

Systeme de controle d'axes

GalilV2

Auteur : J. Coquet

Version courante du document : 1.1

Date de creation document : Mars 2007

Derniere modification : le 21 mai 2007

Historique des modifications

Date	Revision	Description	Auteur	Relecteur
13/03/07	1.0	Version initiale	J. Coquet	
21/05/07	1.1	Modifs mineures suite à relecture		A.Buteau

1	But de l'application.....	1
2	Structure matérielle et logicielle de l'application.....	2
2.1	Architecture matérielle.....	2
2.2	Architecture logicielle	3
3	Que sait faire GalilAxis (et le microcode embarqué).....	3
4	Limitation connues de GalilAxis.....	4
4.1	Limitations sur les commandes proposées.....	4
4.2	Types d'axes supportés.....	4
4.3	Types de positionnement supportés.....	4
5	Interface logicielle Tango.....	5
5.1	Properties.....	5
5.2	Les attributs.....	6
5.3	Les commandes.....	8
6	Configuration logicielle.....	8
7	TODO : Suite de la doc.....	10
7.1	Documentation de la classe AxisRawDataReader(si besoin)	10
7.2	Documentation de la classe GalilSlit (si besoin)	10

1 But de l'application

Le système décrit dans ce document a été conçu pour permettre la commande générique d'axes motorisés.

Ces axes sont essentiellement destinés à être commandés en *positionnement*.

Une **commande en vitesse** est possible mais sans contrôle précis.

Ces axes sont constitués au minimum de :

- Moteur pas à pas avec ou sans codeur
- Moteur « servo » à courant continu ou brushless obligatoirement couplé à 1 ou 2 codeurs
- Moteur piezo avec retour de position par codeur ou analogique
- Des limit switch seront présents pour les mouvements finis (avec butées mécaniques)

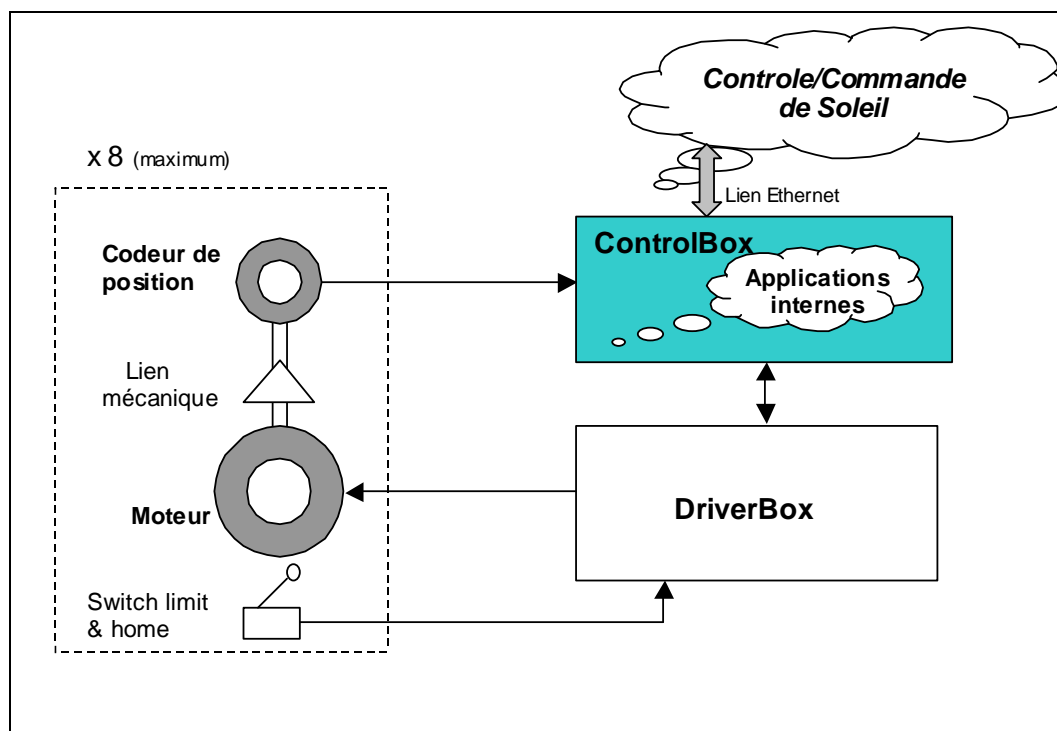
Codeurs supportés :

- codeurs incrémentaux à signaux TTL ou RS422 avec ou sans Top 0 de position initiale
- codeur absolu norme SSI
- codeur sinus avec conversion TTL/RS422.

