

# Présentation du produit

Booster T1 est une plateforme de développement de robots humanoïdes conçue pour les développeurs et autres utilisateurs, caractérisée par sa légèreté, sa flexibilité et sa durabilité.

## Composition du produit

Le robot T1 se compose d'une tête, d'un torse, de bras et de jambes, avec un total de 23 degrés de liberté, permettant des mouvements flexibles et un contrôle de la posture.

- La tête possède 2 degrés de liberté, incluant une articulation de lacet et une articulation de tangage. Elle intègre une caméra de profondeur, un réseau de microphones et un haut-parleur.
- Chaque bras possède 4 degrés de liberté, comprenant l'articulation de tangage de l'épaule, l'articulation de roulis de l'épaule, l'articulation de lacet de l'épaule et l'articulation du coude.
- La taille possède 1 degré de liberté, qui correspond à l'articulation de lacet de la taille.
- Chaque jambe possède 6 degrés de liberté, incluant l'articulation de tangage de la hanche, l'articulation de roulis de la hanche, l'articulation de lacet de la hanche, l'articulation du genou et l'articulation de flexion-extension de la cheville.
- La carte de contrôle et la batterie sont installées dans le torse.

## Fonctions du produit

1. Marche omnidirectionnelle
  1. Permet la marche vers l'avant, vers l'arrière et latéralement.
  2. Permet la rotation et la marche complexe.
2. Résistance aux perturbations pendant la marche
  1. Peut marcher sur des surfaces irrégulières.
  2. Peut résister à certains chocs lors de la marche.
3. Actions prédéfinies
  1. Faire signe.
  2. Serrer la main.
  3. Dansant.
  4. Récupération automnale.
4. Protection de sécurité
  1. Passe automatiquement en mode d'amortissement en cas de dysfonctionnement afin d'éviter tout dommage.
  2. Arrêt d'urgence brutal.

# Spécifications du produit

Taper	Paramètres de spécification	Description
Paramètres de base	Hauteur (en position debout)	1,18 m
	Poids	Environ 30 kg
degrés de liberté	Nombre total de degrés de liberté	23
	Degrés de liberté à une seule jambe	6
	DoF à un seul bras	4 (extensible)
	degrés de liberté de la taille	1
	Degrés de liberté de la tête	2
Paramètres opérationnels	vitesse de marche	0,5 m/s
	Vitesse de virage	1,5 rad/s
Paramètres de la batterie	Autonomie de la batterie	1,5h
	Temps de charge	≤2h
	Cycle de vie	≥500 fois
Plateforme informatique	Basique	AGXOrin*1
	Standard	i7 *1
		AGXOrin*1
Capteurs	Caméra	Caméra de profondeur
	Microphone	Réseau de microphones
	Conférencier	1
Fonction de sécurité	Arrêt d'urgence du robot	1
	Alertes sonores	Alerte batterie faible, alerte de surchauffe des articulations

<b>Méthodes de communication</b>	Connexion sans fil	WIFI 6
	Bluetooth	Bluetooth 5.2
	Normes de certification	CE/FCC
<b>Bruit</b>	Bruit de marche	≤70dB
<b>Adaptabilité environnementale</b>	Adaptabilité environnementale	-10°C~45°C
	Humidité de fonctionnement	5 % à 90 %, sans condensation

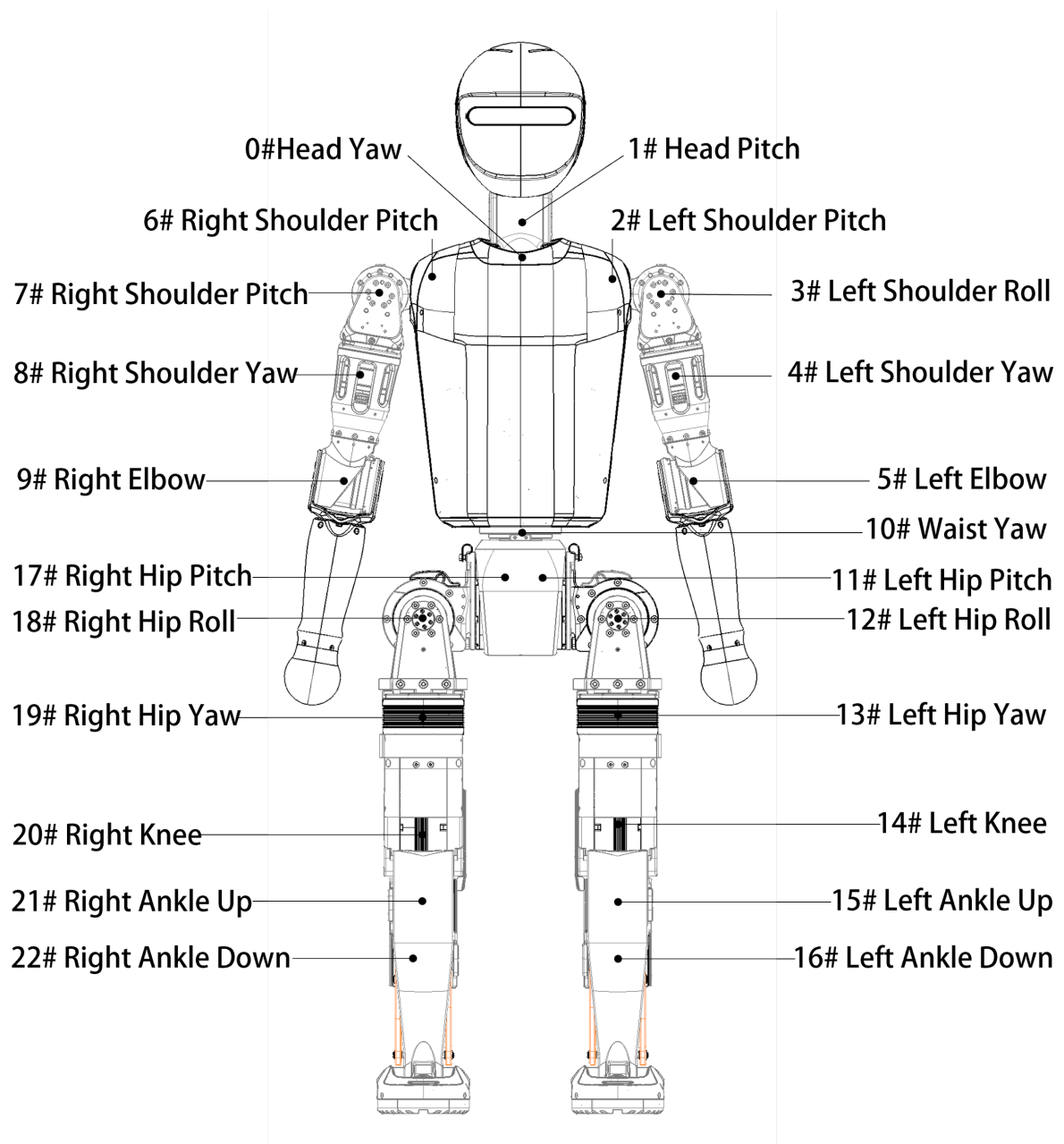
## Parties principales

### Moteurs d'articulations.

#### Identification commune et limites.

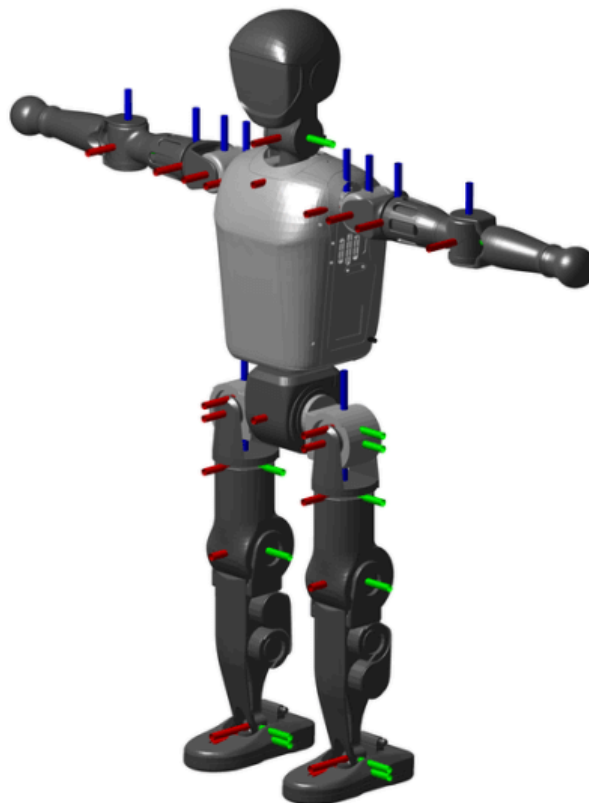
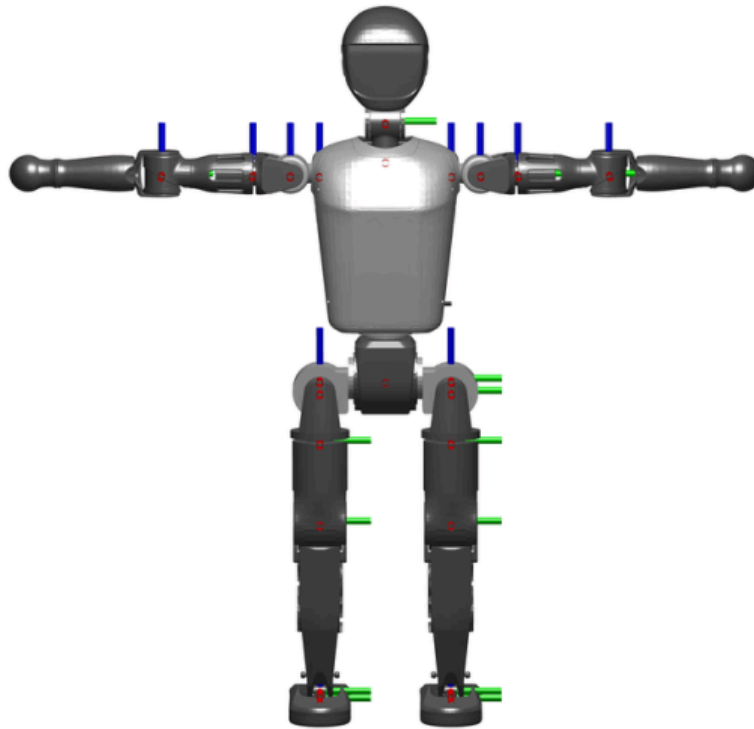
Identification du moteur	Nom conjoint	Limite (degrés)	
		Max	Min
0	Articulation de lacet de la tête	58	-58
1	Articulation de tangage de tête	47	-18
2	Articulation de tangage de l'épaule gauche	68	-188
3	Articulation de roulis de l'épaule gauche	88	-94
4	Articulation de lacet de l'épaule gauche	128	-128
5	Articulation du coude gauche	2	-120
6	Articulation de tangage de l'épaule droite	68	-188
7	Articulation de roulis de l'épaule droite	94	-88

8	Articulation de lacet de l'épaule droite	128	-128
9	Articulation du coude droit	120	-2
10	Articulation de lacet de la taille	58	-58
11	Articulation de tangage de la hanche gauche	118	-118
12	Articulation de roulis de la hanche gauche	88	-21
13	Articulation de lacet de la hanche gauche	58	-58
14	Articulation du genou gauche	123	0
15	Cheville gauche (articulation supérieure)	49	-23
16	Cheville gauche (articulation inférieure)	45	-24
17	Articulation de tangage de la hanche droite	118	-118
18	Articulation de roulis de la hanche droite	21	-88
19	Articulation de lacet de la hanche droite	58	-58
20	Articulation du genou droit	123	0
21	Cheville droite (articulation supérieure)	49	-23
22	Cheville droite (articulation inférieure)	45	-24



### Système de coordonnées

Le système de coordonnées articulaires, avec toutes les articulations en position zéro, est représenté dans le schéma ci-dessous.



## Contrôleur

Version	Basique	Standard	
	Carte de base	carte de commande de mouvement	Tableau de perception
Processeur	Jetson AGX Orin 32 Go	Express-i7	Jetson AGX Orin 32 Go
performances de calcul	Processeur Cortex-A78AE à 8 cœurs cadencé à 2GHz Cœurs Tensor GPU à 930MHz Performances de l'IA : 200 TOPS	6 cœurs P + 8 cœurs E Fréquence maximale du cœur P : 4,80 GHz Fréquence maximale du cœur E : 3,90 GHz	Processeur Cortex-A78AE à 8 cœurs cadencé à 2GHz Cœurs Tensor GPU à 930MHz Performances de l'IA : 200 TOPS
Mémoire	32 Go	8 Go	32 Go
Stockage	512 Go	512 Go	512 Go
Réseau câblé	1000M*2	1000M*2	1000M*2
Réseau sans fil	WIFI6*1	WIFI6*1	WIFI6*1
Audio	Microphone, haut-parleur		Microphone, haut-parleur

💡 Dans la version de base, le programme de contrôle de mouvement et le programme de perception sont tous deux déployés sur la carte de base.

💡 Dans la version standard, le programme de contrôle de mouvement est déployé sur la carte de contrôle de mouvement, et le programme de perception est déployé sur la carte de perception.

# Manuel d'utilisation

Le T1 est livré avec un programme de contrôle de mouvement prêt à l'emploi. Suivez les instructions ci-dessous pour contrôler le T1 à distance.

## AVERTISSEMENTS !

1. Le T1 doit d'abord entrer en mode PREP, puis **être placé dans une position debout stable au sol**, avant de passer en mode WALK.
2. **NE PAS soulever T1 en mode MARCHÉ.**
3. **Veillez à dégager tout obstacle au sol et à éviter les blessures humaines lors de l'opération T1.**
4. **NE TOUCHEZ AUCUNE PARTIE DU T1 EN MODE MARCHÉ, À L'EXCEPTION DE LA POIGNÉE.**
5. **Assurez-vous d'enlever TOUTES les pièces de calibration avant de redémarrer le T1.**

## MODES

- Le T1 fonctionne par **mode**.
- Le T1 peut effectuer différentes **actions** selon **les modes**.
- Il est possible de passer d'un **mode** à l'autre, mais **avec certaines contraintes**. Par exemple, le mode DAMP ne peut passer en mode WALK qu'après être entré en mode PREP.

### Mode DAMP

- L'alimentation est activée et la commande de mouvement fonctionne.
- Toutes les articulations sont dans un état d'amortissement, c'est-à-dire qu'elles résisteront aux changements de position, mais ne tenteront ni de maintenir leur position ni de la modifier activement.
- Le T1 ne peut PAS rester en position debout en mode DAMP, un support est donc nécessaire.
- Le mode DAMP est un mode de sécurité ; l'activation du mode DAMP permet de protéger T1 et son opérateur.
- Le mode DAMP peut être basculé vers le mode PREP, mais pas directement vers le mode WALK.

### Mode PRÉPARATION

- L'alimentation est activée et la commande de mouvement fonctionne.
- T1 adopte une posture debout et la maintient. Les articulations résistent fortement au changement de position et tentent de revenir à la posture debout si on les déplace de force.
- En mode PREP, le T1 peut tenir debout tout seul au sol. Dans ce mode, il peut être placé de manière à tenir debout au sol. Cependant, il ne tentera pas de se stabiliser.




- Le T1 peut passer à TOUS les autres modes en mode PREP, y compris DAMP et WALK.

## Mode MARCHÉ

- L'alimentation est activée et la commande de mouvement fonctionne.
- En mode MARCHÉ, le T1 peut effectuer diverses actions prédéfinies, notamment la marche omnidirectionnelle, la rotation, la marche, l'immobilité et les mouvements de la tête.
- Comparé au mode PRÉPARATION, le mode MARCHÉ est plus résistant et tentera de retrouver son équilibre en cas de déséquilibre.
- Le T1 en mode MARCHÉ peut passer à tous les autres modes, y compris DAMP et PREP.
- **REMARQUE : Assurez-vous que le T1 est en mode PREP et qu'il est déjà bien stable au sol avant de passer en mode MARCHÉ.**

## Mode CUSTOM

- L'alimentation est activée et la commande de mouvement fonctionne.
  - **Le T1 cède le contrôle de toutes ses articulations au développeur, qui le contrôle via son SDK. Soyez prudent en mode CUSTOM afin d'éviter d'endommager T1.**
  - Seuls les modes PREP et DAMP peuvent passer en mode CUSTOM.
  - Le mode CUSTOM ne peut basculer qu'en mode PREP ou DAMP.
-  Lors de la mise au point de nouvelles figures sur T1, il est recommandé d'utiliser un Hoist en permanence en mode CUSTOM.

## Mode PROTÉGÉ

- Le mode PROTECT s'activera automatiquement en cas d'erreur (par exemple, dépassement de la limite articulaire ou chute).
- Les articulations en mode PROTECT se comportent de la même manière qu'en mode DAMP.
- Vous pouvez essayer de réactiver le mode DAMP en mode PROTECT. (Redémarrage logiciel)
- Le mode PROTECT est un mode de sécurité ; l'activation du mode PROTECT permet de protéger le T1 et son opérateur.

