

PROCÉDURE DE CALIBRATION ET TESTS

Bras Robotique 6 Axes – Servomoteurs Dynamixel AX-12

AVERTISSEMENT DE SÉCURITÉ

Avant toute manipulation, s'assurer que :

- L'alimentation est déconnectée lors des connexions/déconnexions
- La tension d'alimentation est conforme : 12V DC $\pm 10\%$
- La polarité est respectée (inversée = destruction des servos)
- L'espace de travail est dégagé (risque de mouvement imprévu)

1. PRÉPARATION DU MATÉRIEL

1.1 Matériel nécessaire

Qté	Désignation	Référence
6	Servomoteur Dynamixel AX-12A	ROBOTIS AX-12A
1	Contrôleur USB2AX	Xevelabs USB2AX v3.2a
1	Alimentation 12V DC / 5A min	Barrel jack 2.1mm
1	Câble USB A vers micro-USB	-
1	PC sous Linux ou Windows	Python 3.x + pyserial

1.2 Logiciels requis

- **Dynamixel Wizard 2.0** :
https://emanual.robotis.com/docs/en/software/dynamixel/dynamixel_wizard2/
- **Python 3.x avec bibliothèques** : pyserial, dynamixel-sdk

Installation Python :

```
pip install pyserial dynamixel-sdk
```

2. CONNEXION ET VÉRIFICATION DU SYSTÈME

2.1 Schéma de connexion

[ALIMENTATION 12V] → [HUB POWER] → [SERVO 1] → [SERVO 2] → ... → [SERVO 6]

↓ (bus TTL 3 fils)
[USB2AX] ←USB→ [PC]

2.2 Procédure de connexion

1. **DÉBRANCHER l'alimentation 12V**
2. Connecter les servomoteurs en série (bus TTL : Data, VCC, GND)
3. Connecter le dernier servo au contrôleur USB2AX

4. Connecter l'USB2AX au PC via câble USB
5. **Vérifier la polarité de l'alimentation (CRITIQUE !)**
6. Brancher l'alimentation 12V au hub de distribution
7. Vérifier la LED d'alimentation du premier servo (doit s'allumer en vert)

2.3 Identification du port série

Linux :

```
ls -l /dev/ttyACM*
```

Généralement : /dev/ttyACM0 ou /dev/ttyUSB0

Windows :

Gestionnaire de périphériques → Ports (COM et LPT) → USB Serial Device (COMx)
Noter le numéro de port COM (ex: COM3)

3. SCAN DES SERVOMOTEURS AVEC DYNAMIXEL WIZARD

3.1 Lancement du scan

8. Lancer Dynamixel Wizard 2.0
9. Cliquer sur l'icône [Options] (roue dentée) en haut à droite
10. Configurer les paramètres de scan :

Paramètre	Valeur recommandée
Port	/dev/ttyACM0 (Linux) ou COMx (Windows)
Baud Rate	1000000 bps (1 Mbps) – valeur par défaut AX-12
Protocol	Dynamixel Protocol 1.0 (AX-12 utilise Protocol 1.0)
ID Range	0-252 (scan complet)

11. Cliquer sur [Scan] pour détecter les servomoteurs
12. Noter les ID détectés et leurs informations dans le tableau suivant :

Axe	ID	Modèle	Firmware	Baud	Position actuelle
1					
2					
3					
4					
5					
6					

3.2 Diagnostic des problèmes de scan

Symptôme	Solution
----------	----------

Aucun servo détecté	<ul style="list-style-type: none"> • Vérifier l'alimentation 12V (LED verte sur servo) • Vérifier la connexion USB2AX au PC • Tester un autre baud rate (57600 bps) • Vérifier le câblage du bus série
Seulement quelques servos détectés	<ul style="list-style-type: none"> • Vérifier les connexions en série • Un servo peut avoir un baud rate différent • Scanner à plusieurs baud rates
Erreur de communication	<ul style="list-style-type: none"> • Réduire la longueur des câbles • Vérifier la qualité des connexions • Réduire le baud rate à 57600 bps
Conflit d'ID (deux servos même ID)	<ul style="list-style-type: none"> • Isoler chaque servo individuellement • Changer l'ID de chacun séparément • Recommandation : ID 1-6 pour axe 1-6

4. CALIBRATION DES POSITIONS ET BUTÉES

4.1 Définition de la position HOME

La position HOME est la configuration de référence du robot (tous les axes alignés, bras replié ou vertical).