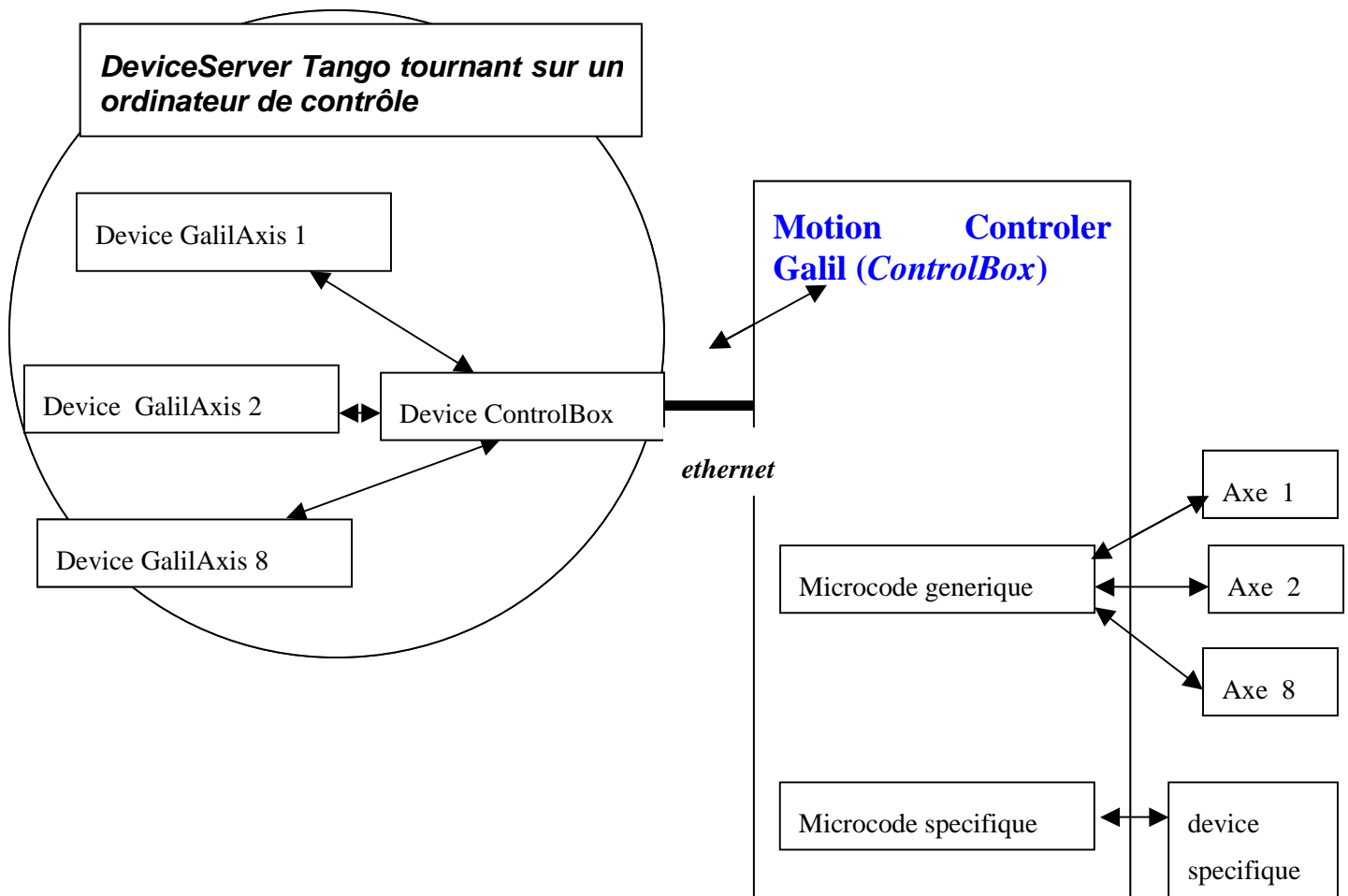
Un détecteur HOME de position initiale peut être présent (avec des limitations décrites plus bas)