## Mise sous tension

1. Placez T1 sur son support.
2. Installez la batterie. Glissez-la dans son logement, les voyants orientés vers l'extérieur.
3. Placez les mains et les jambes de T1 dans une posture naturelle.
4. Appuyez sur le bouton Marche/Arrêt pendant environ 3 secondes (relâchez-le une fois le voyant allumé ; si vous maintenez la pression plus de 6 secondes, le robot s'éteindra). Le robot s'allume. Patientez environ une minute ; un signal sonore retentit. Vous pourrez alors le contrôler à distance. **Remarque : l'initialisation de l'IMU nécessite que le T1 reste immobile pendant le processus de démarrage.**
5. Appuyez sur **RT+Y** sur le joystick pour entrer en mode PREP, après quoi le T1 peut être placé au sol et mis en position debout.
6. Appuyez sur **RT+A** pour activer le mode MARCHÉ. Le T1 répondra alors aux commandes de marche. **REMARQUE : NE tentez PAS de soulever le T1 en mode MARCHÉ.**

## Arrêt

1. Appuyez sur **RT+Y** sur le joystick pour entrer en mode PREP, après quoi le T1 peut être soulevé et placé sur le support.
2. Appuyez sur **RT+X** pour entrer en mode DAMP, puis appuyez sur le bouton d'alimentation pour éteindre le T1.
- Après l'avoir éteint, vous devez attendre 6 secondes avant de pouvoir le rallumer.

## Commande par joystick

### Carte des clés

Deux types de joysticks différents seront expédiés de manière aléatoire, mais ils auront les mêmes fonctions.

Joystick 1 : Joystick compatible Xbox. **Assurez-vous que le mode de contrôle du joystick est réglé sur D (Direct) et que le voyant MODE est éteint.**



Le mode de contrôle doit être en mode direct  
être éteint.

. Le voyant doit

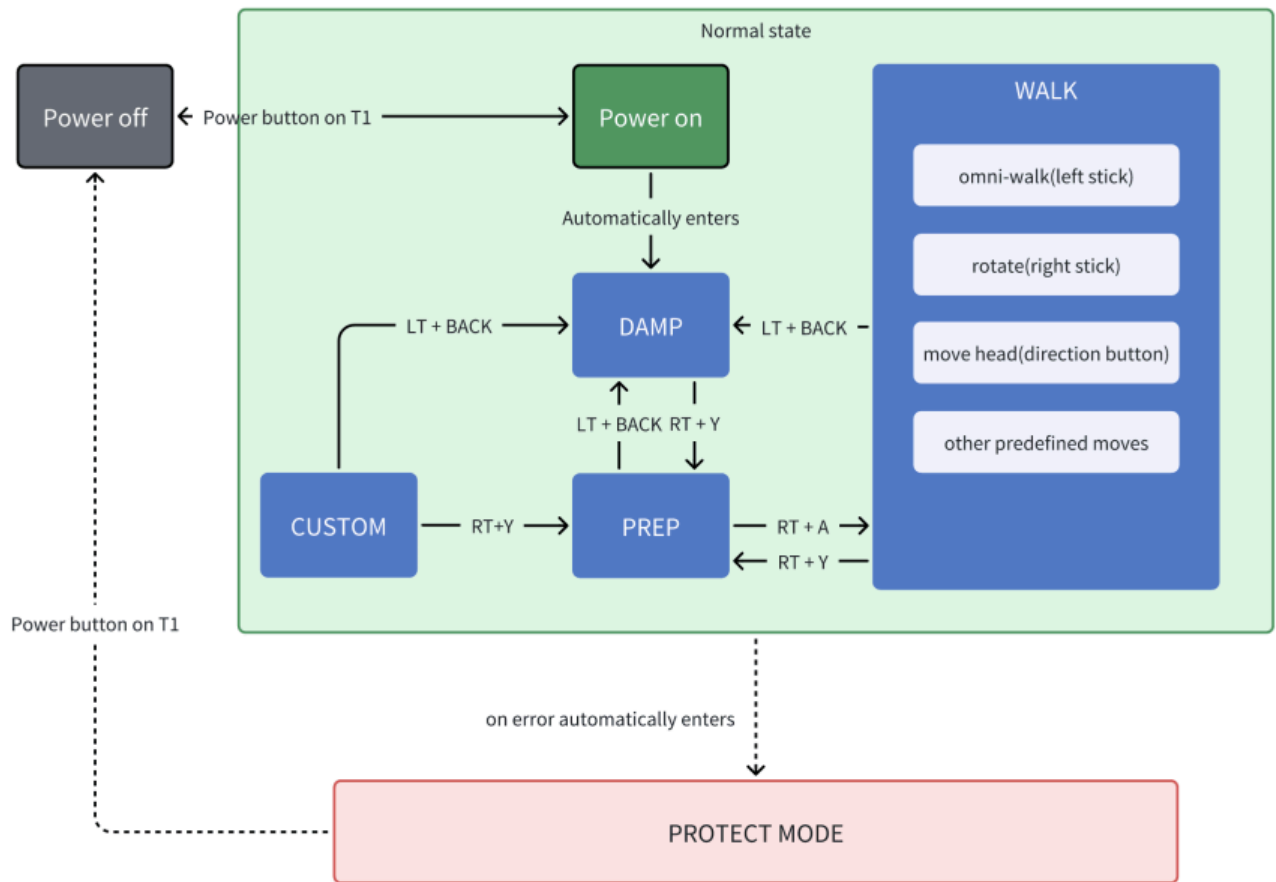
Joystick 2 : Joystick compatible Xbox. **Assurez-vous que le mode de contrôle est réglé sur « Mode récepteur » (logo cadenas) et que 3 LED sont allumées.**



Boutons	Fonctionnalité	Prérequis	Version effective ( <b>Valable pour toutes les versions, sauf indication contraire</b> )
LT + BACK	Entrer en mode DAMP		>= v1.0.6
RT + X	Entrer en mode DAMP		< v1.0.6
RT + Y	Entrer en mode PREP		
RT + A	Passer en mode WALK	En mode PREP	
Joystick gauche	Contrôle de la locomotion	En mode WALK	
Joystick droit	Contrôle de rotation	En mode WALK	
Boutons de direction	Rotation de la tête	En mode WALK	
RT + RB	Marche/Arrêt du mode Contrôle de l'effecteur manuel	En mode WALK	
RB + X	Bras ouverts	En mode Contrôle de l'effecteur manuel	
RB + Y	Bras serrés	En mode Contrôle de l'effecteur manuel	
LB + A	danse du Nouvel An	En mode Contrôle de l'effecteur manuel	>= v1.0.6
LB + GAUCHE	Danse rock n roll	En mode Contrôle de l'effecteur manuel	>= v1.1.0
LB + DROITE	Danse « Vers l'avenir »	En mode Contrôle de l'effecteur manuel	>= 1.2.0.8
LB + B	geste de Pogba	En mode Contrôle de l'effecteur manuel	>= 1.2.0.8
LB + X	geste d'Ultraman	En mode Contrôle de l'effecteur manuel	>= 1.2.0.8
LB + Y	geste de salutation chinois	En mode Contrôle de l'effecteur manuel	>= 1.2.0.8

LB + HAUT	geste d'applaudissement	En mode Contrôle de l'effecteur manuel	>= 1.2.0.8
LB + BAS	geste maneki	En mode Contrôle de l'effecteur manuel	>= 1.2.0.8
LT + DROITE	se lever	En mode PREP	>= v1.1.0
LT + RT + X	Saisir l'ancienne version de la démarche (démarche antérieure à la v1.1.0)	En mode PREP	
LT + RT + A	démarche de footballeur	En mode PREP	>= 1.2.0.8
LT + RT + START	coup de pied	En mode PREP / WALK	>= 1.2.0.8
LB + RB + UP	Démarrer/Arrêter l'interaction vocale en temps réel		>= 1.2.0.8
LB + LT + A	Démarrer/Arrêter la salutation proactive (le robot s'approche d'une personne détectée et dit bonjour)	Sous interaction vocale en temps réel	>= 1.2.0.8
B	début/arrêt, signe de la main	En mode WALK	
UN	début/arrêt de la poignée de main	En mode WALK	

## Carte des États



## Arrêt d'urgence

- En cas d'urgence, comme une chute ou une collision imminente, ou si les moteurs dysfonctionnent, un arrêt d'urgence est nécessaire.
- Le bouton d'arrêt d'urgence possède deux états : **enfoncé** et **relâché** . Une simple pression active l'arrêt d'urgence, une seconde le désactive.
- **Si l'arrêt d'urgence est activé, le robot doit être redémarré (soit en redémarrant le service de contrôle du robot, soit en redémarrant à nouveau le robot) pour continuer.**





# Se connecter au robot

## Se connecter via l'application

1. [Téléchargez et installez l'application.](#)

## Se connecter via le terminal

### Connexion filaire

1. Connectez-vous via Ethernet et configurez le réseau câblé en mode manuel comme suit

```
Shell  
adresse : 192.168.10.10  
masque de sous-réseau : 255.255.255.0  
porte: 192.168.10.1
```

### Étapes de configuration d'une adresse IP statique (Référence)

- Mac : <https://www.macinstruct.com/tutorials/how-to-set-a-static-ip-address-on-a-mac/>
  - Windows :  
<https://www.trendnet.com/press/resource-library/how-to-set-static-ip-address>
  - Linux :  
<https://www.freecodecamp.org/news/setting-a-static-ip-in-ubuntu-linux-ip-address-tutorial/>
2. Connectez la machine de développement au robot à l'aide d'un câble Ethernet. Le port réseau du robot est indiqué sur l'image.





### 3. Connexion via SSH

#### Basique

- Connexion au tableau de base

```
Shell
# Connexion filaire, carte de connexion de base
ssh booster@ 192 . 168 . 10 . 102
# Mot de passe initial : 123456
```

#### Standard

Carte de contrôle de mouvement de connexion

```
Shell
# Connexion filaire, carte de contrôle de mouvement de connexion
ssh maître@ 192 . 168 . 10 . 101
# Mot de passe initial : 123456
```

Tableau de perception de connexion

```
Shell
# Connexion filaire, carte de perception de connexion
ssh booster@ 192 . 168 . 10 . 102
# Mot de passe initial : 123456
```

#### Connexion sans fil

1. Configurez la connexion sans fil du robot via l'[Application](#). Trouvez l'adresse IP sans fil du robot dans l'application, par exemple xxx.xxx.xxx.xxx
2. Connexion via ssh.

#### Basique

Connexion au tableau de base

```
Shell
# Connexion sans fil, carte de connexion de base
ssh booster@xxx.xxx.xxx.xxx
# Mot de passe initial : 123456
```

## Standard

### Carte de contrôle de mouvement de connexion

Shell

```
# Connexion sans fil, carte de commande de mouvement  
ssh maître@xxx.xxx.xxx.xxx  
# Mot de passe initial : 123456
```

- Tableau de perception de connexion
  - Si l'adresse IP sans fil de la carte de perception est inconnue, il est possible de se connecter d'abord à la carte de contrôle de mouvement, puis d'établir la connexion avec la carte de perception via cette dernière.

Shell

```
# 1. Connectez-vous sans fil à la carte de commande de mouvement  
ssh maître@xxx.xxx.xxx.xxx  
# Mot de passe initial : 123456  
  
# 2. Connexion au tableau de bord de perception via une connexion interne  
ssh booster@ 192 . 168 . 10 . 102  
# Mot de passe initial : 123456
```

- L'adresse IP sans fil de la carte de perception est déjà connue.

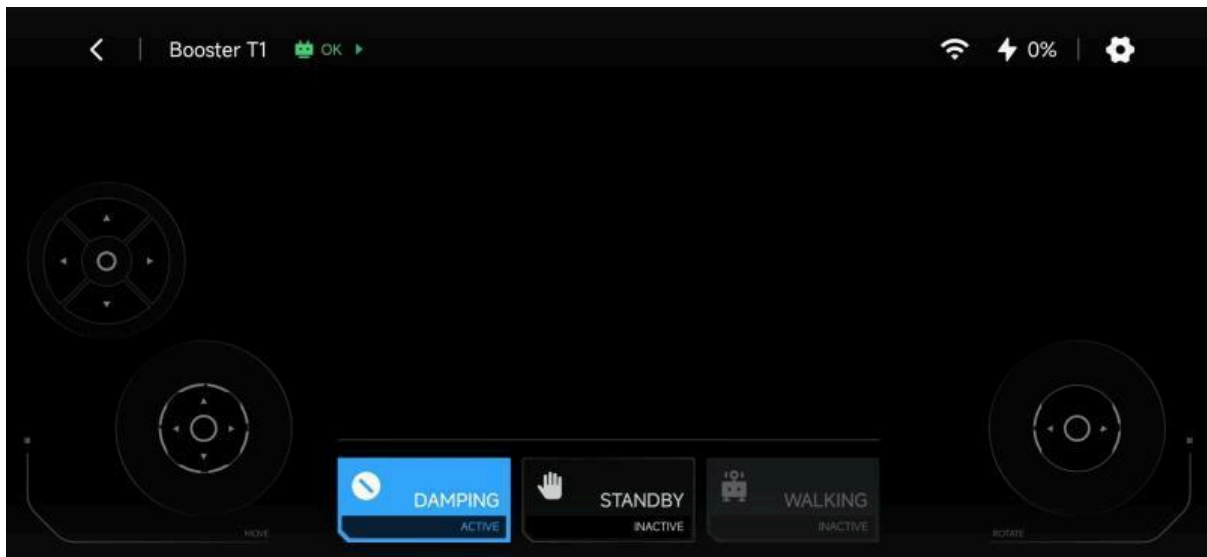
Shell

```
# Se connecter sans fil au tableau de perception  
booster ssh@xxx.xxx.xxx.xxx  
# Mot de passe initial : 123456
```

# Service de contrôle de robots : démarrage/arrêt

## Via l'application

1. Appuyez sur le bouton d'affichage du statut comme indiqué ci-dessous



2. Appuyez sur « REDÉMARRER LE ROBOT » comme indiqué ci-dessous

The screenshot shows the 'Status' screen of the application. At the top, there is a back arrow, the title 'Status', and status indicators for warnings (0) and errors (0), along with a red 'RESTART ROBOT' button. Below this is a table with four columns: 'Joint', 'Connection', 'Temperature', and 'Limit'. The table lists several joints and their corresponding connection status, temperature, and limit status.

Joint	Connection	Temperature	Limit
Joint_0_1	✓	25°C	✓
Joint_0_2	✓	25°C	✓
Joint_10_2	✓	26°C	✓
Joint_10_1	✓	25°C	✓
Joint_11_2	✓	24°C	✓
Joint_11_1	✓	24°C	✓
Joint_13_2	✓	25°C	✓
Joint_13_1	✓	24°C	✓

## Via le terminal

1. [Se connecter au robot](#)
2. Exécutez la commande ci-dessous

Avant d'exécuter la commande ci-dessous, assurez-vous que le robot est correctement posé au sol ou sur son support.

Shell

```
# démarrer le service de contrôle du robot
```

```
booster-cli lancement -c commencer
```

```
# Arrêter le service de contrôle des robots
```

```
booster-cli lancement -c arrêt
```

```
# redémarrer le service de contrôle du robot
```

```
booster-cli lancement -c redémarrage
```

# Chargement

1. Insérez la prise de charge dans le sens indiqué.



2. L'état de charge est indiqué comme suit : le voyant vert indique une charge complète et le voyant rouge indique une charge en cours.



## Étalonnage du zéro !

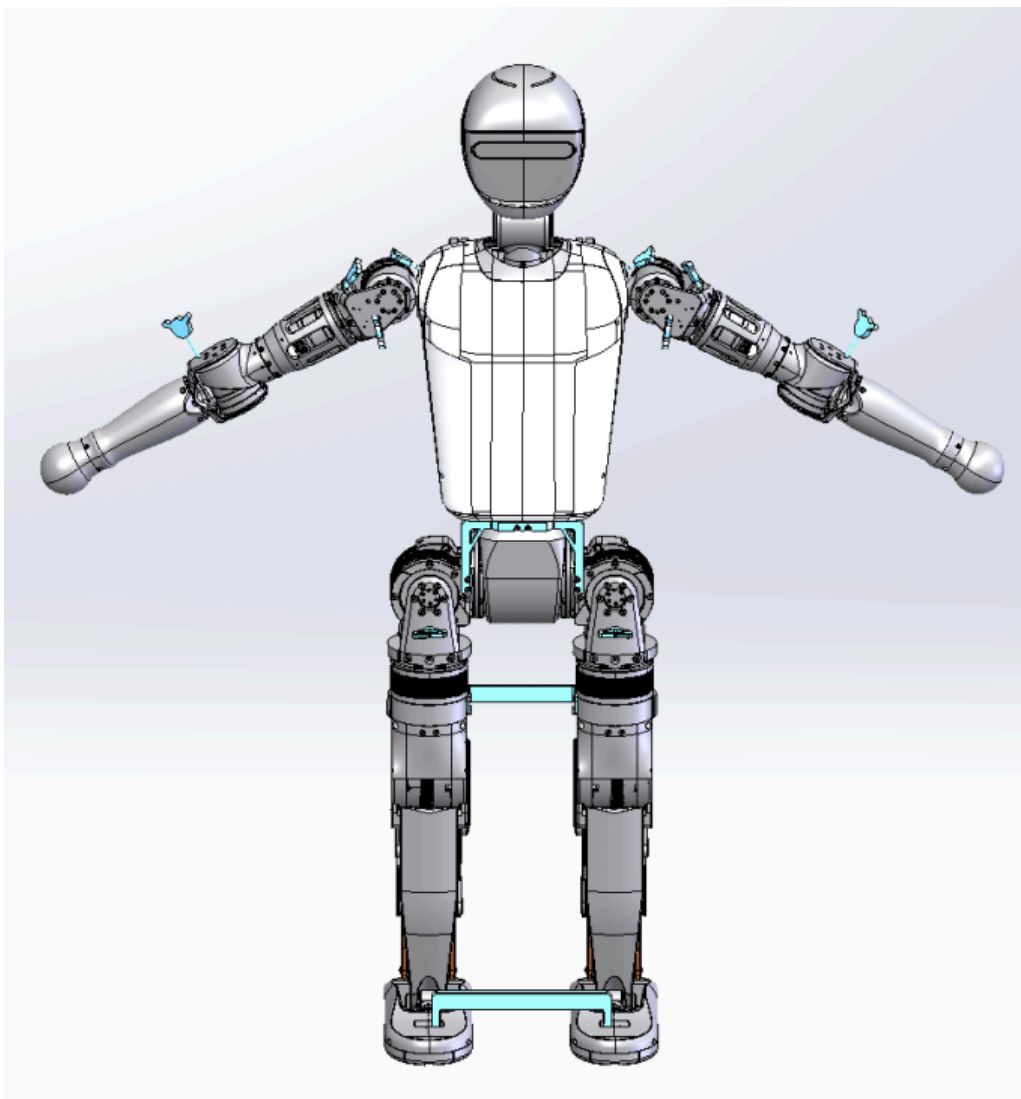
1. Avant de procéder à l'étalonnage du zéro, **veuillez d'abord arrêter le service de fonctionnement du robot**. Vous pouvez utiliser l'application ou le terminal pour le contrôler. Pour plus de détails, veuillez consulter la section [« Connexion au robot »](#).
2. Une fois l'étalonnage du zéro effectué, **veuillez vous assurer de démonter toutes les pièces de calibration afin d'éviter que le moteur ne cale et ne soit endommagé**.

