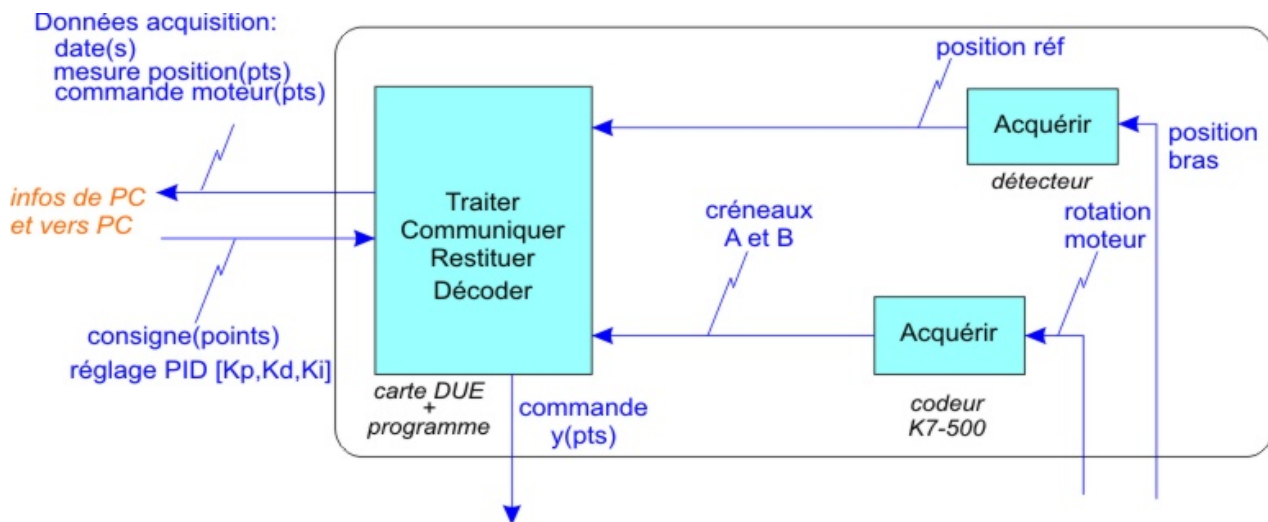
La programmation de la carte DUE impose une commande en points représentée par un entier dans l'intervalle  $[-255 .. 255]$

### 2.1.3 Autres composants : leur description est dans le dossier technique initial



## 2.2 Chaîne d'information

### 2.2.1 Codeur incrémental: 500 traits / tour de moteur.



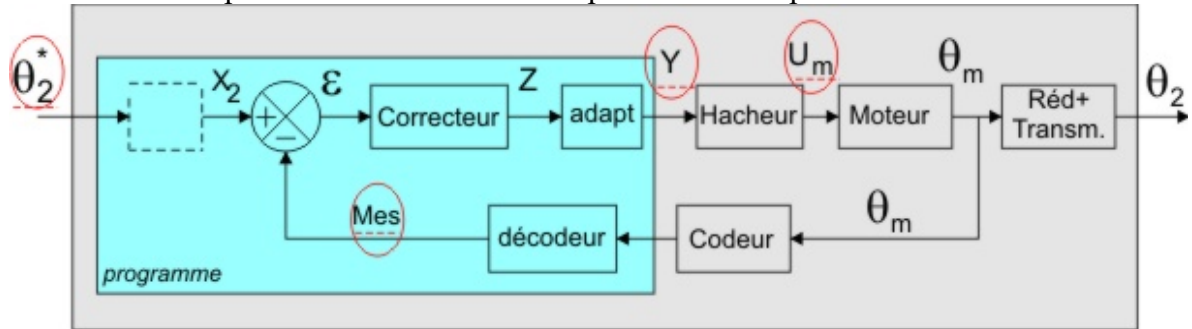
Le débit des informations envoyées par le codeur incrémental **K7-500** peut atteindre des dizaines de milliers par seconde. La carte **Arduino Due** dispose d'un **décodage intégré en quadrature**. On obtient donc une résolution de  $500 \times 4 = 2000$  pts/tr moteur

### 2.2.2 Carte Due +programme :

- Elle permet la communication avec le PC de l'utilisateur par la liaison série :
  - réception du choix dans le menu, choix des consignes .....
  - envoi des mesures et autres messages.
- La carte évalue l'écart  $\epsilon$  dans l'asservissement de position et élabore la commande  $y$  en points qui sont envoyés à la **carte moteur**.