2 Structure matérielle et logicielle de l'application

2.1 Architecture matérielle



2.2 Architecture logicielle



- Le device server ControlBox embarque les devices suivants :
 - ControlBox pour les commandes du domaine de la ControlBox toute entière : version/révision, RESET, informations sur les erreurs de communication...
 - GalilAxis permet un contrôle individuel d'1 moteur
 - GalilSlit permet la gestion d'une fente à 2 volets pilotés par 2 moteurs indépendants
 - AxisRawDataReader permet de voir les valeurs codeur, moteur ; position commandées en unités hardware, pour des finalités de maintenance

3 Que sait faire GalilAxis (et le microcode embarqué)

GalilAxis est essentiellement conçu pour

- positionner un axe à une côte précise.
- la mise en position initiale de l'axe (déplacement de l'axe à une position connue par exemple limit switch de précision ou sur détecteur Home précis plus éventuellement sur le top 0 d'un codeur incrémental)

Il offre en plus

- Un mode de déplacement manuel (mode jog)

4.1 Limitations sur les commandes proposées

4.2 Types d'axes supportés

4.2.1 Axes testés

- Moteur Pas à pas seul
- Moteur Pas à pas avec codeur incrémental (ou Sinus + convertisseur TTL)

4.2.2 Axes en cours de validation

- Moteur Pas à pas avec codeur absolu SSI
- Moteur Servo (brush ou brushless) avec codeur incrémental (ou Sinus + convertisseur TTL)
- Moteur Servo (brush ou brushless) avec codeur absolu SSI
- Moteur Piezo sans retour de position
- Moteur Piezo avec retour de position analogique

4.3 Types de positionnement supportés

4.3.1 Sans codeur : Moteurs pas à pas uniquement

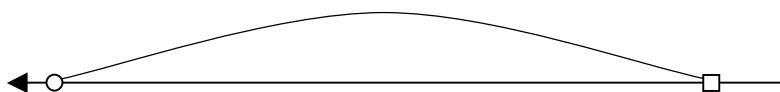
- Positionnement simple
- Positionnement avec rattrapage de jeu mécanique

4.3.2 Avec codeur : tous type de moteur

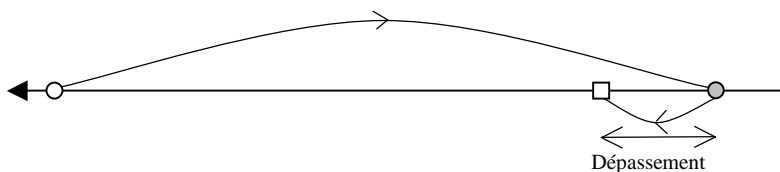
- Avec les moteurs de type « servo » (essentiellement les moteurs DC) il y a toujours un contrôle de suivi de trajectoire et l'axe se mettra en défaut si l'erreur de suivi est trop importante. Consulter le motoriste pour le réglage des paramètres d'erreur de suivi.
- Avec les moteurs Stepper, il est possible d'activer un mode dit de « maintenance de position » qui simule le suivi de trajectoire. En cas d'erreur de suivi, le contrôleur arrête le mouvement en cours, corrige l'erreur, et relance le mouvement, sans intervention.