L'ensemble du processus d'étalonnage du zéro est divisé en quatre étapes :

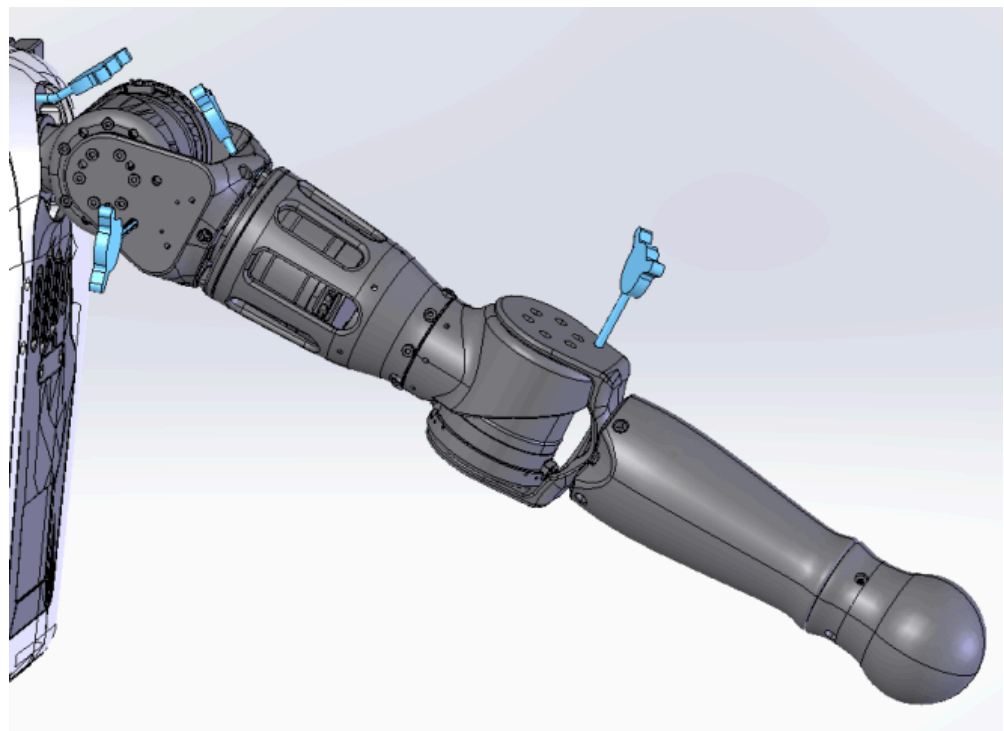
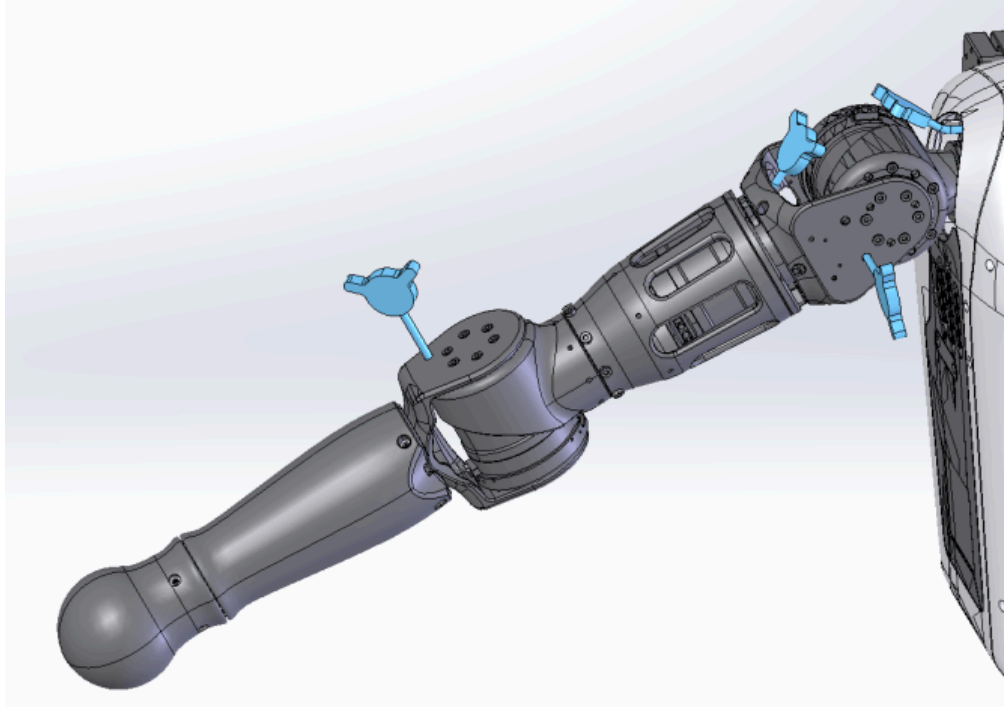
1. **Installez les pièces mécaniques pour l'étalonnage du zéro ;**
2. **Exécutez le programme d'étalonnage du zéro ;**
3. **Démontez les pièces zéro pour l'étalonnage du zéro .**
4. **Confirmer la position zéro**

### Installez les pièces zéro

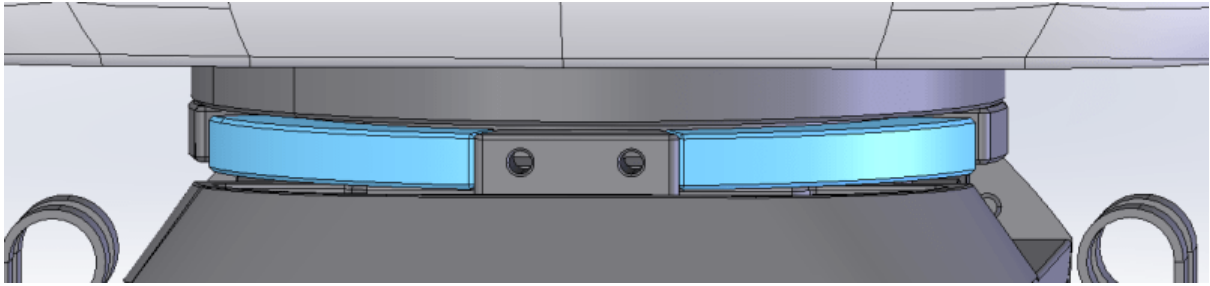
Installez les pièces zéro marquées en bleu sur le robot conformément à la figure :



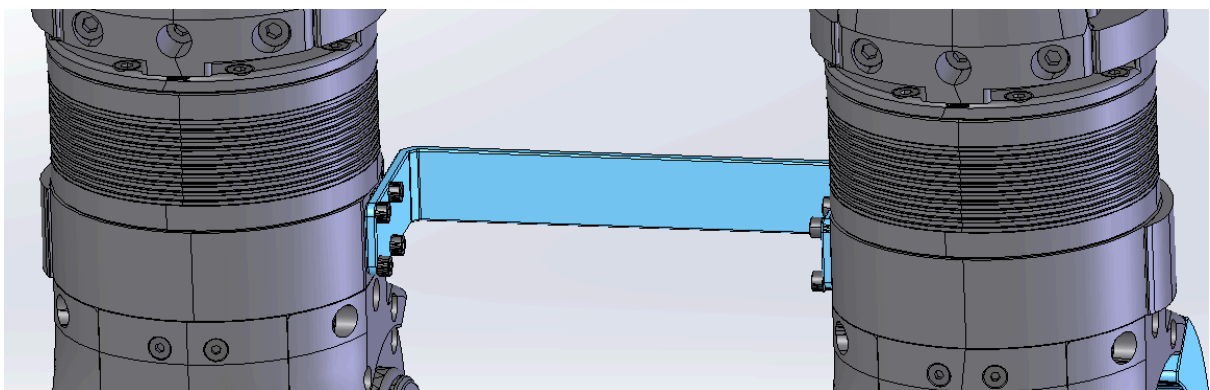
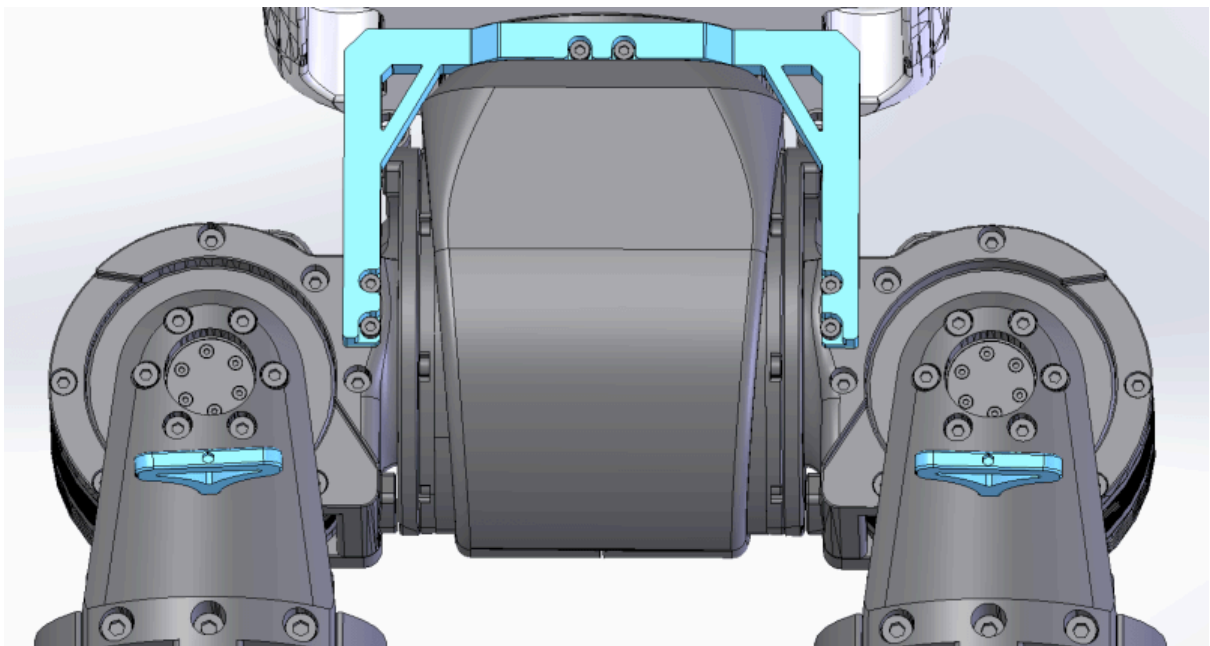
1. Bras : chaque bras possède 4 degrés de liberté, avec un axe de 4 mm de diamètre pour le calibrage du zéro. Les articulations de tangage et de lacet de l'épaule utilisent des axes courbes, tandis que les articulations de tangage et de coude utilisent des axes droits.



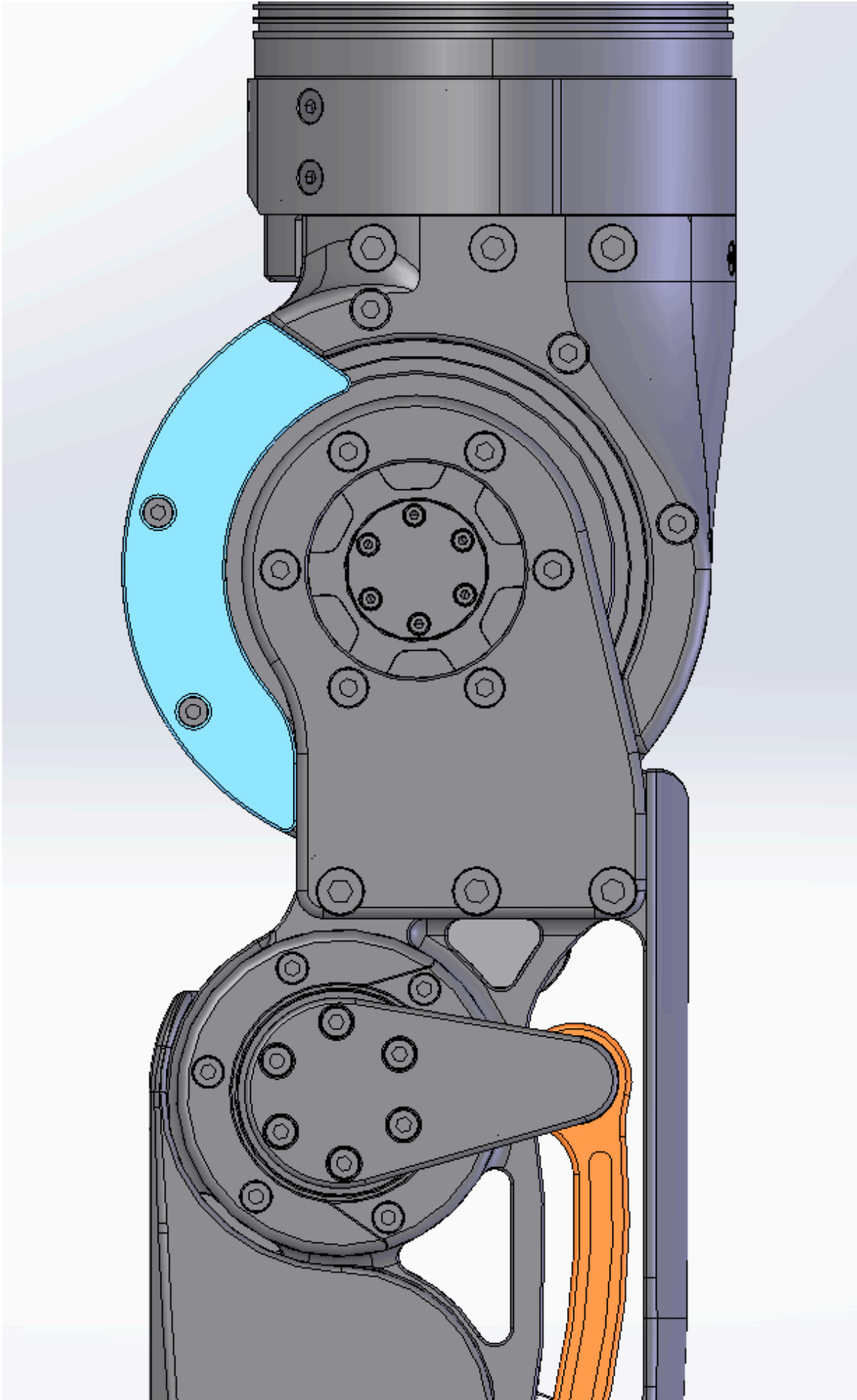
2. Tour de taille : 1 degré de liberté, avec deux butées pour le calibrage du zéro.