## 2.3 Schéma-bloc de l'asservissement de position :

L'asservissement en position de l'articulation d'épaule est décrit par le schéma ci-dessous :




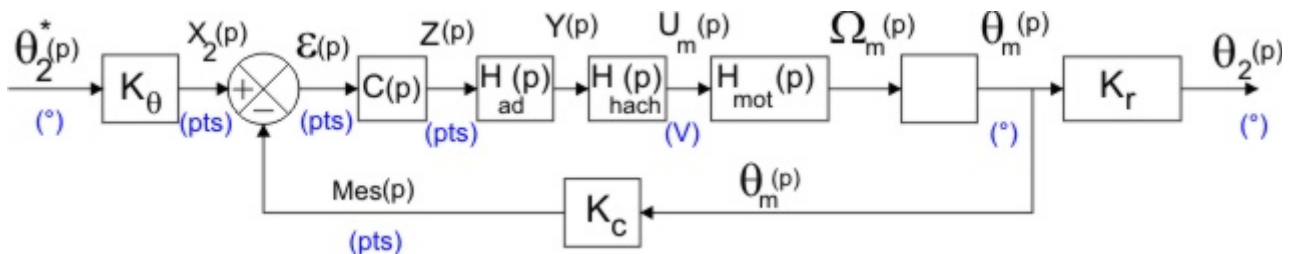
 grandeurs directement accessibles par le logiciel.

Schéma-bloc en vue de simulation :




- $C(p) = K_p$  correcteur proportionnel pour ce T.P.
- $H_{ad}(p)$  : L'écart en point est un entier très grand et la commande du hacheur doit être comprise entre -255 et 255 points ; l'adaptation réalise donc deux fonctions :
  - diviser le signal par 1000,
  - limiter le signal par une saturation en dehors de l'intervalle -255..+255.
- $H_{mot}(p)$  : on choisira une fonction de transfert d'ordre 1 simple dont le gain statique  $K_m$  et la constante de temps  $\tau_m$  seront déterminés expérimentalement.

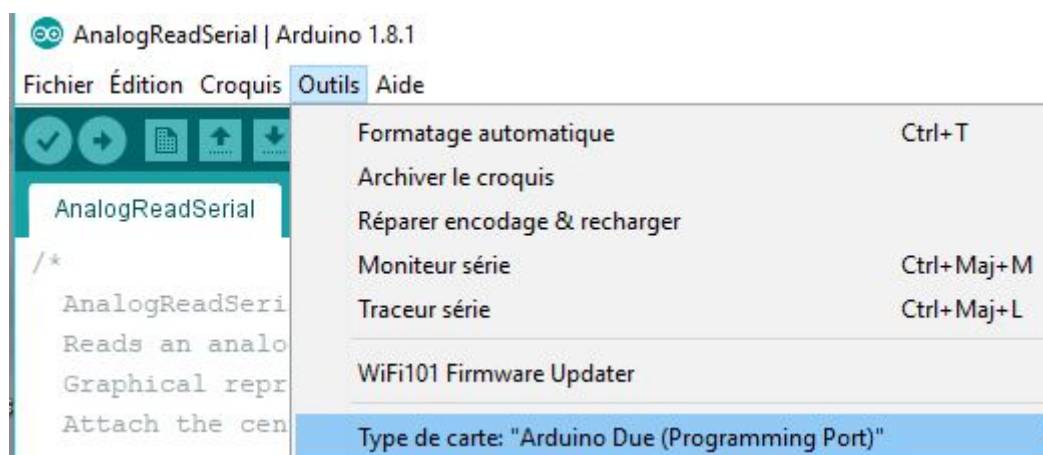
## 3 Objectifs :