4.3.3 Types de positionnement supportés

- Positionnement simple



- Positionnement avec rattrapage de jeu mécanique



5.1 Propriétés

Les propriétés sont créées automatiquement au démarrage du device, mais leurs valeurs doivent être saisies par l'utilisateur

AxisNumber	<p>Numéro de l'axe pour la control box. Valeur possible : De A à H On peut saisir également le numéro d'axe de 0 à 7. (Pour les amateurs Galil les 4 lettres d'axe XYZW sont aussi supportés) Correspondance :</p> <table><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr><tr><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td><td>G</td><td>H</td></tr><tr><td>X</td><td>Y</td><td>Z</td><td>W</td><td></td><td></td><td></td><td></td></tr></table> <p>Doit être défini obligatoirement Pas de valeur par défaut.</p>	0	1	2	3	4	5	6	7	A	B	C	D	E	F	G	H	X	Y	Z	W				
0	1	2	3	4	5	6	7																		
A	B	C	D	E	F	G	H																		
X	Y	Z	W																						
AxisPositionRatio	<p>Nombre d'unités utilisateur par pas codeur si présence codeur, par pas moteur si pas de codeur. Exemple : unité utilisateur = 1 micron, 1 pas codeur = 100 nano, ➔AxisPositionRatio = 10.0 La formule utilisée est la suivante : <i>position _unités_utilisateurs = position codeur _unités_codeur * AxisPositionRatio - offset</i> peut être < 0. pour définir un sens de fonctionnement opposé au sens « motoriste » Dans ce cas le device adapte les limit switch pour que les attributs et status soient conformes au sens utilisateur défini par le signe de AxisPositionRatio Valeur par défaut : 1.0</p>																								
AxisEncoderType	<p>Type de codeur raccordé à l'axe.</p> <ul style="list-style-type: none">0 : pas de codeur.1 : codeur incrémental2 : codeur absolu <p>Défaut = -1 (device en erreur) Doit être défini. Attention : Toute modification de cette property définie par le motoriste implique une intervention du motoriste pour reparamétrages internes dans la ControlBox .</p>																								
AxisInitType	<p>Type de mise en position initiale de l'axe. Chaîne ASCII décrivant la stratégie pour obtenir la position initiale Possibilités :</p> <ul style="list-style-type: none">LSBWD : x ; mise en position initiale sur le limit switch backward, x étant le nombre de fois ou il accoste le LimitSwitch backward avec des vitesses décroissantes, x = [1...15]LSFWD : x ; idem LSBWD , pour le limit switch forwardFI : x ; pour l'index du codeur incrémental, x = [-7...+7] le signe donne le sens de la première rechercheFH : x ; pour le détecteur Home s'il existe, x = [-7...+7] le signe donne le sens de la première rechercheDP ; pas de prise de référence ; utiliser Define positionexemple d'abord rechercher le limit switch forward puis 3 fois l'index<ul style="list-style-type: none">LSFWD :1 ;FI :-3 ; <p>Valeur par défaut : DP</p>																								
AxisInitPosition	<p>Position initiale de l'axe en unités utilisateurs. Cette valeur est chargée sur succès de la commande InitializeReferencePosition. Valeur par défaut : 0.0</p>																								
EnableMaintenance Mode	<p>Moteurs steppers avec codeurs uniquement. Valide la correction dynamique de trajectoire Valeur par défaut : false (inhibé)</p>																								
AllowRemoteCbox	<p>La RemoteCBox est la télécommande manuelle qui se branche directement sur la ControlBox pour piloter « à vue » les axes. Cette télécommande utilise des E/S TOR de la ControlBox. Dans certains cas (utilisation de ces E/S pour d'autres applications par</p>																								

	ex.) cette property permet d'inhiber (valeur = false) ou d'autoriser (valeur = true) cette RemoteCBox.
EnableDutyCycle	Cette Property est destinée aux motoristes. Elle n'est pas créée automatiquement. Elle valide pour certains moteurs un comportement spécifique : le « Duty Cycle » déterminant le pourcentage de temps pendant lequel le moteur peut être Power ON. Utilisé pour certains moteurs sous vide et certains piezos sous vide avec le micro-code spécifique « Duty Cycle ». Cette property a true n'interdit pas une commande de déplacement moteur OFF. <i>La valeur par défaut est « false » (inhibé)</i>
Percent	Cette Property est destinée aux motoristes. Elle n'est pas créée automatiquement. Elle permet de compenser une non-linéarité de l'axe dans certaines limites : voir doc du micro-code. <i>Valeur par défaut : 1.0</i>
Retry	Cette Property est destinée aux motoristes. Elle n'est pas créée automatiquement. Nombre de réessais avant que le positionnement ne soit déclaré en erreur. <i>Valeur par défaut : 5</i>