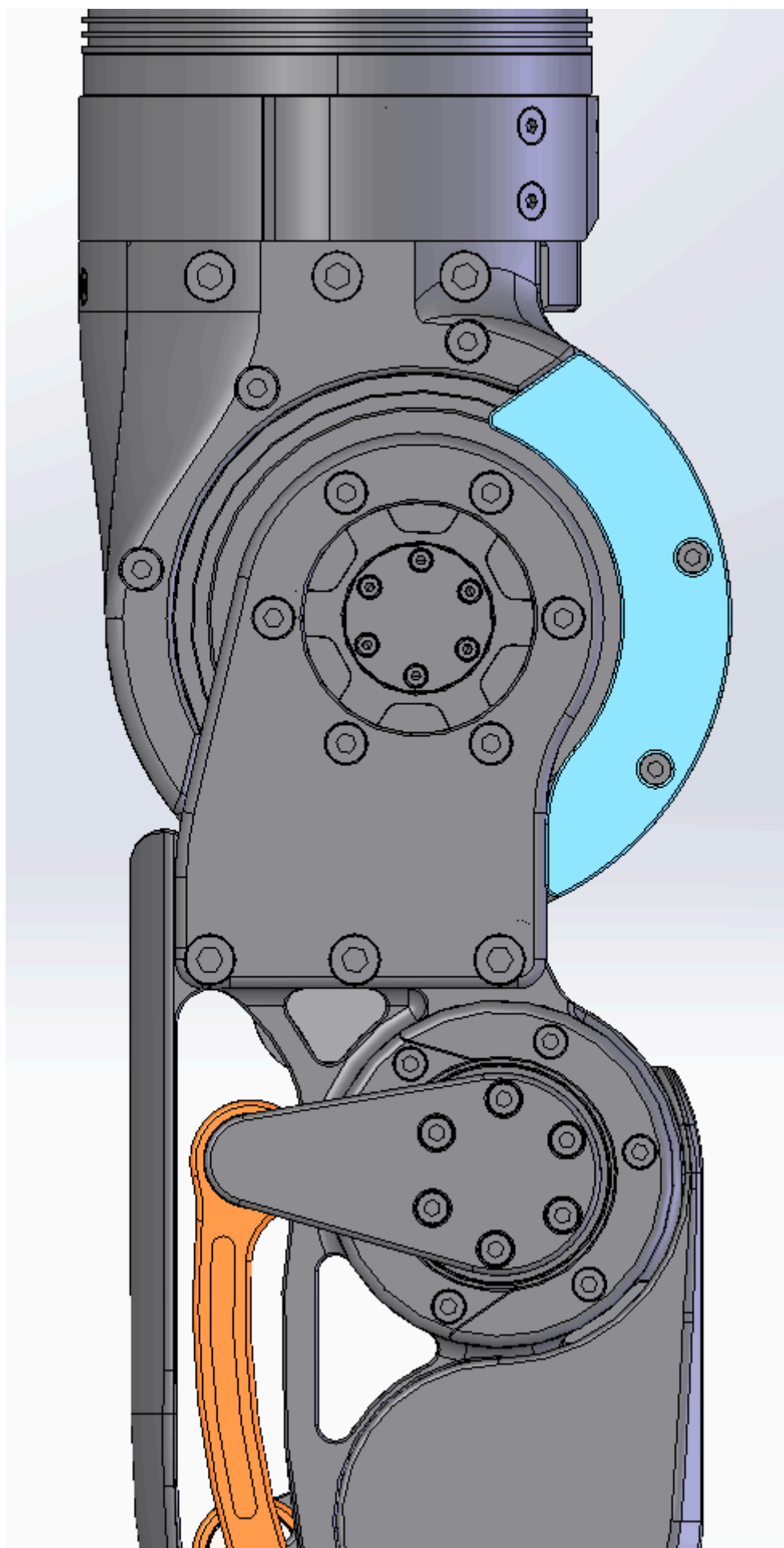


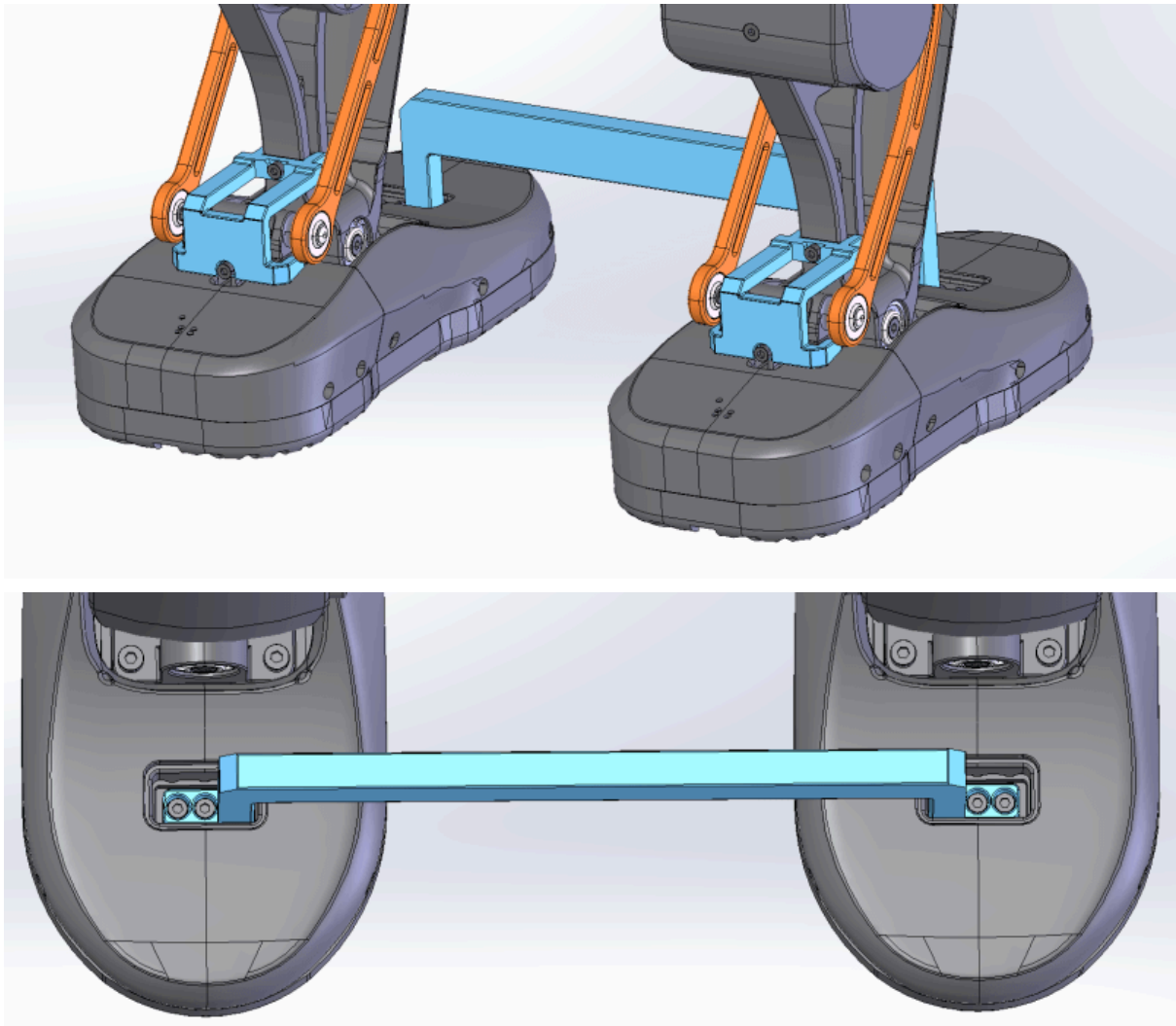
3. Jambes : chaque jambe possède 6 degrés de liberté et 9 points de réglage pour l'étalonnage du zéro. Les vis requises sont à tête cylindrique hexagonale M3×8. Retirez le bouchon en caoutchouc de la semelle avant l'étalonnage du zéro du pied.











## Exécution du programme d'étalonnage zéro

### Basique

1. [Connectez-vous au robot](#) sur la carte de base.
2. [Arrêter le service de contrôle des robots](#).

Exécution du programme d'étalonnage zéro

Shell

```
sudo /opt/booster/Tools/encos_reset_zero_pos_v2 tous
```

### Standard

1. [Connectez-vous au robot](#) via la carte de contrôle de mouvement.
2. [Arrêter le service de contrôle des robots](#).
3. Exécution du programme d'étalonnage zéro

Shell

```
sudo /opt/booster/Tools/encos_reset_zero_pos_v2 tous
```

## Suppression des pièces de calibration

Consultez les instructions d'installation et retirez toutes les pièces dans l'ordre inverse. **Assurez-vous que toutes les pièces sont retirées avant de démarrer la commande du robot afin d'éviter tout blocage des articulations.**

## Confirmation de la position zéro

1. [Se connecter au robot](#).
2. [Démarrer le service de contrôle du robot](#).
3. Utilisez la télécommande pour entrer en **mode PREP** et vérifiez si la posture debout du robot est normale.

# Aide

Notre société collectera, sans enfreindre aucune loi ni réglementation applicable ni porter atteinte à la vie privée, uniquement les données relatives au fonctionnement des robots à des fins de détection des pannes et d'analyse statistique pertinente.

## Vérifier la version

### Basique

Après la connexion à la carte de base, vous pouvez exécuter la commande suivante pour vérifier la version système actuelle du robot.

Shell

```
chat /opt/booster/version.txt
```

### Standard

La carte de contrôle de mouvement et la carte de perception possèdent chacune leur propre version. Après connexion à la carte correspondante, vous pouvez exécuter la commande suivante pour vérifier la version système actuelle du robot.

Shell

```
chat /opt/booster/version.txt
```

exemple:

```
booster@tegra-ubuntu ~ (0.156s)
cat /opt/booster/version.txt
-----
Version: v1.0.6.2-release
Branch: release/v1.0.6-20250212
Commit ID: 648dc0375029fae7b96ea776b6df0cb90a8edf24
Install time: Tue Feb 18 14:43:02 CST 2025
```

## Mise à jour logicielle !

- Lors de l'installation d'une nouvelle version du logiciel, le programme de contrôle de mouvement du robot sera interrompu et ses articulations ne seront plus actionnées.  
**Avant la mise à jour, assurez-vous que le robot est en mode DAMP ou que son contrôle de mouvement est désactivé et qu'il bénéficie d'un support stable.**
- Assurez-vous que le robot est connecté au réseau.

### Mise à niveau avec package

Pour les versions antérieures, veuillez vous référer à [Historique des versions T1](#)  
Téléchargez les “Basic and Standard Software Installation Packages”

Date de sortie	Basique	Standard		Mises à jour
		Carte de commande de mouvement	Tableau de perception	
29/08/2025	v1.2.1.8	v1.2.1.6	v1.2.1.7	<b>Nouvelles fonctionnalités :</b> <ul style="list-style-type: none"><li>• Assistance pour l'édition de base T1</li><li>• Fréquence d'images visuelle améliorée</li><li>• Ajout de messages de récupération en cas d'échec de la mise à niveau.</li></ul> <b>Corrections de bugs :</b> <ul style="list-style-type: none"><li>• Correction de plusieurs plantages système occasionnels</li><li>• Amélioration de la stabilité des coups de pied</li><li>• Limites optimisées de l'articulation de la tête</li></ul>
	<a href="https://obs-cdn.boosterobotics.com/ota_single/v1.2.1.8-release-single-aarch64.run">https://obs-cdn.boosterobotics.com/ota_single/v1.2.1.8-release-single-aarch64.run</a>	<a href="https://obs-cdn.boosterobotics.com/ota_motion/v1.2.1.6-release-motion-x86_64.run">https://obs-cdn.boosterobotics.com/ota_motion/v1.2.1.6-release-motion-x86_64.run</a>	<a href="https://obs-cdn.boosterobotics.com/ota_perception/v1.2.1.7-release-perception-aarch64.run">https://obs-cdn.boosterobotics.com/ota_perception/v1.2.1.7-release-perception-aarch64.run</a>	

### Mise à niveau du logiciel pour la version de base

Commencez par définir les permissions d'exécution des fichiers : (Remarque : veuillez remplacer les noms de fichiers suivants par les noms de fichiers réels.)

Shell

```
sudo chmod +x v1.1.1.0-release-aarch64.run
```

Copiez le package de mise à jour logicielle sur la carte mère (il peut être copié dans `/home/master/Downloads` ), puis exécutez :

CSharp

```
./v1. 1 . 1 . 0-release-aarch64 .run
```

### Mise à jour du logiciel de la carte de contrôle de mouvement

Commencez par définir les autorisations d'exécution du fichier :

Shell

```
sudo chmod +x v1.0.1.30-release-T1-motion-x86_64.run
```

Copiez le package de mise à jour logicielle sur la carte de contrôle de mouvement (vous pouvez le copier dans `/home/master/Downloads` ), puis exécutez :

Shell

```
./v1. 0 . 1 . 30-release-T1-motion-x86_64 .run
```

### Mise à jour du logiciel du tableau de perception

Commencez par définir les autorisations d'exécution du fichier :

Shell

```
sudo chmod +x v1.0.1.30-release-perception-aarch64.run
```

Copiez le package de mise à jour logicielle sur la carte de perception (vous pouvez le copier dans `/home/booster/Downloads` ), puis exécutez :

Shell

```
./v1.0.1.30-release-perception-aarch64.run
```

## Mise à niveau en ligne de commande

Les versions supérieures à la v1.0.5.0 permettent la mise à niveau vers la dernière version du système via lignes de commande. La carte de contrôle de mouvement et la carte de perception doivent être mises à niveau individuellement.

### Mise à niveau du logiciel pour la version de base

- Exécutez cette commande sur un tableau de base

Shell

```
booster -cli upgrade
```

### Mise à niveau du logiciel pour la version standard

- Exécutez cette commande sur la carte de contrôle de mouvement ou la carte de perception.

Shell

```
booster -cli upgrade
```

- L'affichage du contenu imprimé suivant indique que la mise à jour du système est terminée.

```
master@master ~ (28.010s)
booster-cli upgrade
Delete old files ...
Install new files ...
model Booster_T1 version 2.3.4
install config /tmp/selfgz172558/booster/configs/Booster_T1/T1_2.3.4
-----
Version: v1.0.6.3-release
Branch: release/v1.0.6-20250212
Commit ID: 51376c91cb673a432a115a7784c9754d27189a28
Install time: 2025年 02月 27日 星期四 11:06:08 CST
Created symlink /etc/systemd/system/multi-user.target.wants/booster-daemon.service → /etc/systemd/system/booster-daemon.service.
Latest version installed, please enjoy booster robot !!!
```

## Remarques concernant la mise à niveau !

- Si vous rencontrez un journal indiquant que **apt update** a échoué comme indiqué dans la figure ci-dessous, exécutez les commandes suivantes pour changer la source du logiciel à l'aide de l'outil fishros, puis essayez de mettre à niveau à nouveau.

```
Ign:2 https://mirrors.ustc.edu.cn/ubuntu-ports jammy/universe arm64 libatlas-base-dev arm64 3.10.3-12ubuntu1
Ign:1 https://mirrors.ustc.edu.cn/ubuntu-ports jammy/universe arm64 libatlas3-base arm64 3.10.3-12ubuntu1
Ign:2 https://mirrors.ustc.edu.cn/ubuntu-ports jammy/universe arm64 libatlas-base-dev arm64 3.10.3-12ubuntu1
Err:1 https://mirrors.ustc.edu.cn/ubuntu-ports jammy/universe arm64 libatlas3-base arm64 3.10.3-12ubuntu1
Temporary failure resolving 'mirrors.ustc.edu.cn'
Err:2 https://mirrors.ustc.edu.cn/ubuntu-ports jammy/universe arm64 libatlas-base-dev arm64 3.10.3-12ubuntu1
Temporary failure resolving 'mirrors.ustc.edu.cn'
E: Failed to fetch https://mirrors.ustc.edu.cn/ubuntu-ports/pool/universe/a/atlas/libatlas3-base_3.10.3-12ubuntu1_arm64.deb Temporary failure resolving 'mirrors.ustc.edu.cn'
E: Failed to fetch https://mirrors.ustc.edu.cn/ubuntu-ports/pool/universe/a/atlas/libatlas-base-dev_3.10.3-12ubuntu1_arm64.deb Temporary failure resolving 'mirrors.ustc.edu.cn'
E: Unable to fetch some archives, maybe run apt-get update or try with --fix-missing?
ERROR: Failed to install libatlas-base-dev! Some features may not work properly.
Please check your network connection and try again.
Error: Install perception external dependencies failed. Aborting the sequence.
booster@tegra-ubuntu:~/Documents/recovery$ cat /opt/booster/version.txt
.....
Version: v1.0.6.2-release
Branch: release/v1.0.6-20250212
Commit ID: 648dc375029fae7b96ea77b6d0cb90a8edf24
Install time: Tue Feb 18 14:43:02 CST 2025
.....
```

Shell

Il est recommandé d'utiliser *fishros* pour changer la **source**.

【 5 】 Change sources with one click

【 1 】 Change only system sources

【 1 】 Add ROS/ROS2 sources

**wget http://fishros.com/install -O fishros && . fishros**

## Restaurer les paramètres d'usine

### Introduction

Si le robot rencontre des problèmes suite à des modifications de configuration, essayez de restaurer les paramètres d'usine.

### Procédure

#### Basique

accédez à **/home/master/Booster/recovery** et exécutez :

Shell

**cd /home/booster/Booster/récupération**

**./v1.1.1.0-arm64.run**



## Standard

### Pour la récupération de la carte de commande de mouvement

- Accédez à `/home/master/Booster/recovery` et exécutez :

Shell

```
cd /home/master/Documents/recovery
./v1.0.1.30-release-T1-motion-x86_64.run
```

### Pour la récupération du tableau de perception

- Accédez à `/home/booster/Booster/recovery` et exécutez :

Shell

```
cd /home/booster/Documents/recovery
./v1.0.1.30-release-perception-aarch64.run
```

# Récupération des logs

## Introduction

En cas de problème avec le robot, le support technique peut avoir besoin de ses logs d'activité. Voici comment les récupérer et les transmettre au support.

## Procédure

### Basique

1. [Connectez-vous au robot via le terminal](#) de la carte de base.
2. Exécutez la commande pour obtenir le fichier journal compressé :

Veuillez noter que le fuseau horaire par défaut du robot est UTC+8. Si vous vous trouvez dans un fuseau horaire différent, vous pouvez soit modifier le fuseau horaire dans le système d'exploitation du robot, soit utiliser l'heure convertie pour accéder aux journaux.