- Construire un modèle de l'asservissement de position de l'axe d'épaule en utilisant les informations du dossier technique que l'on comparera avec des informations issues d'expériences.
- Valider le modèle en réalisant des essais.
- Vérifier les performances :
  - Course :  $-45^\circ \rightarrow 90^\circ$  en déplaçant manuellement le bras ;
  - Vitesse maximum : 70 degrés/s à partir de l'expérience réalisée pour l'identification ;
  - accélération : 410 degrés/s<sup>2</sup> à partir de l'expérience réalisée pour l'identification ;

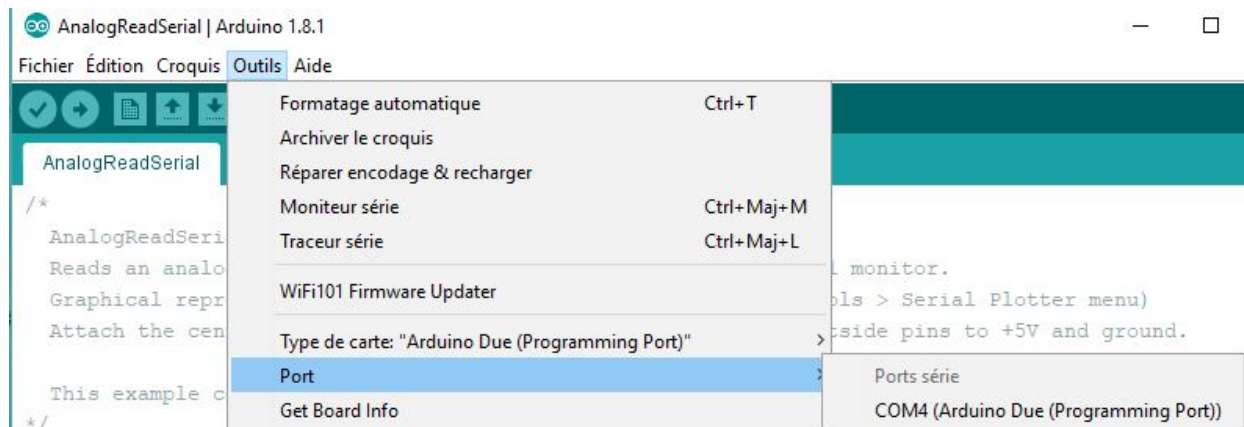
## 4 Annexe : pilotage de l'axe d'épaule

Remarque : Le pilotage de l'axe d'épaule se fait en utilisant le moniteur série intégré au logiciel Arduino. On n'a pas à charger un programme, ni en modifier un quelconque et encore moins d'en télécharger un dans la carte.

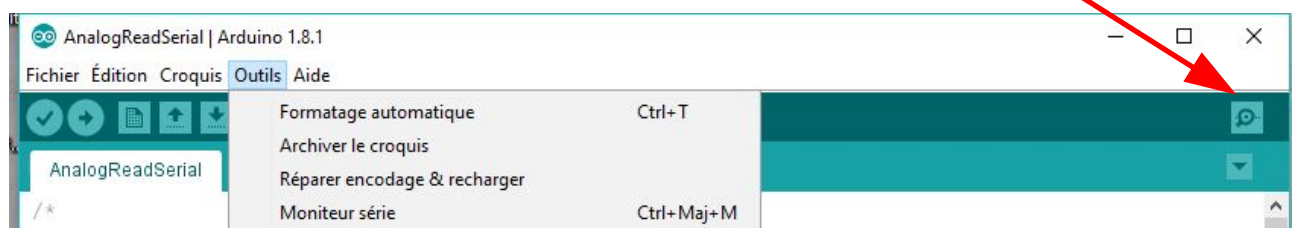
- Brancher l'alimentation du système : bloc 24VDC
- Brancher le câble USB sur le PC
- Démarrer l'IDE Arduino : le logiciel reconnaît la présence de la carte sur un port série
- Préciser le type de carte :  dans la liste choisir (en bas) Arduino DUE sur programming port