5.2 Les attributs

position	Read/Write <ul style="list-style-type: none"> Read : position courante de l'axe en unités utilisateur Write : lance un positionnement à la valeur en unités utilisateur La formule utilisée pour le calcul de la valeur lue est la suivante : <i>position_unités_utilisateurs = (CurrentEncoderPosition _unités_codeur * UserEncoderRatio)-Offset</i>
acceleration	Read/Write <ul style="list-style-type: none"> Read : Accélération courante de l'axe. Write : modification de l'accélération <i>En unité utilisateur par seconde²</i>
deceleration	Read/Write <ul style="list-style-type: none"> Read : Décélération courante de l'axe. Write : modification de la décélération <i>En unité utilisateur par seconde²</i>
velocity	Read/Write <ul style="list-style-type: none"> Read : vitesse courante de l'axe. Write : modification de la vitesse <i>En unité utilisateur par seconde</i>
accuracy	Read/Write Précision du mouvement, fenêtre dans laquelle le positionnement est considéré comme correct. <ul style="list-style-type: none"> Read : précision courante de l'axe. Write : modification de la précision <i>En unité utilisateur.</i> Limite inférieure : 1 pas moteur.
backlash	Read/Write Compensation de jeu mécanique par approche du point de positionnement dans la direction choisie par le signe de l'attribut. <ul style="list-style-type: none"> Read : distance courante de backlash. Write : modification de la distance de backlash <i>En unité utilisateur</i>
offset	Read/Write Offset ajouté à la position courante (attribut position). Laisse inchangées les positions codeur et moteur. <ul style="list-style-type: none"> Read : Offset courant de l'axe. Write : modification de l'offset