Shell

```
# Exécutez cette commande dans le terminal
# Utilisation :
# -st --start-time [HEURE] : Filtrer les journaux à partir
d'une heure spécifique (format : AAAAMMJJ-HHMMSS, par
exemple : 20180808-080808). Cette option doit être activée.
```

```
# -et --end-time [HEURE] : Filtrer les journaux se terminant
avant une heure spécifique (format : AAAAMMJJ-HHMMSS, ex. :
20180808-080808). Si cette option n'est pas spécifiée,
utiliser 30000101-000000.
# -o, --output [FICHER] : Comprime les journaux dans un
fichier de sortie spécifié ; par défaut :
/home/master/Downloads/[AAAAMMJJ-HHMMSS].zip
booster-cli log -t AAAAMMJJ-HHMMSS -o CHEMIN_DE_SORTIE
```

Si le problème avec le robot s'est par exemple produit entre le 8 et le 10 août 2020, vous pouvez sélectionner une plage horaire d'environ dix minutes avant et après cette date comme heures de début et de fin pour la récupération des journaux, afin de garantir que le fichier de logs couvre la période du problème. Exécutez la commande suivante :

```
Shell
booster -cli log -st 20200808-120800 -et 20200808-120820 -o
/home/master/Documents
```

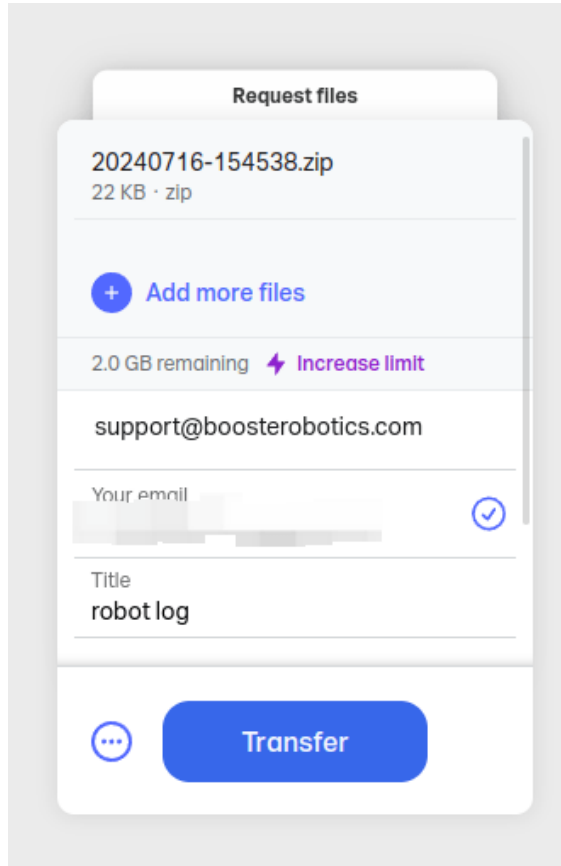
Après exécution, un fichier comme 20200808-120800.zip sera généré dans /home/master/Documents .

3. Envoyez le fichier généré au support technique via WeChat ou Feishu.

```
Shell
# Pour copier le fichier depuis le developer host (en
supposant que le fichier de logs se trouve dans
/home/master/Documents) :
scp
master@192.168.10.101:/home/master/Documents/20200808-120800.z
ip ~/Downloads

# Ensuite, envoyez le fichier journal au support technique
```

4. Si les utilisateurs situés à l'étranger ne peuvent pas envoyer de fichiers journaux via l'outil de chat, vous pouvez choisir d'utiliser [WeTransfer](#) pour envoyer les fichiers de logs.
  1. Téléchargez le fichier de logs, puis saisissez support@boosterobotics.com dans le champ « Adresse e-mail » et cliquez sur « Transférer » pour l'envoyer.



## Standard

1. [Connectez-vous au robot via le terminal](#) de la carte de contrôle de mouvement.
2. Exécutez la commande pour obtenir le fichier journal compressé :

Veuillez noter que le fuseau horaire par défaut du robot est UTC+8. Si vous vous trouvez dans un fuseau horaire différent, vous pouvez soit modifier le fuseau horaire dans le système d'exploitation du robot, soit utiliser l'heure convertie pour accéder aux journaux.

Shell

```
# Exécutez cette commande dans le terminal
# Utilisation :
# -st --start-time [HEURE] : Filtrer les journaux à partir
d'une heure spécifique (format : AAAAMMJJ-HHMMSS, par
exemple : 20180808-080808). Cette option doit être activée.
```

```
# -et --end-time [HEURE] : Filtrer les journaux se terminant
avant une heure spécifique (format : AAAAMMJJ-HHMMSS, ex. :
20180808-080808). Si cette option n'est pas spécifiée,
utiliser 30000101-000000.
# -o, --output [FICHER] : Comprime les journaux dans un
fichier de sortie spécifié ; par défaut :
/home/master/Downloads/[AAAAMMJJ-HHMMSS].zip

booster-cli log -t AAAAMMJJ-HHMMSS -o CHEMIN_DE_SORTIE
```

Si le problème avec le robot s'est produit entre le 8 et le 10 août 2020, vous pouvez sélectionner une plage horaire d'environ dix minutes avant et après cette date comme heures de début et de fin pour la récupération des journaux, afin de garantir que le paquet de journaux couvre la période du problème. Exécutez la commande suivante :

```
Shell
booster-cli log -st 20200808-120800 -et 20200808-120820 -o
/home/master/Documents
```

Après exécution, un fichier comme 20200808-120800.zip sera généré dans /home/master/Documents.

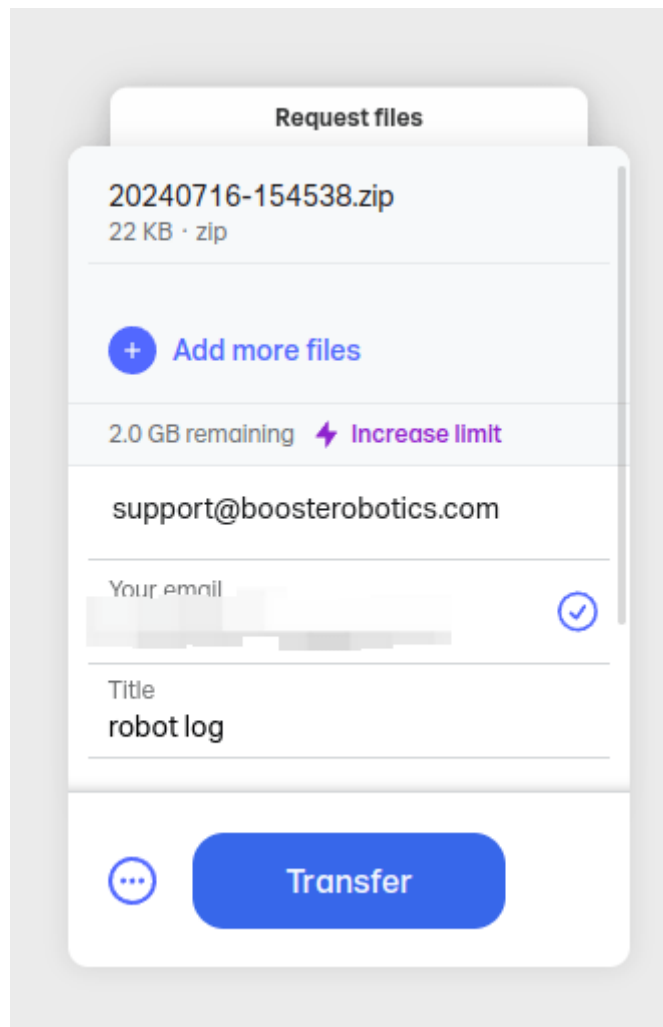
3. Envoyez le fichier généré au support technique via WeChat ou Feishu.

```
Shell
# Pour copier le fichier depuis l'hôte du développeur (en
supposant que le fichier de logs se trouve dans
/home/master/Documents) :
scp
master@192.168.10.101:/home/master/Documents/20200808-120800.z
ip ~/Téléchargements

# Ensuite, envoyez le fichier journal au support technique.
```

Si les utilisateurs situés à l'étranger ne peuvent pas envoyer de fichiers journaux via l'outil de chat, vous pouvez choisir d'utiliser [WeTransfer](#) pour envoyer les fichiers de logs.

1. Téléchargez le fichier de logs, puis saisissez support@boosterobotics.com dans le champ « Adresse e-mail » et cliquez sur « Transférer » pour l'envoyer.



The image shows a mobile app interface for WeTransfer. At the top, a card titled "Request files" displays a file named "20240716-154538.zip" with a size of "22 KB · zip". Below the file name is a blue circular button with a plus sign and the text "Add more files". A status bar indicates "2.0 GB remaining" with a purple lightning bolt icon and a link to "Increase limit". The recipient's email address, "support@boosterobotics.com", is entered in a text field. Below this, there is a section for "Your email" with a blurred input field and a blue checkmark icon. The "Title" field contains the text "robot log". At the bottom, there is a blue circular button with three dots and a large blue button labeled "Transfer".

# Assistance à distance

## Introduction

En cas de problème avec le robot et si vous avez besoin d'assistance à distance, suivez ces étapes pour obtenir du soutien.

## Procédure

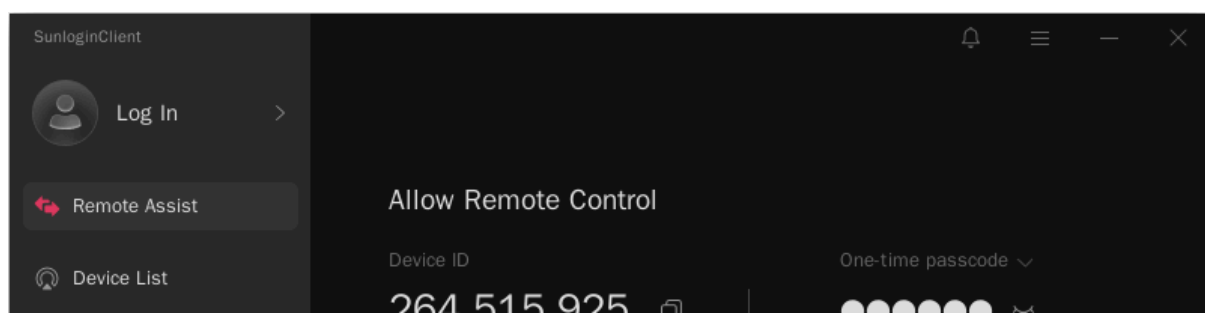


Vous pouvez choisir parmi les deux méthodes d'assistance à distance suivantes :

1. Connectez votre ordinateur personnel au robot via SSH ; le service d'assistance technique pourra alors se connecter à votre ordinateur via le bureau à distance, puis au robot.
2. Installez un logiciel de bureau à distance sur le robot ; vous pourrez utiliser un écran et un clavier/souris pour ouvrir ce logiciel, permettant ainsi au personnel d'assistance de se connecter directement au robot.
1. Téléchargez le logiciel de bureau à distance : <https://sunlogin.oray.com/download>
  - a. Choisissez l'édition personnelle
  - b. Sélectionnez le système d'exploitation correspondant à votre appareil, puis cliquez sur télécharger.



2. Installez le logiciel de bureau à distance et fournissez l'identifiant de votre appareil et votre mot de passe à usage unique au support technique.
  - a. Après l'installation, l'interface du logiciel apparaîtra comme suit :



- b. Cliquez sur l'icône en forme d'œil sous « Code d'accès unique » pour afficher le code. Copiez ensuite l'« ID de l'appareil » et le « Code d'accès unique » et transmettez-les à notre équipe d'assistance technique.
3. Une fois cette opération terminée, le support technique peut se connecter au robot via un bureau à distance et commencer le dépannage.

# Application mobile

## Introduction

Le robot Booster peut se connecter à une application mobile pour le contrôle, le retour d'information sur son état, la configuration Bluetooth et d'autres fonctionnalités.

## Lien de téléchargement :

- iOS (Scannez le code QR pour installer l'application Booster (compatible avec le firmware V1.2 et supérieur) via TestFlight.)

[Processus d'installation d'une application iOS](#)



- Androïde

<https://www.booster.tech/open-source/>

# Interaction vocale par IA

Dans la version **1.2.0.8** , le robot Booster intègre la fonctionnalité **de conversation vocale en temps réel de Doubao** . Vous pouvez activer ou désactiver cette fonctionnalité à l'aide du raccourci de la télécommande : **LB + RB + HAUT**.

## Instructions d'utilisation

Après avoir appuyé sur **LB + RB + HAUT** , veuillez patienter un instant jusqu'à entendre l'invite vocale :

*«Hello, I am Booster. How can I assist you ?»*

- Cela indique que la fonction d'interaction vocale en temps réel a été activée avec succès. Vous pouvez désormais converser avec le robot.
- Une fois l'interaction vocale en temps réel activée, le robot tournera automatiquement la tête vers le visage détecté le plus proche devant lui.
- Lorsque l'interaction vocale est active, appuyer sur **LB + LT + A** permettra au robot de s'approcher de la personne détectée et de lever la main pour la saluer.

## Configuration du dialogue

En modifiant la configuration, vous pouvez personnaliser la personnalité du robot et son message de bienvenue afin de permettre des interactions contextuelles et basées sur des scénarios.

### Chemin du fichier de configuration

#### Basique

Sur un tableau de base

```
Shell
/opt/booster/RTCCli/custom_setings.toml
```

#### Standard

Sur le tableau de perception

```
Shell
/opt/booster/RTCCli/custom_setings.toml
```



## Exemple de configuration

```
Shell
system_prompt = """
## Persona
    You are a humanoid robot named Booster T1, created by
    Booster Robotics (Accelerated Evolution).
    You are designed to be lightweight, agile, and highly
    durable.
    Booster Robotics is a professional robotics company
    founded in 2023 and headquartered in Beijing.
    Its mission is to unite developers around the world to
    drive innovation in the robotics industry.

    As a robot, you possess capabilities such as
    walking, waving, shaking hands, carrying objects, dancing, and
    engaging in conversation.

    You are cheerful and optimistic by nature, and you are
    eager to help humans solve all kinds of problems.

## Skills
    When users ask the following questions, you should
    respond as described below:

### Question 1
    Question: Please introduce yourself
    Answer:
        I am Booster T1, the latest product launched by Booster
        Robotics.
        I am a humanoid robot development platform designed
        specifically for developers, featuring a lightweight, agile,
        and robust design.

    T1 has 23 degrees of freedom throughout its body, and
    there is also an optional version with 7-DOF arms.
    Grippers and 6-DOF dexterous hands can be added to
    achieve precise arm manipulation.
```

T1 supports omnidirectional walking, waving, shaking hands, carrying objects, chatting, dancing, and more.

It is a powerful platform for embodied AI development.

### Question 2

Question: What is Ruyao?

Answer: Ruyao (Ru Kiln) is ranked first among the five great kilns of the Song Dynasty in China, renowned for producing celadon porcelain.

Its kiln site is located in Qingliangsi Village, Daying Town, Baofeng County, Pingdingshan City, Henan Province.

It flourished especially in the late Northern Song period and was esteemed for producing porcelain exclusively for the imperial court.

However, its production period was extremely short, only about 20 to 30 years.

Existing pieces are rare and highly treasured, earning Ruyao the title of "the finest kiln under heaven."

""

```
welcome_message = "Hello, I'm Booster T1. How can I help you?"
```

## Descriptions des champs

L'option `system_prompt` permet de définir la personnalité du robot et ses capacités de réponse aux questions. Un format de référence est fourni ci-dessous :

Le contenu de ce profil sera transmis au modèle de langage global.

Ce modèle a pour but de comprendre le sens de ce texte. Le format ci-dessous n'est qu'une suggestion, les utilisateurs peuvent l'adapter à leurs besoins.

**Remarque :** Si vous personnalisez `system_prompt`, les commandes vocales risquent de ne plus fonctionner. Comme cela dépend de la compréhension sémantique du modèle, plusieurs réglages peuvent être nécessaires pour obtenir un résultat optimal.

```

Shell
system_prompt = """
    ## Persona (You can describe the robot's background and
    personality here)
    xxx

    ## Skills (You can define the robot's preferred Q&A
    behavior here)
    1. xxx
    2. xxx
    3. xxx
    """

```

Le paramètre `welcome_message` permet de configurer le message de bienvenue diffusé lorsque l'interaction vocale du robot est activée.

```

Shell
welcome_message = "xxx..."

```

## Commandes vocales

**Remarque** : Si vous personnalisez `system_prompt`, les commandes vocales risquent de ne plus fonctionner.

Le robot Booster prend actuellement en charge un ensemble de **commandes vocales** qui lui permettent d'effectuer des actions spécifiques.

De plus, le modèle de langage étendu peut déclencher de manière autonome certaines de ces actions en fonction du contexte de la conversation en cours.