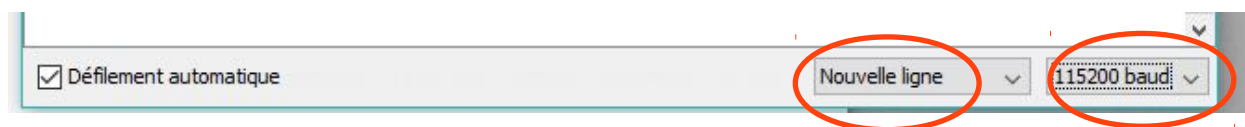
- Préciser le port de communication



- Activer le moniteur série : Outil puis Moniteur série ou Ctrl+Maj+M ou



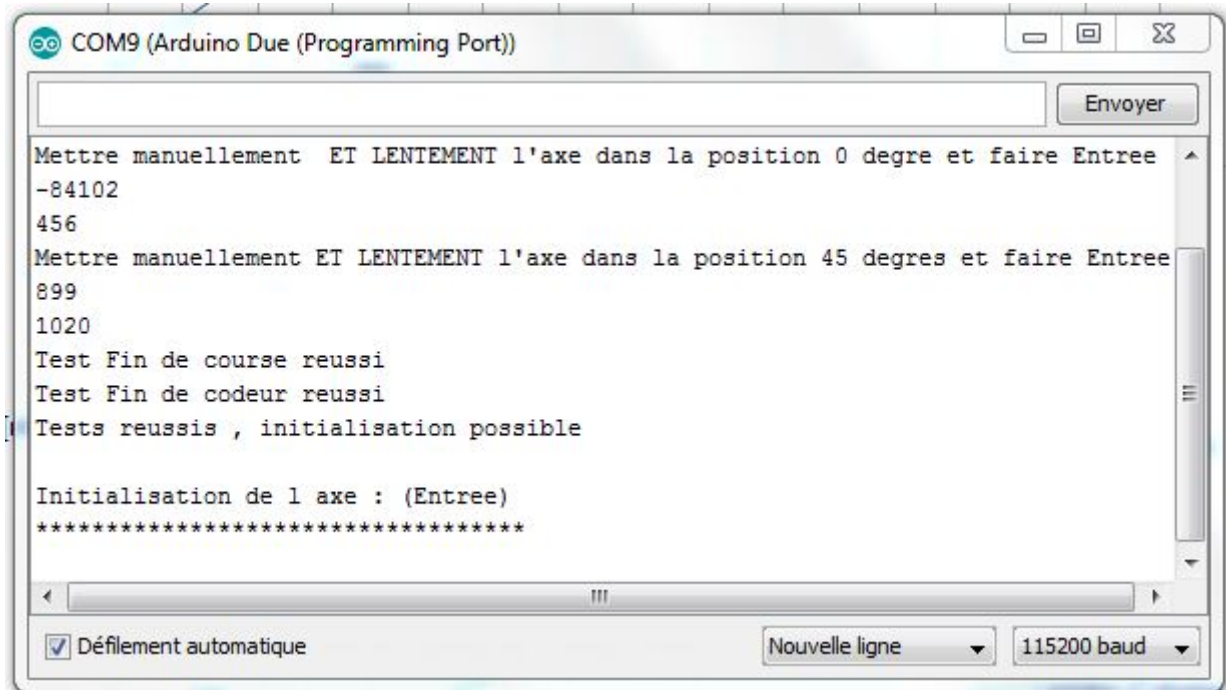
Paramètres du moniteur série : dans le bas de la fenêtre du moniteur série on doit avoir :



## Logiciel de pilotage du robot : par le moniteur série

**Attention, les mouvements du robot sont rapides !!!!!**

- **menu de démarrage** : il permet de vérifier que les capteurs utiles sont bien actifs que le frein est relâché.