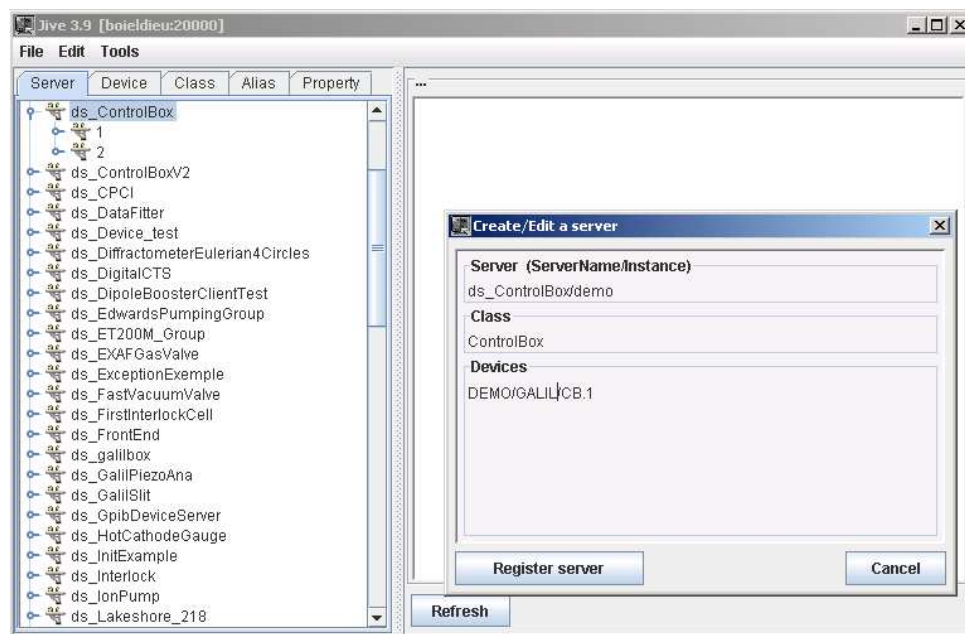
5.3 Les commandes

Forward	Commande un mouvement continu dans le sens + utilisateur selon UserDirection
Backward	Commande un mouvement continu dans le sens - utilisateur selon UserDirection
Stop	Arrêt du mouvement en cours
State	Retourne l'état de l'axe <ul style="list-style-type: none">• OFF : pas de communication• STANDBY : attente de commande• MOVING : axe en déplacement• ALARM : axe en alarme (non bloquant)• FAULT : axe en défaut (erreurs 2, 7, 14, 15)• DISABLE : signale une erreur non traitée.
Status	Texte d'erreur
InitializeReferencePosition	Initialise l'axe à une position connue en fonction de la property AxisInitType. Exécute selon la property AxisInitType : <ul style="list-style-type: none">• recherche d'index• recherche de Home• Recherche du limit switch dans le sens défini par signe de la property AxisPositionRatio, puis charge la valeur de la property AxisInitPosition
MotorON	Validation du driver de puissance.
MotorOFF	Inhibition du driver de puissance. Seulement possible après une commande stop. Attention : le moteur étant hors tension, la charge peut éventuellement entraîner le moteur.
DefinePosition	définit la position à la valeur utilisateur passée en argument
ComputeNewOffset	on lui passe la position utilisateur à laquelle on voudrait être, elle calcule l'offset et en assure la mémorisation.

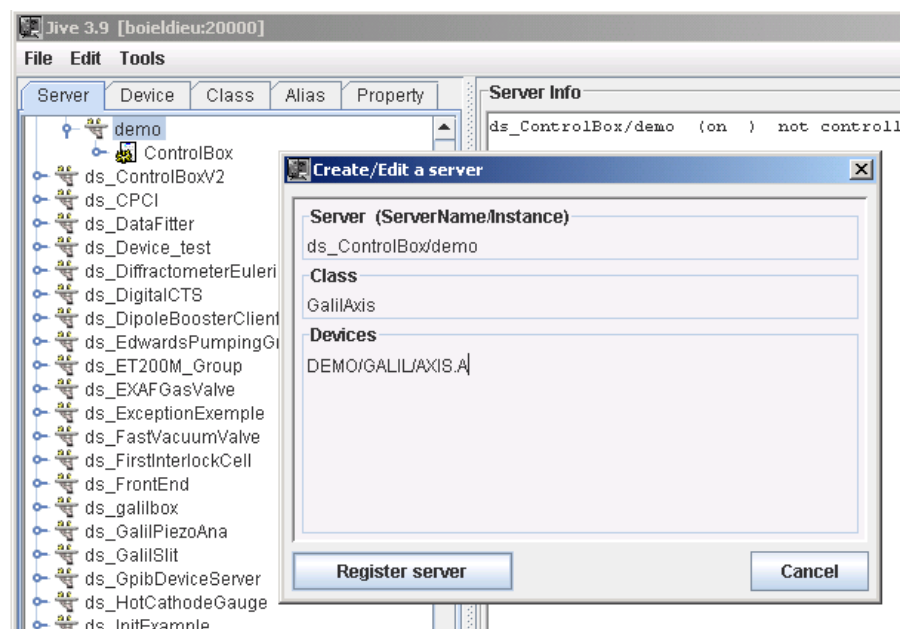
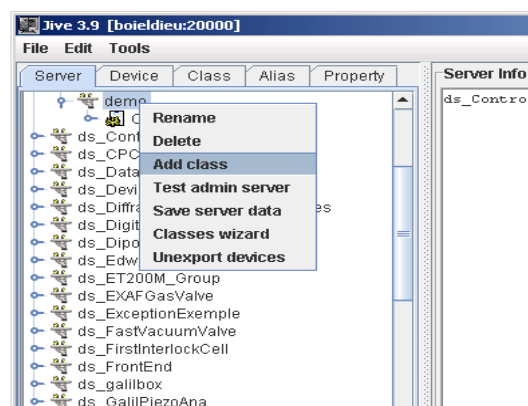
6 Configuration logicielle

La ControlBox doit avoir été configurée par les motoristes. Les axes doivent être raccordés, testés par les motoristes.

Installez dans la Database Tango et configurez le nouveau DServer voir installation device ControlBox



Ajoutez la classe GalilAxis



7 TODO : Suite de la doc

7.1 Documentation de la classe AxisRawDataReader(si besoin)

7.2 Documentation de la classe GalilSlit (si besoin)