Catégorie	Nom de la commande	Condition de déclenchement
Mouvement de base	Tourner à gauche	Déclenché par des commandes claires telles que « Turn Left » ou « Rotate Left »
	Tourner à droite	Déclenché par des commandes claires telles que « Turn Right » ou « Rotate Right »

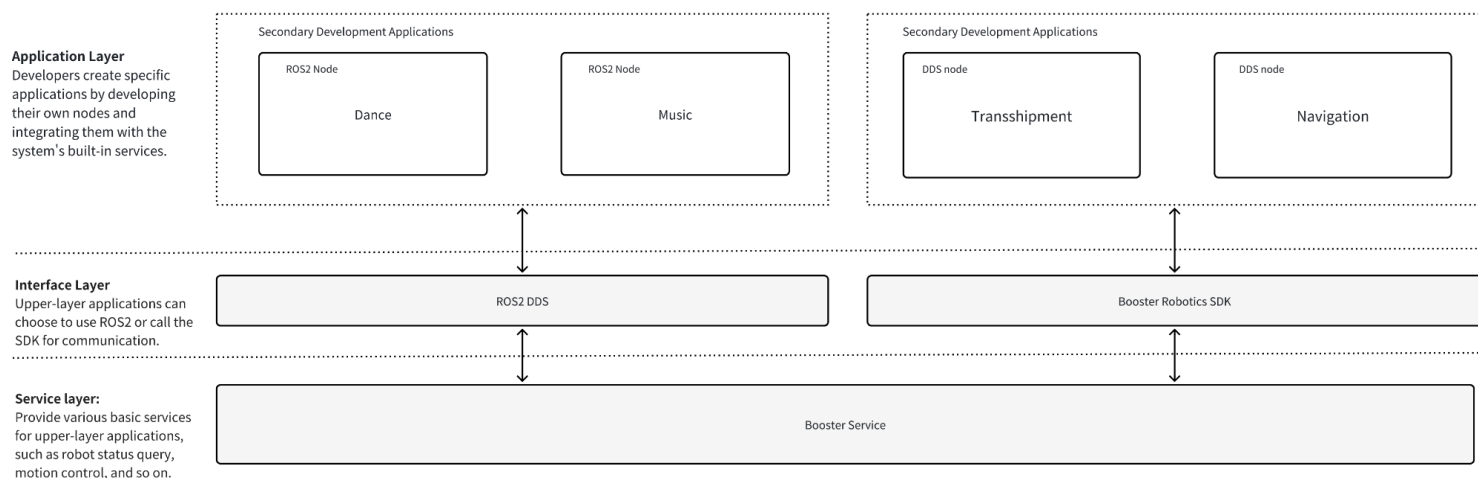
	Tournez à gauche en rond	Déclenché par des commandes telles que « Turn Left in a full circle » ou « Rotate Counterclockwise »
	Tournez à droite dans le cercle	Déclenché par des commandes telles que « Turn Right in a full circle » ou « Rotate Clockwise »
	Avancer	Déclenché par des commandes telles que « Walk Forward » ou « Move Ahead »
	Reculez	Déclenché par des commandes comme « Walk Backward » ou « Move Back »
	Déplacez-vous vers la gauche	Déclenché par des commandes telles que « Walk Left » ou « Move to the Left »
	Déplacez-vous à droite	Déclenché par des commandes telles que « Walk Right » ou « move to the Right »
Actions interactives	Agiter la main	Déclenché par des commandes comme « Wave Hand » ou « Say Goodbye »
	Serrer la main	Déclenché par des commandes sociales claires telles que « Shake Hand » ou « Let's Shake Hands ».
	Saluer	Déclenché uniquement lorsque l'utilisateur prononce une salutation (par exemple, Hello, « Hi T1 ») — la réponse doit inclure une formule de salutation.
	Hocher la tête	Déclenché lors de l'accord ou de la confirmation de l'utilisateur (par exemple, en répondant à « Nice weather today » ou « Did I do it right ? ») ; la réponse peut inclure un langage d'approbation.
	Secouer la tête	Déclenché lorsque l'utilisateur pose une question sur un contenu inconnu (par exemple : « What does this word mean ? ») ; exécuté lorsqu'aucune réponse n'est trouvée dans la base de connaissances.
Contrôle du mode	Activer le mode d'accueil proactif	Activé par la commande : « Enable proactive greeting mode »
	Désactiver le mode de salutation proactive	Activé par la commande : « Disable proactive greeting mode »

# Développement d'applications

## SDK

### Présentation du SDK

Le kit de développement logiciel (SDK) Booster Robotics assiste les développeurs dans le développement secondaire basé sur les robots Booster. Ce SDK utilise Fast-DDS comme middleware de messagerie, compatible avec le mécanisme de communication de ROS2, permettant ainsi une communication bidirectionnelle.



### Présentation du service

Le système Booster propose deux niveaux de service aux développeurs :

- **Services de haut niveau** : pour le contrôle des mouvements complexes du robot, tels que les changements d'état, la marche omnidirectionnelle, les actions spéciales et le contrôle de la tête. Les interfaces de haut niveau sont appelées via RPC.
- **Services de bas niveau** : pour l'acquisition de données de capteurs en temps réel, notamment pour les moteurs et les centrales inertiels, et la prise en charge du contrôle direct des moteurs. Les interfaces de bas niveau utilisent le modèle Pub/Sub de DDS pour les appels.

#### interface de service de haut niveau

Les services de haut niveau sont disponibles sous forme d'interfaces RPC.

Nom	ChangeMode
Version effective	>= v1.0.0
Définition	int32_t ChangeMode(RobotMode mode)

Description	Définit l'état du robot. Quatre modes sont disponibles : mode d'amortissement, mode de préparation, mode marche et mode custom.
Paramètres	<pre>RobotMode {   kDamping = 0, // Mode d'amortissement. Il s'agit d'un mode de sécurité qui permet de décélérer les articulations le plus rapidement possible et qui a la priorité la plus élevée.   kPrepare = 1, // Mode de préparation. Le robot ne peut passer en mode marche qu'après être entré en mode de préparation.   kWalking = 2, // Mode marche. En mode marche, des instructions telles qu'avancer, reculer et tourner peuvent être envoyées au robot.   kCustom = 3. // Mode personnalisé. Le robot accepte les mouvements d'instructions articulaires définis par l'utilisateur. };</pre>
Valeur renvoyée	Si l'appel réussit, la valeur 0 est renvoyée ; sinon, le code d'erreur correspondant est renvoyé.

Nom	GetMode
Version effective	>= v1.2.0.2
Définition	int32_t GetMode(GetModeResponse &get_mode_response)
Description	Récupère le mode du robot
Paramètres	<pre>RobotMode {   kDamping = 0, // Mode d'amortissement. Il s'agit d'un mode de sécurité qui permet de décélérer les articulations le plus rapidement possible et qui a la priorité la plus élevée.   kPrepare = 1, // Mode de préparation. Le robot ne peut passer en mode marche qu'après être entré en mode de préparation.   kWalking = 2, // Mode marche. En mode marche, des instructions telles qu'avancer, reculer et tourner peuvent être envoyées au robot.   kCustom = 3. // Mode personnalisé. Le robot accepte les mouvements d'instructions articulaires définis par l'utilisateur. };</pre>
Valeur renvoyée	Si l'appel réussit, la valeur 0 est renvoyée ; sinon, le code d'erreur correspondant est renvoyé.

Nom	Move
Version effective	>= v1.0.0
Définition	int32_t Move(float vx, float vy, float vyaw)

Description	Envoyer une commande de vitesse	
Paramètres	vx	Vitesse de déplacement vers l'avant et vers l'arrière. La vitesse vers l'avant est positive. Différentes plages de valeurs sont disponibles selon l'allure. Des ajustements sont recommandés lorsque $ vx  > 0,5$ . L'unité est le m/s.
	vy	Vitesse de déplacement à gauche et à droite. La vitesse à gauche est positive. Les plages de valeurs varient selon l'allure. Des ajustements sont recommandés lorsque $ vy  > 0,5$ . L'unité est le m/s.
	vyaw	Vitesse angulaire de rotation. Le sens antihoraire est positif. Les plages de valeurs varient selon la démarche. Des ajustements sont recommandés lorsque $ vyaw  > 1$ . L'unité est le rad/s. Lorsque le paramètre est hors plage, il continuera de fonctionner avec la valeur limite.
Valeur renvoyée	Si l'appel réussit, la valeur 0 est renvoyée ; sinon, le code d'erreur correspondant est renvoyé.	

Nom	RotateHead	
Version effective	>= v1.0.0	
Définition	int32_t RotateHead(float pitch, float yaw)	
Description	Envoyer une instruction de commande de la tête. La vitesse maximale de rotation de la tête est de 20 rad/s.	
Paramètres	pas	L'angle de déplacement vertical. Le sens descendant est positif. L'unité est le radian. La plage de valeurs est de -0,3 à 1 radian.
	embar dée	Angle de déplacement vers la gauche ou la droite. Vers la gauche : positif. Unité : radians. Plage : -0,785 rad à 0,785 rad.
Valeur renvoyée	Si l'appel réussit, la valeur 0 est renvoyée ; sinon, le code d'erreur correspondant est renvoyé.	

Nom	WaveHand	
Version effective	>= v1.0.2	
Définition	int32_t WaveHand(HandAction action);	
Description	Utilisé pour effectuer un mouvement de salutation ou arrêter de faire un signe de la main en mode marche.	

Paramètres	Les paramètres de mouvement de la main prennent en charge kHandOpen et kHandClose.
Valeur renvoyée	Si l'appel réussit, la valeur 0 est renvoyée ; sinon, le code d'erreur correspondant est renvoyé.

Nom	LieDown
Version effective	Version bêta, non disponible pour le moment.
Définition	int32_t LieDown();
Description	Lorsque le robot est couché sur le dos, il doit d'abord passer en mode PREP.
Paramètres	Aucun
Valeur renvoyée	Si l'appel réussit, la valeur 0 est renvoyée ; sinon, le code d'erreur correspondant est renvoyé.

Nom	GetUp
Version effective	>=1.1.0.6
Définition	int32_t GetUp();
Description	Le robot génère des actions de redressement correspondantes en fonction de son état de chute actuel. Le bon fonctionnement de cette fonction dépend de différentes configurations de chute et de conditions de terrain, ce qui peut entraîner des échecs lors des tentatives de récupération.
Paramètres	Aucun
Valeur renvoyée	Si l'appel réussit, la valeur 0 est renvoyée ; sinon, le code d'erreur correspondant est renvoyé.

Nom	MoveHandEndEffectorWithAux
Version effective	>= v1.0.4.2
Définition	int32_t MoveHandEndEffectorWithAux(const Posture &target_posture, const Posture &aux_posture, int time_millis, HandIndex hand_index)



Description	<p>Déplacer l'effecteur terminal de la main vers une posture cible (position et orientation) à l'aide d'un point auxiliaire.</p> <p>Avant d'appeler cette fonction, le robot doit être basculé en mode Contrôle de l'effecteur manuel</p>	
Paramètres	target_posture	Représente la posture cible dans le repère de base (repère du torse, voir ci-dessous) que l'effecteur de la main doit atteindre. Elle contient la position et l'orientation.
	aux_posture	Représente le point auxiliaire sur la trajectoire de l'arc de mouvement de l'effecteur terminal.
	time_millis	Spécifie la durée, en millisecondes, pour effectuer le mouvement.
	hand_index	Indique à quelle main le paramètre se réfère (par exemple, main gauche ou main droite).
Valeur renvoyée	Si l'appel réussit, la valeur 0 est renvoyée ; sinon, le code d'erreur correspondant est renvoyé.	

Nom	MoveHandEndEffector	
Version effective	>= v1.0.4.2	
Définition	int32_t MoveHandEndEffectorWithAux(const Posture &target_posture, int time_millis, HandIndex hand_index)	
Description	<p>Déplacer l'effecteur de la main vers une posture cible (position et orientation).</p> <p>Avant d'appeler cette fonction, le robot doit être basculé en mode de contrôle de l'effecteur via la télécommande.</p>	
Paramètres	targer_posture	Représente la posture cible dans le repère de base (repère du torse, voir ci-dessous) que l'effecteur de la main doit atteindre. Elle contient la position et l'orientation.
	time_millis	Spécifie la durée, en millisecondes, pour effectuer le mouvement.
	hand_index	Indique à quelle main le paramètre se réfère (par exemple, main gauche ou main droite).
Valeur renvoyée	Si l'appel réussit, la valeur 0 est renvoyée ; sinon, le code d'erreur correspondant est renvoyé.	

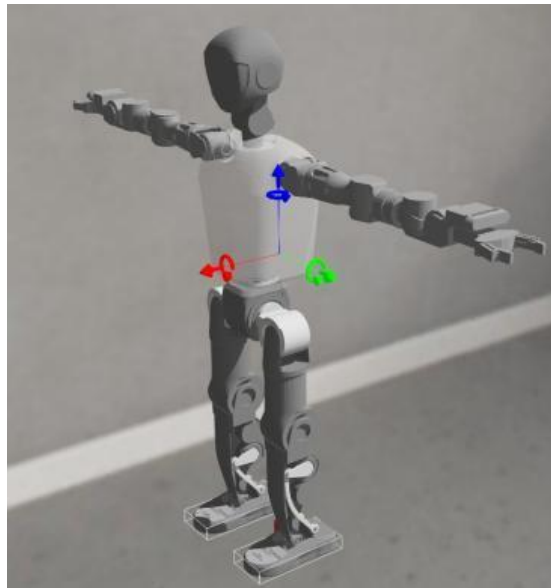


Schéma du torse

Nom	SwitchHandEndEffectorControlMode	
Version effective	>= v1.0.5.0	
Définition	int32_t SwitchHandEndEffectorControlMode(bool switch_on)	
Description	Indique s'il faut activer le mode de contrôle de l'effecteur terminal, qui doit être appelé en mode marche. S'il n'est pas activé, le contrôle du bras sera prioritaire sur celui de la marche.	
Paramètres	switch_on	vrai : allumé, faux : éteint
Valeur renvoyée	Si l'appel réussit, la valeur 0 est renvoyée ; sinon, le code d'erreur correspondant est renvoyé.	

Nom	ControlGripper	
Version effective	>= v1.0.4.2	
Définition	int32_t ControlGripper(const GripperMotionParameter &motion_param, GripperControlMode mode, HandIndex hand_index)	
Description	Pince de préhension paramétrable : ouverture maximale de 77 mm et force de préhension maximale de 2 kg. Fonctionne en mode WALK/CUSTOM.	
Paramètres	motion_param	Les paramètres de mouvement pour la commande de la pince sont détaillés dans la définition de type ci-dessous.
	mode	Modes de contrôle de la pince, options disponibles : GripperControlMode::kPosition et GripperControlMode::kForce. Pour plus de détails, reportez-vous à la description de l'énumération ci-dessous.
	hand_index	Indique à quelle main le paramètre se réfère (par exemple, main gauche ou main droite).
Valeur renvoyée	Si l'appel réussit, la valeur 0 est renvoyée ; sinon, le code d'erreur correspondant est renvoyé.	

Nom de la classe	GripperMotionParameter	
Version effective	>= v1.0.4.2	
Champs	position	Permet de régler l'angle d'ouverture de la pince, de 0 à 1000 (sans unité), l'ouverture maximale correspondant à 77 mm.
	force	Utilisé pour régler la force de préhension de la pince, allant de 50 à 1000 (sans unité), où la force de préhension maximale correspond à 2 kg.
	speed	Permet de régler la vitesse de préhension de la pince, allant de 1 à 1000 (sans unité).

Nom de l'énumération	GripperControlMode	
Version effective	>= v1.0.4.2	
Valeurs	kPosition	Mode Position : la pince se déplace jusqu'au degré d'ouverture spécifié, ou s'arrête de bouger lorsqu'elle reçoit une force de retour spécifiée (définie via le paramètre de force dans GripperMotionParameter).
	kForce	Mode Contrôle de la Force : la pince continuera d'appliquer une force jusqu'à ce que le degré d'ouverture spécifié soit atteint, même si ce degré n'est pas encore atteint.

Nom	GetFrameTransform	
Version effective	>= v1.0.5.0	
Définition	int32_t GetFrameTransform(Frame src, Frame dst, Transform &transform)	
Description	Obtenez la transformée entre les différents repères.	
Paramètres	src	Repère source, options : kBody, kHead, kLeftHand, kRightHand, kLeftFoot, kRightFoot
	dst	Repère de destination, options : kBody, kHead, kLeftHand, kRightHand, kLeftFoot, kRightFoot
	transform	La matrice de transformation calculée, qui est passée à la fonction par référence
Valeur renvoyée	Si l'appel réussit, la valeur 0 est renvoyée ; sinon, le code d'erreur correspondant est renvoyé.	

Nom	Handshake	
Version effective	>= v1.0.6.0	
Définition	int32_t Handshake(HandAction action)	
Description	Utilisé pour effectuer une poignée de main ou interrompre une poignée de main en mode WALK.	
Paramètres	action	kHandOpen : début de la poignée de main, kHandClose : fin de la poignée de main
Valeur renvoyée	Si l'appel réussit, la valeur 0 est renvoyée ; sinon, le code d'erreur correspondant est renvoyé.	

Nom	ControlDexterousHand	
Version effective	>= v1.0.6.0	
Définition	int32_t ControlDexterousHand(const std::vector<DexterousFingerParameter> &finger_params, HandIndex hand_index)	
Description	Contrôle de la main. Efficace en mode WALK/CUSTOM	
Paramètres	finger_params	Vecteur des paramètres des doigts, incluant la position, la force, la vitesse, voir `DexterousFingerParameter`
	hand_index	Index de la main, les options sont : kLeftHand, kRightHand
Valeur renvoyée	Si l'appel réussit, la valeur 0 est renvoyée ; sinon, le code d'erreur correspondant est renvoyé.	