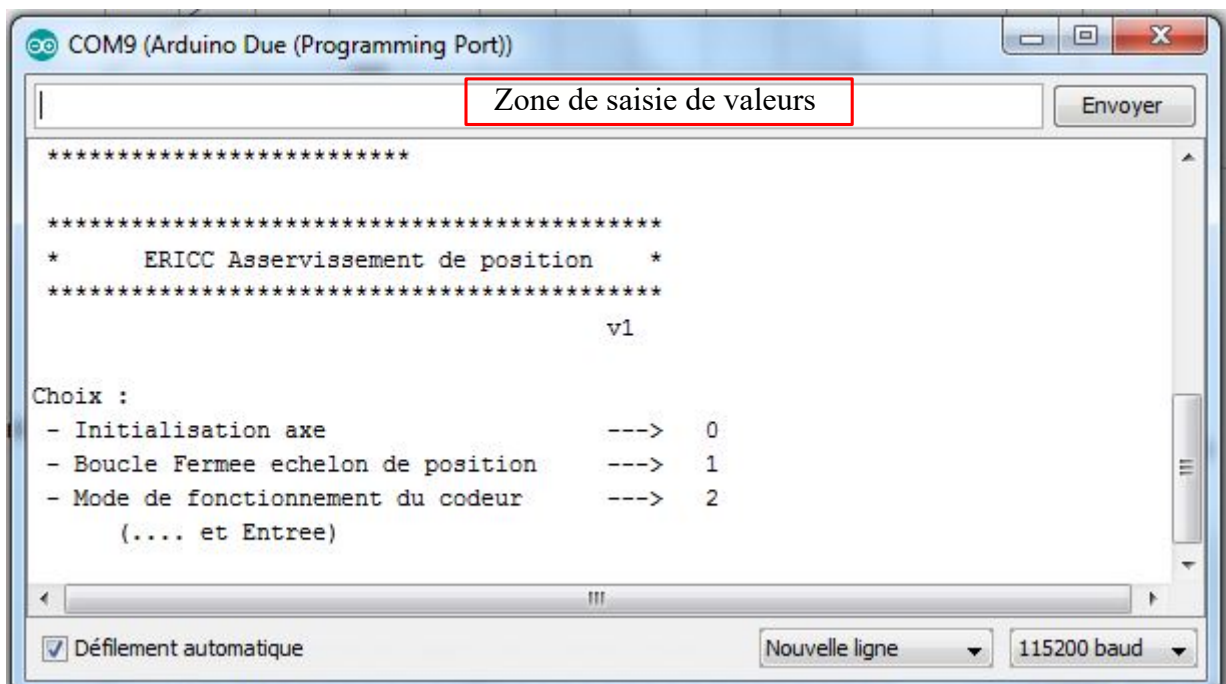
```
COM9 (Arduino Due (Programming Port))

Mettre manuellement ET LENTEMENT l'axe dans la position 0 degre et faire Entree
-84102
456
Mettre manuellement ET LENTEMENT l'axe dans la position 45 degres et faire Entree
899
1020
Test Fin de course reussi
Test Fin de codeur reussi
Tests reussis , initialisation possible

Initialisation de l axe : (Entree)
*****

Défilement automatique [X] Nouvelle ligne 115200 baud
```

**menu principal** : si les tests sont réussis on accède au menu principal



```
COM9 (Arduino Due (Programming Port))

Zone de saisie de valeurs

*****

*****
*   ERICC Asservissement de position   *
*****

                                v1

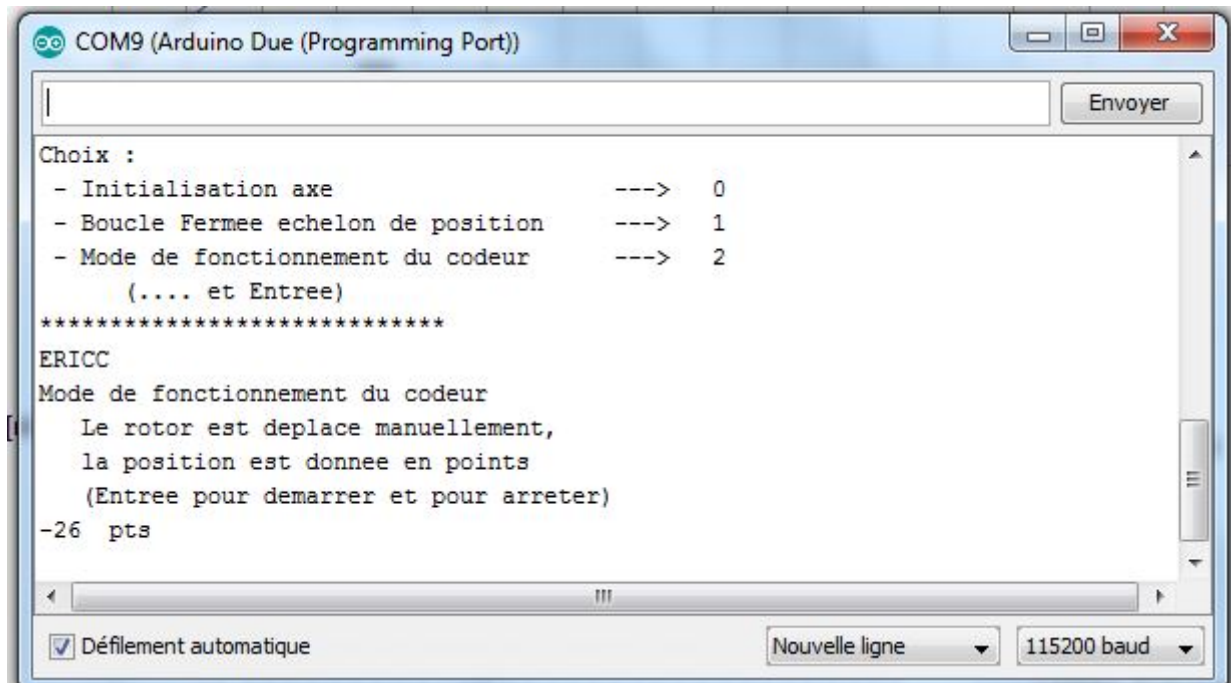
Choix :
- Initialisation axe                --->  0
- Boucle Fermee echelon de position --->  1
- Mode de fonctionnement du codeur  --->  2
  (... et Entree)

Défilement automatique [X] Nouvelle ligne 115200 baud
```