Procédure pour chaque axe :

13. Dans Dynamixel Wizard, sélectionner le servo correspondant à l'axe
14. Activer le mode Position (vérifier que CW/CCW Angle Limit $\neq 0$)
15. Débloquer le servo en désactivant le Torque (Torque Enable = 0)
16. Positionner manuellement l'axe à la position HOME souhaitée
17. Lire la valeur du codeur (Present Position : 0-1023)
18. Noter cette valeur comme position HOME de l'axe

4.2 Configuration des butées logicielles

Les butées logicielles (CW Angle Limit et CCW Angle Limit) empêchent les mouvements hors de la plage sécurisée et protègent la structure mécanique.

Méthodologie :

19. Identifier les limites mécaniques de chaque axe (butées physiques, câbles)
20. Mesurer les positions limites en mode Torque Off :
 - Position CW (sens horaire) : valeur minimale
 - Position CCW (sens antihoraire) : valeur maximale
21. Appliquer une marge de sécurité de 5% (environ 50 unités sur 1024)
22. Écrire les valeurs dans les registres du servo :
 - CW Angle Limit (adresse 6-7)
 - CCW Angle Limit (adresse 8-9)

Exemple de calcul pour l'axe 1 (base) :

Position mécanique CW min : 100 unités

Position mécanique CCW max : 900 unités

Marge de sécurité 5% : 50 unités

→ **CW Angle Limit = 150 unités**

→ **CCW Angle Limit = 850 unités**

4.3 Tableau de calibration à compléter

Axe	HOME (unités)	CW Limit	CCW Limit	Plage (°)
1				
2				
3				
4				
5				

Formule de conversion : Plage angulaire (°) = (CCW - CW) × 300 / 1024

5. TESTS DE COMMUNICATION SÉRIE

5.1 Test avec script Python basique

Créer un script Python pour tester la lecture de position d'un servo :

```
#!/usr/bin/env python3 import serial import time # Configuration PORT =
'/dev/ttyACM0' # Adapter selon système BAUDRATE = 1000000 SERVO_ID = 1 # Ouvrir
le port série ser = serial.Serial(PORT, BAUDRATE, timeout=1) time.sleep(0.1) #
Paquet de lecture de position (Instruction 0x02, Read Data) # Format: [0xFF, 0xFF,
ID, Length, Instruction, Address, Data_Length, Checksum] packet = [0xFF, 0xFF,
SERVO_ID, 0x04, 0x02, 0x24, 0x02] checksum = (~(sum(packet[2:])) & 0xFF)
packet.append(checksum) # Envoi du paquet ser.write(bytearray(packet))
time.sleep(0.01) # Lecture de la réponse response = ser.read(8) if len(response)
== 8: position = response[5] + (response[6] << 8) print(f"Position du servo
ID {SERVO_ID} : {position} (unités)") print(f"Position angulaire : {position *
300 / 1024:.2f}°") else: print("Erreur de communication") ser.close()
```

5.2 Test de mouvement contrôlé

Tester le déplacement d'un servo vers différentes positions :

23. Dans Dynamixel Wizard, sélectionner un servo
24. Activer le Torque (Torque Enable = 1)
25. Modifier la consigne Goal Position par incréments de 100
26. Observer le mouvement et vérifier :
 - Le servo atteint la position cible (erreur < 3 unités)
 - Pas de vibrations ou oscillations
 - Le servo s'arrête aux butées définies
 - Température stable (< 60°C)

5.3 Vérification des paramètres PID

Les servos AX-12 possèdent un asservissement PID configurable. Valeurs par défaut :

- P Gain (adresse 28) : 32
- I Gain (adresse 27) : 0
- D Gain (adresse 26) : 0

En cas d'oscillations ou de manque de précision, ajuster le P Gain :

- Oscillations → réduire P Gain (tester 24, 16)
- Manque de réactivité → augmenter P Gain (tester 40, 48)

6. CHECKLIST DE VALIDATION

✓	Critère de validation
---	-----------------------

	Tous les servomoteurs (6) sont détectés lors du scan
	Chaque servo possède un ID unique (1-6)
	La position HOME est définie pour chaque axe
	Les butées logicielles CW et CCW sont configurées avec marge de sécurité
	Les servos répondent correctement aux commandes de position
	Aucune oscillation ou vibration anormale détectée
	La température des servos reste < 60°C en fonctionnement
	La communication série est stable (pas d'erreurs de checksum)
	Le script Python de test fonctionne correctement

Validation par l'enseignant : _____ *Date :* _____