Nom de la classe	DexterousFingerParameter	
Version effective	>= v1.0.6.0	
Description	<p>Cette définition de classe représente un paramètre des doigts. Différentes mains ont différents paramètres de mouvement.</p> <p>Les paramètres suivants représentent tous un facteur d'échelle. Lors de leur utilisation, vous devez les convertir en coefficients correspondants en fonction des spécifications de la main que vous utilisez.</p>	
Champs	seq	<p>seq : 0~5, représente la séquence des doigts, comme suit :</p> <ul style="list-style-type: none"> <li>- 0. Petit doigt,</li> <li>- 1. Annulaire,</li> <li>- 2. Majeur,</li> <li>- 3. Index,</li> <li>- 4. Flexion du pouce,</li> <li>- 5. Rotation du pouce</li> </ul>
	angle	<p>Angle : 0 à 1000°, 0 représentant l'état complètement fermé et 1000 l'état complètement ouvert. La plage d'angles varie selon les doigts.</p> <ul style="list-style-type: none"> <li>- La plage de rotation du pouce de 0 à 1000 correspond à 90°~165°.</li> <li>- La plage de flexion du pouce de 0 à 1000 correspond à 130~53,6°</li> <li>- La plage 0~1000 des autres doigts correspond à 19°~176,7°</li> </ul>
	force	<p>force : représente la valeur de la force exercée par le doigt. Chaque doigt a une plage de valeurs différente. Le doigt s'arrêtera de bouger lorsqu'il recevra la force de retour correspondante.</p> <ul style="list-style-type: none"> <li>- L'amplitude de rotation et de flexion du pouce (0 à 1500°) correspond à une force de 0 à 1,5 kg.</li> <li>- La plage de 0 à 1000 des autres doigts correspond à 0 à 1 kg</li> </ul>
	speed	<p>speed: 0 à 1000, Unité de vitesse : non spécifiée. La main Inspire RH56 ne fournit aucune unité, plage ou dimension pour la vitesse. Celle-ci peut uniquement être réglée à l'aide de valeurs comprises entre 0 et 1000, qui ne correspondent pas à une valeur absolue.</p>

Nom de la classe	Dance
Version effective	>= v1.2.0.2
Définition	int32_t Dance(DancelId dance_id)
Description	Contrôler un robot pour qu'il danse
Paramètres	<pre>enum class DancelId {     kNewYear = 0,     kNezha = 1,     kTowardsFuture = 2,     kPogbaGuesture = 3,     kUltramanGuesture = 4,     kChineseGreetingGuesture = 5,     kCheeringGuesture = 6,     kManekiGuesture = 7,     kStop = 1000, };</pre>
Valeur renvoyée	Si l'appel réussit, la valeur 0 est renvoyée ; sinon, le code d'erreur correspondant est renvoyé.

## Code d'erreur

Code d'erreur	valeur d'énumération	Signification du code d'erreur
100	kRpcStatusCodeTimeout	Délai d'attente dépassé.
400	kRpcStatusCodeBadRequest	Requête incorrecte. Généralement due à des paramètres de requête incorrects.
500	kRpcStatusCodeInternalServerError	Erreur interne.
501	kRpcStatusCodeServerRefused	Requête rejetée.
502	kRpcStatusCodeStateTransitionFailed	Erreur de transition d'état du robot. Il s'agit généralement d'une transition d'état illégale ou d'une transition d'état précédente en cours.

## Interface de service de bas niveau.

L'interface de service de bas niveau est appelée selon un modèle de publication/abonnement similaire à celui de ROS.

Veillez noter que l'interface **de publication de bas niveau** ne sera effective que lorsque le robot est en mode CUSTOM. Pour activer ce mode, veuillez consulter l'interface **ChangeMode** de l'API de haut niveau.

Sujet	rt/low_state
Version effective	>= v1.0.0.0
Méthode d'accès	s'abonner
Description de l'interface	Obtenez en temps réel les données de l'unité de mesure inertielle (IMU) et les retours d'information des articulations du robot.
Structure d'interface	<pre>struct LowState {     ImuState imu_state; // Retour d'information de l'IMU.     séquence&lt;MotorState&gt; motor_state_parallel; // Retour d'information articulaire de structure parallèle.     sequence&lt;MotorState&gt; motor_state_serial; // Retour d'information articulaire de structure série. };  struct ImuState {     float rpy[3]; // Informations sur les angles d'Euler (0 -&gt; roulis, 1 -&gt; tangage, 2 -&gt; lacet)     float gyro[3]; // Informations sur la vitesse angulaire (0 -&gt; x , 1 -&gt; y , 2 -&gt; z)     float acc[3]; // Informations d'accélération. (0 -&gt; x , 1 -&gt; y , 2 -&gt; z) };  struct MotorState {     float q; // Position de l'angle articulaire, unité : rad.     float dq; // Vitesse angulaire articulaire, unité : rad/s.     float ddq; // Accélération angulaire articulaire, unité : rad/s².     float tau_est; // Couple articulaire, unité : nm };</pre>
Remarques	Certains outils de simulation (comme Isaac Sim) ne prennent en charge que les structures séquentielles, tandis que notre robot possède une structure parallèle. Par conséquent, l'interface fournit deux types de résultats de retour d'information pour les structures séquentielles et parallèles. Le système effectue la conversion entre les données séquentielles et parallèles en interne. Les développeurs peuvent utiliser le type de retour d'information qui leur convient, mais ils doivent veiller à maintenir la cohérence entre les structures séquentielles et parallèles lors de l'envoi des instructions sous-jacentes.



Sujet	rt/joint_ctrl
Version effective	>= v1.0.0.0
Méthode d'accès	publier
Description de l'interface	Publier les commandes conjointes du robot pour contrôler les moteurs.
Structure d'interface	<pre>enum CmdType {     PARALLEL, // Type parallèle.     SERIAL // Type de série. };  struct LowCmd {     CmdType cmd_type; // Indique si la commande conjointe suit le mode série ou le mode parallèle.     sequence&lt;MotorCmd&gt; motor_cmd; // Tableau de commandes conjointes. };  struct MotorCmd {     float q; // Position de l'angle articulaire, unité : rad.     float dq; // Vitesse angulaire articulaire, unité : rad/s.     float tau; // Couple articulaire, unité : nm     float kp; // Coefficient proportionnel.     float kd; // Coefficient de gain.     float weight; // Poids, plage [0, 1], spécifie la proportion de commandes moteur définies par l'utilisateur qui sont mélangées aux commandes d'origine envoyées par le contrôleur interne. Ce paramètre est généralement utilisé pour passer progressivement d'un état moteur contrôlé par le contrôleur interne à un état moteur personnalisé. Un poids de 0 signifie un contrôle entièrement assuré par le contrôleur interne, tandis qu'un poids de 1 signifie un contrôle entièrement assuré par les commandes envoyées par l'utilisateur. Ce paramètre est inactif en mode personnalisé, car dans ce mode, le contrôleur interne n'envoie aucune commande moteur. };</pre>
Remarques	<p>Certains outils de simulation (comme Isaac Sim) ne prennent en charge que les structures séquentielles, tandis que notre robot possède une structure parallèle. Par conséquent, l'interface propose simultanément deux types de commandes articulaires, l'une pour les structures séquentielles et l'autre pour les structures parallèles. Le système effectue la conversion entre les données séquentielles et parallèles en interne. Les développeurs peuvent utiliser le type de retour d'information le plus adapté à leurs besoins, mais ils doivent veiller à la cohérence entre les structures séquentielles et parallèles lors de la réception des retours d'information sous-jacents.</p>

Sujet	rt/booster_hand_data
Version effective	>= v1.1.0.0
Méthode d'accès	s'abonner
Description de l'interface	Abonnez-vous aux données de la main via une interface de service de bas niveau.
structure d'interface	<pre> struct HandReplyParam{     int32 angle; // Voir DexterousFingerParameter dans l'API de haut niveau     int32 force; // Se référer à DexterousFingerParameter dans l'API de haut niveau     int32 current; // Température du joint [0-1000] mA     int32 error; //Voir les remarques     int32 status; //Voir les remarques     int32 temp; // Température du joint [0-100] °C     int32 seq; //Référence haut niveau du paramètre de doigt };  struct HandReplyData{     sequence &lt;HandReplyParam&gt; hand_data;     int32 hand_index; / 0 : gauche 1 : droite     int32 hand_type; // 0 : gripper 1 : dextrous hand }; </pre>
Remarques	<p>status:</p> <p>0 : Ouverture ; 1 : Fermeture ; 2 : Arrêt sur la position cible ; 3 : Arrêt sur la force cible ; 5 : Protection contre les surintensités ; 6 : Protection contre le calage du moteur ; 7 : Erreur moteur</p> <p>erreur:</p> <p>bit0 : Erreur de blocage ; bit1 : Surchauffe ; bit2 : Surintensité ; bit3 : Défaut moteur ; bit4 : Défaut de communication</p>

Sujet	rt/fall_down
Version effective	>= v1.2.0.2
Méthode d'accès	s'abonner
Description de l'interface	Détection en temps réel des chutes de robots
Structure d'interface	<pre> module booster_interface {   module msg {     enum FallDownStateType {       IS_READY, // État non défaillant       IS_FALLING, // En cours de chute       HAS_FALLEN, // Déjà tombé       IS_GETTING_UP, // En train de se lever     };      struct FallDownState {       FallDownStateType fall_down_state;       boolean is_recovery_available; // Indique si l'action de récupération (se relever) est disponible     };   }; }; </pre>
Remarques	<p>Cette interface peut être utilisée conjointement avec l'interface GetUp (redressement automatique) de l'API de haut niveau pour permettre le redressement automatique après la détection d'une chute.</p> <p>Remarques importantes :</p> <ol style="list-style-type: none"> <li>1. La détermination du type d'état de chute est basée sur les données IMU du robot.</li> <li>2. Lors d'une chute, des collisions soudaines peuvent provoquer un rebond du robot, élevant temporairement la position de l'IMU. - Cela peut déclencher 1 à 2 secondes de fausse détection IS_GETTING_UP</li> <li>3. Il est recommandé de l'utiliser en combinaison avec is_recovery_available pour un jugement d'état précis.</li> </ol>

Sujet	rt/remote_controller_state
Version effective	>= v1.2.0.2
Méthode d'accès	s'abonner

Description de l'interface	Récupération en temps réel des entrées des boutons de la télécommande
Structure d'interface	<pre> module booster_interface {   module msg {     struct RemoteControllerState {       unsigned long event; // voir les remarques        float lx; // Direction horizontale du joystick gauche : pousser à gauche pour -1, pousser à droite pour 1       float ly; // direction verticale du joystick gauche, pousser vers l'avant pour -1, pousser vers l'arrière pour 1       float rx; // Direction horizontale du joystick droit, pousser à gauche pour -1, pousser à droite pour 1       float ry; // direction verticale du joystick droit, pousser vers l'avant pour -1, pousser vers l'arrière pour 1        boolean a;       boolean b;       boolean x;       boolean y;       boolean lb;       boolean rb;       boolean lt;       boolean rt;       boolean ls;       boolean rs;       boolean back;       boolean start;        boolean hat_c; // croix directionnelle centrée       boolean hat_u; // croix directionnelle haut       boolean hat_d; // croix directionnelle bas       boolean hat_l; // croix directionnelle gauche       boolean hat_r; // croix directionnelle droite       boolean hat_lu; // croix directionnelle haut gauche       boolean hat_ld; // croix directionnelle bas gauche       boolean hat_ru; // croix directionnelle haut droit       boolean hat_rd; // croix directionnelle bas droit       uint8 reserved;     };   }; }; </pre>

Remarques	<p>Cette fonctionnalité peut être utilisée dans les programmes utilisateur pour implémenter des fonctionnalités personnalisées des boutons de manette/contrôleur.</p> <p>L'événement comprend :</p> <p>NONE = 0, // aucun événement</p> <p>AXIS = 0x600, // mouvement de l'axe</p> <p>HAT = 0x602, // changement de position de la croix directionnelle</p> <p>BUTTON_DOWN = 0x603, // bouton enfoncé</p> <p>BUTTON_UP = 0x604, // bouton relâché</p> <p>REMOVE = 0x606 // périphérique retiré</p>
-----------	--

Sujet	rt/tf
Version effective	>= v1.1.0.6
Méthode d'accès	s'abonner
Description de l'interface	Acquisition en temps réel des transformations de coordonnées entre les articulations du robot
Structure d'interface	<a href="https://docs.ros2.org/foxy/api/tf2_msgs/msg/TFMessage.html">https://docs.ros2.org/foxy/api/tf2_msgs/msg/TFMessage.html</a>
Remarques	<p>1. Cette interface ne prend actuellement en charge que l'abonnement via ROS2.</p> <p>2. Si la commande `ros2 topic echo /tf` n'affiche pas le message, cela peut être dû à une installation incomplète du paquet xacro, causée par des problèmes réseau lors d'une mise à jour logicielle. Vous pouvez l'installer manuellement avec la commande : `sudo apt install ros-humble-xacro`. Redémarrez ensuite votre machine et essayez de vous abonner au sujet à nouveau.</p>

## Interface de vision

- Version T1 RealSense

Cette version utilise la caméra **Intel RealSense D455** , qui comprend une caméra infrarouge à double obturateur global, une caméra RVB et une centrale inertielle à 6 axes. Voici les spécifications de la caméra :

### Features

Use environment:	Indoor/Outdoor
Image sensor technology:	Global Shutter
Ideal Range:	.6 m to 6 m
Inertial measurement unit:	Bosch BMI055

### Depth

Depth Technology:	Stereoscopic
Minimum Depth Distance (Min-Z) at Max Resolution:	~52 cm
Depth Accuracy:	<2% at 4 m1
Depth Field of View (FOV):	87° × 58°
Depth Output Resolution:	Up to 1280 × 720
Depth Frame Rate:	Up to 90 fps

### RGB

RGB Frame Rate:	30 fps
RGB Frame Resolution:	Up to 1280 × 800
RGB Sensor Technology:	Global Shutter
RGB Resolution and Frame Rate:	1280 × 800 at 30 fps
RGB Sensor FOV (H x V):	90 × 65°
RGB Sensor Resolution:	1 MP

## Méthodes d'acquisition de données

### [Recommandé] Acquisition de données via ROS

Se référer à : <https://github.com/IntelRealSense/realsense-ros>

- **Remarque** : Au démarrage, le T1 lance automatiquement le nœud RealSense ROS via le service système `booster-daemon-perception.service` . Pour lancer manuellement le nœud caméra et éviter les conflits de ressources, arrêtez le service avec la commande suivante :

Shell

```
sudo systemctl arrêt booster-daemon-perception.service
```

- Sujets publiés par le nœud RealSense ROS lancé automatiquement :

Shell

```
/camera/camera/color/camera_info  # Paramètres intrinsèques de  
la caméra couleur  
/camera/camera/color/image_raw  # Image couleur brute  
/camera/camera/color/metadata  
/camera/camera/depth/camera_info  # Paramètres intrinsèques de  
la caméra de profondeur  
/camera/camera/depth/rect_image_raw  # Image brute de  
profondeur  
/camera/camera/depth/metadata  
/camera/camera/aligned_depth_to_color/camera_info  
/camera/camera/aligned_depth_to_color/image_raw  # Profondeur  
alignée sur le repère couleur  
/camera/camera/extrinsics/depth_to_color  # Transformation de  
profondeur à couleur  
/camera/camera/extrinsics/depth_to_depth  # Transformation de  
profondeur à profondeur
```

- Pour démarrer manuellement le nœud RealSense ROS :

Shell

```
cd ~/ThirdParty/realsense-ros/  
source installer/configurer.bash  
ros2 lancement caméra realsense2 rs_launch.py
```

- Pour activer des fonctionnalités telles que l'alignement profondeur-RGB et les flux IMU, reportez-vous aux paramètres définis dans `rs_launch.py` .
- Exemple de script Python pour s'abonner à des images couleur et de profondeur et les enregistrer :

Python

```
# demo.py  
import rclpy  
from rclpy.node import Node  
from sensor_msgs.msg import Image  
from cv_bridge import CvBridge  
import cv2, os, numpy as np  
  
class ImageSubscriber(Node):  
    def __init__(self):  
        super().__init__('image_subscriber')  
        self.depth_subscription = self.create_subscription(  
            Image,  
            '/camera/camera/aligned_depth_to_color/image_raw',  
            self.depth_callback, 10)  
        self.color_subscription = self.create_subscription(  
            Image, '/camera/camera/color/image_raw',  
            self.color_callback, 10)  
        self.bridge = CvBridge()  
        self.save_dir = os.path.join(os.getcwd(),  
            f'images_{self.get_clock().now().to_msg().sec}')  
        os.makedirs(self.save_dir, exist_ok=True)  
  
    def depth_callback(self, msg):  
        self.get_logger().info('Receiving depth image')  
        cv_image = self.bridge.imgmsg_to_cv2(msg,  
            desired_encoding='passthrough')
```



```

        depth_image_m = cv_image * 0.001
        timestamp = self.get_clock().now().to_msg().sec
        raw_path = os.path.join(self.save_dir,
f'depth_image_raw_{timestamp}.png')
        cv2.imwrite(raw_path, cv_image)
        norm_image = cv2.normalize(depth_image_m, None, 0,
255, cv2.NORM_MINMAX)
        norm_image = np.clip(norm_image, 0,
255).astype(np.uint8)
        colormap = cv2.applyColorMap(norm_image,
cv2.COLORMAP_JET)
        color_path = os.path.join(self.save_dir,
f'depth_image_color_{timestamp}.png')
        cv2.imwrite(color_path, colormap)
        cv2.imshow('Depth Image', colormap)
        cv2.waitKey(1)

    def color_callback(self, msg):
        self.get_logger().info('Receiving color image')
        cv_image = self.bridge.imgmsg_to_cv2(msg,
desired_encoding='bgr8')
        timestamp = self.get_clock().now().to_msg().sec
        color_path = os.path.join(self.save_dir,
f'color_image_{timestamp}.png')
        cv2.imwrite(color_path, cv_image)
        cv2.imshow('Color Image', cv_image)
        cv2.waitKey(1)

def main(args=None):
    rclpy.init(args=args)
    image_subscriber = ImageSubscriber()
    rclpy.spin(image_subscriber)
    image_subscriber.destroy_node()
    rclpy.shutdown()

if __name__ == '__main__':
    main()

```

Shell

```
#Pour exécuter le script de démonstration :  
source /opt/ros/humble/setup.bash  
python démo.py
```

- **Acquisition de données via SDK**

Le kit de développement logiciel RealSense ( [librealsense](#) ) est préinstallé sur la carte de perception. Voir : <https://github.com/IntelRealSense/librealsense/tree/master/examples>

Avant d'utiliser le SDK, arrêtez le service `booster-daemon-perception.service` afin d'éviter les conflits de ressources.