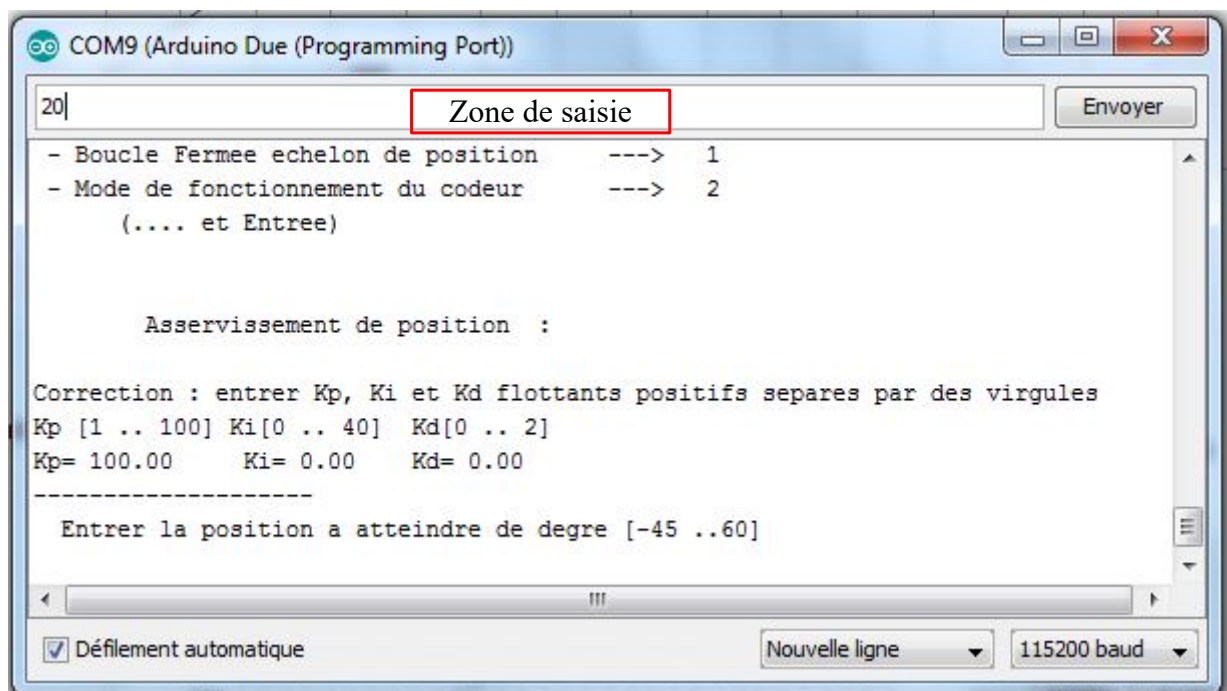
**Initialisation de l'axe :** permet de mettre le robot en position 0 degré.

**fonctionnement du codeur:** permet d'afficher l'évolution des points du codeur en déplaçant manuellement le bras.



```
COM9 (Arduino Due (Programming Port))
Choix :
- Initialisation axe          ---> 0
- Boucle Fermee echelon de position ---> 1
- Mode de fonctionnement du codeur ---> 2
  (... et Entree)
*****
ERICC
Mode de fonctionnement du codeur
  Le rotor est deplace manuellement,
  la position est donnee en points
  (Entree pour demarrer et pour arreter)
-26 pts
```

**Boucle fermée échelon de position :**



```
COM9 (Arduino Due (Programming Port))
20| Zone de saisie
- Boucle Fermee echelon de position ---> 1
- Mode de fonctionnement du codeur ---> 2
  (... et Entree)

Asservissement de position :

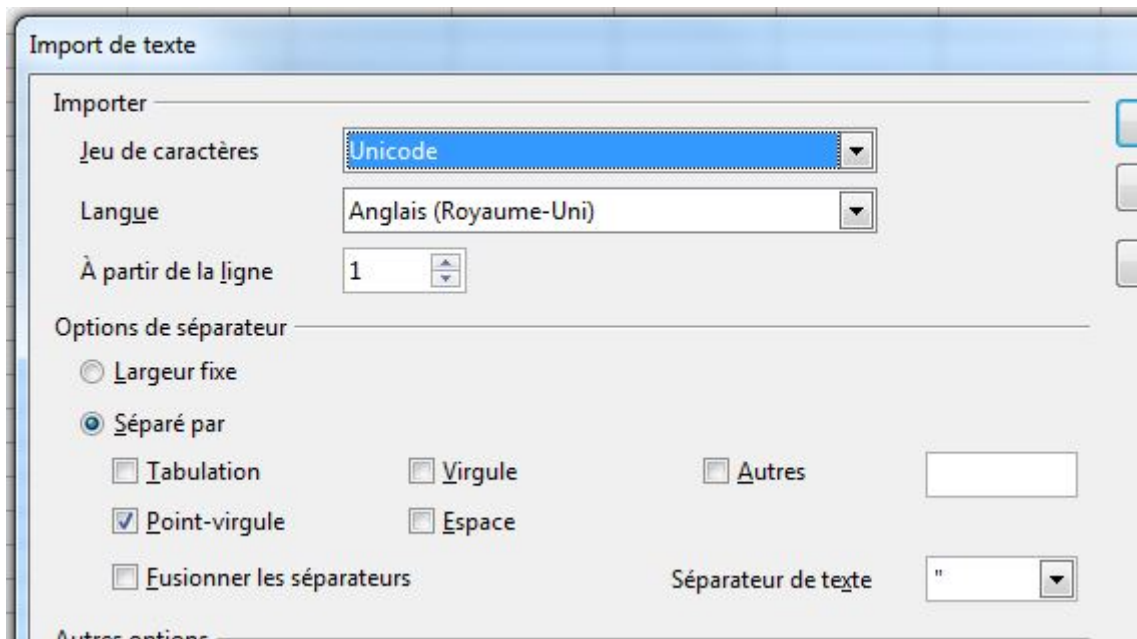
Correction : entrer Kp, Ki et Kd flottants positifs separes par des virgules
Kp [1 .. 100] Ki[0 .. 40] Kd[0 .. 2]
Kp= 100.00 Ki= 0.00 Kd= 0.00
-----
Entrer la position a atteindre de degre [-45 ..60]
```

A la fin du mouvement on copie le tableau de mesures affiché dans le moniteur série



## Exploitation des mesures dans le tableur de Libre Office

Dans une feuille vide on colle le tableau de mesures ; en prenant les précautions suivantes on les points du tableaux(séparateur décimal) deviendront des virgules dans Libre Office ce qui est obligatoire.



Utilisation de Scilab et du bloc Read CSV : on collera le tableau de mesure dans un fichier texte . Le séparateur décimal dans Scilab est le point.