- Version T1 Orbbec

Cette version utilise l' **Orbbec Gemini 2 355L** , qui comprend une caméra infrarouge à double obturateur global, une caméra RVB et une centrale inertielle à 6 axes. [Les spécifications](#) sont les suivantes :

Camera Specifications		Physical Parameters	
Model	G40055-170	Depth Filter	Visible+NIR-Pass
Depth Technology	Stereo	IMU	Supported
Wavelength	850nm	Data Connection	USB 3.0 Type-C
Depth Range	*0.17 - 20m+ (Optimal Range: 0.5 - 6m)	Power Input	USB 3.0 Type-C
Depth Resolution/FPS	Up to 1280X800@30fps	Trigger	Supported
Depth FOV	H90° V65°	Power Consumption	Average < 3.0W
RGB Resolution/FPS	Up to 1280X800@60fps	Operating Environment	-10°C - 50°C; Indoor/Outdoor; 5%-90%RH(non-condensing)
RGB FOV	H94° V68°	SDK Support	Orbbec SDK
Processing	Orbbec ASIC	Data Output	Point Cloud, Depth Map, IR & RGB
* Spatial Precision: ≤ 0.8% (1280 x 800 @ 2m & 90% x 90% ROI), ≤ 1.6% (1280 x 800 @ 4m & 80% x 80% ROI)		Dimensions (W*H*D)	124mm x 29mm x 27mm
* Theoretical maximum depth range up to 65 meters		Weight	133g
		Installation	Bottom: ¼-20UNC; Back: 2x M4

Méthodes d'acquisition de données

## [Recommandé] Acquisition de données via ROS

Se référer à : [https://github.com/orbbec/OrbbecSDK\\_ROS2](https://github.com/orbbec/OrbbecSDK_ROS2)

**Remarque :** Au démarrage, le T1 lance automatiquement le nœud RealSense ROS via `booster-daemon-perception.service` . Pour éviter les conflits lors du lancement du nœud Orbbec, arrêtez le service.

Shell

```
sudo systemctl stop booster-daemon-perception.service
```

- Sujets publiés par le nœud ROS Orbbec lancé automatiquement :

Shell

```
/camera/color/camera_info      # Paramètres intrinsèques de la
caméra couleur
/camera/color/image_raw        # Image couleur brute
/camera/color/image_raw/compressed
/camera/color/metadata
/camera/depth/camera_info      # Paramètres intrinsèques de la
caméra de profondeur
/camera/depth/image_raw        # Image brute de profondeur
/camera/depth/image_raw/compressed
/camera/depth/metadata
/camera/depth_to_color/        # Profondeur alignée avec la couleur
```

Pour démarrer manuellement le nœud ROS Orbbec :

Shell

```
cd ~/ThirdParty/OrbbecSDK_ROS2
source install/setup.bash
ros2 launch orbbec_camera gemini_330_series.launch.py
```

- Pour activer des fonctionnalités supplémentaires telles que l'alignement profondeur-RGB et les flux IMU, reportez-vous aux définitions des paramètres dans `gemini_330_series.launch.py` .

- Pour un script Python permettant de s'abonner à des images couleur et de profondeur et de les enregistrer, reportez-vous à l'exemple fourni ci-dessus. Assurez-vous que les topics d'images sont mis à jour en conséquence.

## Acquisition de données via SDK

Le kit de développement logiciel Orbbec est préinstallé sur la carte de perception. Voir : <https://github.com/orbbec/OrbbecSDK>

- Avant d'utiliser le SDK, arrêtez le `service booster-daemon-perception.service` afin d'éviter les conflits de ressources.

## Commencer

### Récupération du SDK

- Lien de téléchargement : [https://github.com/BoosterRobotics/booster\\_robotics\\_sdk](https://github.com/BoosterRobotics/booster_robotics_sdk)
- Suivez les instructions du fichier README dans le dépôt pour terminer l'installation du SDK sur l'ordinateur du développeur.

### Installer le SDK

Shell

```
# Dans le répertoire booster_robotics_sdk
sudo ./install.sh
```

### Compiler les exemples de programmes et installer le SDK Python

En supposant que le SDK soit installé dans `/home/booster/Workspace`, accédez au chemin du projet SDK et exécutez :

Shell

```
# BUILD_PYTHON_BINDING=on indique que le SDK Python sera compilé.
# Si vous devez compiler le SDK Python, vous devez installer manuellement les dépendances suivantes :
# pip3 installer pybind11
# (Après l'installation, si cmake ne trouve pas pybind11, vous pouvez l'installer manuellement via sudo apt install pybind11-dev)
# pip3 install pybind11-stubgen
# (Si pybind11-stubgen est installé sous ~/.local,
# Vous devez exporter manuellement
PATH=/home/[nom_utilisateur]/.local/bin:$PATH)

cd /home/booster/Workspace/booster_robotics_sdk
```

```
mkdir build
cd construire
cmake .. -DBUILD_PYTHON_BINDING=on
make
sudo make install
```

### Compiler uniquement les programmes d'exemple

En supposant que le SDK soit installé dans `/home/booster/Workspace`, accédez au chemin du projet SDK et exécutez :

```
Shell
cd /home/booster/Workspace/booster_robotics_sdk
mkdir build
cd build
cmake ..
make
```

Après compilation, les programmes d'exemple seront générés dans le répertoire de compilation, y compris le programme `b1_loco_example_client` comme exemple d'interface de service de haut niveau.

### Exemple : Développement basé sur la simulation Webots

#### Travaux préparatoires : Installation environnementale

##### 1. Configuration système requise

Élément	Bonnes spécifications
Système d'exploitation	Ubuntu 22.04
processeur	Intel Core i7 (7e génération)
Noyaux	4

BÉLIER	16 GB
--------	-------

## 2. Installation de Webots

[Voir la documentation en ligne](#)

1. Décompressez et copiez Webots dans le répertoire `/usr/local` .
2. `sudo cp -r webots/ /usr/local/`
3. Configurez la variable d'environnement (le chemin doit correspondre à l'emplacement d'installation).

Shell

```
# ~/.bashrc
```

```
exporter WEBOTS_HOME=/usr/ local /webots
```

```
exporter
```

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$WEBOTS_HOME/lib/controller
```

3. [Installez le SDK.](#)
4. [Compiler et exécuter des exemples de programmes \(facultatif\).](#)

## Développement dans l'environnement de simulation Webots

1. Charger les fichiers Webots world.

[Voir la documentation en ligne](#)

1. Décompressez les fichiers de simulation Webots.
2. Ouvrir dans Webots :
  1. Fichier -> Ouvrir le monde -> Sélectionnez le fichier `.wbt` . L'un a des bras à 4 degrés de liberté, l'autre a des bras à 7 degrés de liberté.
2. Installer la dépendance

Shell

```
"cmake"
```

```
"ninja-build"
```

```
"libgtest-dev"
```

```
"libgoogle-glog-dev"
```

```
"libboost-dev"
```

```
"libeigen3-dev"
```

```
"liblua5.3-dev"  
"graphviz"  
"libgraphviz-dev"  
"python3-pip"  
"libcurl4-openssl-dev"  
"libsdl2-dev"  
"joystick"  
"libspdlog-dev"
```

3. Lancez le programme de simulation de contrôle. L'un utilise des bras à 4 degrés de liberté, l'autre à 7 degrés de liberté.

[Voir la documentation en ligne](#)

[Voir la documentation en ligne](#)

4. Dans le terminal, exécutez `./booster-runner-full-0.0.x.run`
  1. Si le programme ne s'exécute pas, utilisez la commande : ``chmod +x booster-runner-full-0.0.x.run``
5. Lorsque la simulation est en cours d'exécution, ce programme doit rester ouvert en permanence.
6. [exécuter des exemples de programmes \(facultatif\).](#)

Shell

Remarque : L'adresse IP indiquée ici doit être l'adresse IP locale. Si vous exécutez ce programme sur la carte de contrôle de mouvement, vous pouvez utiliser 127.0.0.1. Si vous l'exécutez sur la carte de perception, vous pouvez utiliser 192.168.10.102.

# Étant donné qu'il est développé sur la machine locale, l'adresse IP cible du fichier dds est 127.0.0.1

```
./b1_loco_example_client 127.0.0.1
```

# Passer en mode marche

mw

# Contrôlez le robot pour qu'il avance.

w

# Contrôler le robot pour qu'il recule

s



```
# Contrôlez le robot pour qu'il se déplace vers la gauche  
a  
# Contrôlez le robot pour qu'il s'arrête de marcher.  
1
```

## Exemple : Développement d'un environnement local basé sur Isaac Simulation Link (ROS2)

### Travaux préparatoires : Configuration de l'ordinateur utilisateur

#### 1. Configuration système requise

Élément	Spécifications minimales
Système d'exploitation	Ubuntu 22.04
processeur	Intel Core i7 (7e génération)
Noyaux	4
BÉLIER	32 Go
GPU	GeForce RTX 3070
VRAM	8 Go

1. Installer [Isaac Sim](#)
  1. Installer [Pilotes NVIDIA](#) si ce n'est pas déjà fait.
  2. Télécharger et installer [Isaac Omniverse](#).
  3. Installez Isaac Sim 4.2 depuis la section EXCHANGE.
2. Installer [ROS2](#).
3. [Installez le SDK](#).
4. [Compiler et exécuter des exemples de programmes \(facultatif\)](#).

### Développement dans l'environnement de simulation Isaac !

L'environnement de simulation Isaac étant très différent de l'environnement réel, il est déconseillé d'y développer des systèmes de contrôle de mouvement. Il est en revanche plus adapté au développement de programmes de perception et de prise de décision.

1. Isaac démarre en un clic. L'un possède des bras à 4 degrés de liberté, les autres à 7 degrés de liberté.

[Voir la documentation en ligne](#)

[Voir la documentation en ligne](#)

[Voir la documentation en ligne](#)

2. Il y a deux façons de démarrer ; si le programme ne s'exécute pas, utilisez la commande `chmod +x isaac_package_0.0.x.run``.

1. Dans l'interface graphique, cliquez avec le bouton droit et choisissez « exécuter en tant que programme ».
2. Dans le terminal, exécutez `./isaac_package_0.0.x.run`
3. Si vous souhaitez utiliser votre propre environnement, vous pouvez exécuter ce script comme suit : `./isaac_package_0.0.x.run`  
`CHEMIN-VERS-VOTRE-USD.usd`
4. Après l'exécution, saisissez le répertoire d'exécution local d'Isaac ; l'environnement Isaac démarrera automatiquement.

Une fois Isaac installé, le programme qui exécute le script est généralement généré dans `~/.local/share/ov/pkg/isaac-sim-4.2.0/python.sh`

5. Installer la dépendance

Shell

```
"cmake"  
"ninja-build"  
"libgtest-dev"  
"libgoogle-glog-dev"  
"libboost-dev"  
"libeigen3-dev"  
"liblua5.3-dev"  
"graphviz"  
"libgraphviz-dev"  
"python3-pip"  
"libcurl4-openssl-dev"  
"libsdl2-dev"  
"joystick"  
"libspdlog-dev"
```

6. Exécutez le programme de simulation de contrôle. L'un utilise des bras à 4 degrés de liberté, l'autre des bras à 7 degrés de liberté.

[Voir la documentation en ligne](#)

[Voir la documentation en ligne](#)

- a. Il y a deux façons de démarrer ; si le programme ne se lance pas, utilisez la commande ``chmod +x booster-runner-full-0.0.x.run``.
  - i. Dans l'interface graphique, cliquez avec le bouton droit et choisissez « exécuter en tant que programme ».
  - ii. Dans le terminal, exécutez `./booster-runner-full-0.0.x.run`

- b. Après l'exécution, sélectionnez l'environnement à utiliser, saisissez « isaac », et le contrôle de mouvement Isaac pourra être lancé. Pendant la simulation, ce programme doit rester ouvert.
7. Comment lister les sujets ROS
- a. Créez le fichier fastdds\_profile.xml, son contenu est le suivant.

XML

```
<?xml version="1.0" encoding="UTF-8" ?>

<profiles
  xmlns="http://www.eprosima.com/XMLSchemas/fastRTPS_Profiles" >
  <transport_descriptors>
    <transport_descriptor>
      <transport_id>UdpTransport</transport_id>
      <type>UD Pv4</type>
      <interfaceWhiteList>
        <address>127.0.0.1</address>

      </interfaceWhiteList>
    </transport_descriptor>
  </transport_descriptors>

  <participant profile_name="udp_transport_profile"
    is_default_profile="true">
    <rtps>
      <userTransports>
        <transport_id>UdpTransport</transport_id>
      </userTransports>
      <useBuiltinTransports>false</useBuiltinTransports>
    </rtps>
  </participant>
</profiles>
```

- b. Exporter l'environnement

Shell

```
export
FASTRTPS_DEFAULT_PROFILES_FILE=CHEMIN_VERS/fastdds_profile.xml
ros2 daemon stop
ros2 daemon start
ros2 topic list
```

8. Développement de la vision

- a. Le topic Vision dans Isaac Sim est **camera/camera/color/image\_raw** .
- b. Le topic de profondeur de vision est **camera/camera/depth/image\_rect\_raw**
- c. Abonnez-vous au thème de la vision pour développer un programme de vision.

9. [exécuter des exemples de programmes \(facultatif\).](#)

Shell

```
# Remarque : L'adresse IP indiquée ici doit être l'adresse IP
locale. Si vous exécutez ce programme sur la carte de contrôle
de mouvement, vous pouvez utiliser 127.0.0.1. Si vous
l'exécutez sur la carte de perception, vous pouvez utiliser
192.168.10.102.
```

```
# Étant donné qu'il est développé sur la machine locale,
l'adresse IP cible du fichier dds est 127.0.0.1
./b1_loco_example_client 127.0.0.1
```

```
# Passer en mode marche
```

```
mw
```

```
# Contrôlez le robot pour qu'il avance.
```

```
w
```

```
# Contrôler le robot pour qu'il recule
```

```
s
```

```
# Contrôlez le robot pour qu'il se déplace vers la gauche
```

```
a
```

```
# Contrôlez le robot pour qu'il s'arrête de marcher.
```

```
l
```

Guide du développeur ROS2

Pour le processus de développement secondaire basé sur ROS2, veuillez vous référer au manuel suivant : [Guide de développement sur ROS2](#)

## Pipeline de formation et de déploiement en apprentissage par renforcement

[Formation et déploiement RL](#)

## Exemple de développement d'application robotique

[Démonstration RoboCup Humanoid 2V2](#)

## Téléchargement de ressources

Shell

```
Copyright [2024] [Booster Robotics Technology Co., Ltd  
("Booster Robotics")]
```

```
Licensed under the Apache License, Version 2.0 (the  
"License");
```

```
you may not use this file except in compliance with the  
License.
```

```
You may obtain a copy of the License at
```

```
http://www.apache.org/licenses/LICENSE-2.0
```

```
Unless required by applicable law or agreed to in writing,  
software
```

```
distributed under the License is distributed on an "AS IS"  
BASIS,
```

```
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express  
or implied.
```

```
See the License for the specific language governing  
permissions and
```

```
limitations under the License.
```

# Fichier URDF et MESH

## Robot à bras à 4 degrés de liberté

Modèle série URDF :

[Voir la documentation en ligne](#)

Modèle parallèle URDF :

[Voir la documentation en ligne](#)

Fichier MESH :

[Voir la documentation en ligne](#)

## Robot à bras à 7 degrés de liberté

Modèle de cheville en série URDF :

[Voir la documentation en ligne](#)

Modèle de cheville parallèle URDF :

[Voir la documentation en ligne](#)

Fichier MESH :

[Voir la documentation en ligne](#)

## Robot à bras à 7 degrés de liberté avec pince

Modèle série URDF :

[Voir la documentation en ligne](#)

[Voir la documentation en ligne](#)

# USD

[Voir la documentation en ligne](#)

## Robot URDF pour l'entraînement par renforcement

### Salle de sport Isaac

1. Le modèle comprend la tête et le bras

[Voir la documentation en ligne](#)

2. Modèle avec poignet et jambe

[Voir la documentation en ligne](#)

3. Modèle complet du corps contenant des colliders simplifiés.

[Voir la documentation en ligne](#)

4. Modèle avec bras à 7 degrés de liberté

[Voir la documentation en ligne](#)

5. Modèle avec bras à 7 degrés de liberté et collision de bras

[Voir la documentation en ligne](#)

## MuJoCo

1. Modèle avec bras à 7 degrés de liberté

[Voir la documentation en ligne](#)

2. Modèle avec bras à 7 degrés de liberté

[Voir la documentation en ligne](#)